SYBASE®

Configuring Adaptive Server for UNIX Platforms

Sybase Adaptive Server Enterprise

Version 12.0

UNIX

DOCUMENT ID: 35823-01-1200-01

LAST REVISED: March 2000

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

Contents

About This Book

This guide, *Configuring Adaptive Server for UNIX Platforms*, provides instructions for performing specific configuration tasks for Sybase® Adaptive Server™ Enterprise on the following platforms:

- Digital UNIX
- HP 9000 Series 800 HP-UX
- IBM RISC System/6000 AIX
- Sun Solaris 2.x (SPARC)

The operating system release levels for these platforms on which Adaptive Server is certified to execute is provided in the installation documentation or release bulletin for your platform.

This guide provides:

- An overview of the Adaptive Server infrastructure
- Instructions for reconfiguring certain attributes of your existing Adaptive Server, Backup Server™, and XP Server™ to meet your specific needs

---

Note Before configuring Adaptive Server according to the instructions in this book, you should have a newly installed or upgraded Adaptive Server on your system. If you do not, follow the installation and upgrade instructions in the installation documentation for your platform.

Audience

This guide is written for Sybase System Administrators.

How to use this book

This book contains the following chapters:

- Chapter 1, “Introduction” presents an overview of the configuration changes you might want to make to Adaptive Server after installing or upgrading the server.
- Chapter 2, “Starting and Stopping Servers” describes how to start and stop Adaptive Server, Backup Server, Monitor Server, and XP Server.

- Chapter 3, “Configuring the Operating System for Adaptive Server” describes operating system configuration issues.
- Chapter 4, “Setting Up Communications Across a Network” describes how to use the interfaces file to establish network connections for servers and clients.
- Chapter 5, “Customizing Localization for Adaptive Server” provides background information about Sybase’s localization support as well as task information for reconfiguring character sets, sort orders, and language modules.
- Chapter 6, “Licensing Adaptive Server Features” describes how to license optional Adaptive Server features, and add new licenses to existing servers using Sybase Software Asset Management (SySAM).
- Chapter 7, “Adding Optional Functionality to Adaptive Server” provides instructions for adding optional functionality such as auditing and the sample databases.

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 Adaptive Server installation documentation for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What’s New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 11.9.3, 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.
- *Adaptive Server 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 Programs* manual for your platform – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Adaptive Server Glossary* – defines technical terms used in the Adaptive Server documentation.
- *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 and user-defined functions 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.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using Sybase's DTM XA Interface with X/Open XA transaction managers.

- *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.

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 technical documents and is included with your software. The DynaText browser (included on the Technical Library CD) 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 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'll 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://sybooks.sybase.com.

Sybase certifications on the web

Technical documentation at the Sybase web site is updated frequently.

❖ **For the latest information on product certifications and/or the EBF Rollups:**

1 Point your web browser to Technical Documents at http://techinfo.sybase.com.

2 In the Browse section, click on What's Hot.

3 Select links to Certification Reports and EBF Rollups, as well as links to Technical Newsletters, online manuals, and so on.

❖ **If you are a registered SupportPlus user:**

1 Point your web browser to Technical Documents at http://techinfo.sybase.com.

2 In the Browse section, click on What's Hot.

3 Click on EBF Rollups.

You can research EBFs using Technical Documents, and you can download EBFs using Electronic Software Distribution (ESD).

4 Follow the instructions associated with the SupportPlus^{SM} Online Services entries.

❖ **If you are not a registered SupportPlus user, and you want to become one:**

You can register by following the instructions on the Web.

To use SupportPlus, you need:

1 A Web browser that supports the Secure Sockets Layer (SSL), such as Netscape Navigator 1.2 or later

2 An active support license

3 A named technical support contact

4 Your user ID and password

❖ **Whether or not you are a registered SupportPlus user:**

You may use Sybase's Technical Documents. Certification Reports are among the features documented at this site.

1 Point your web browser to Technical Documents at http://techinfo.sybase.com

2 In the Browse section, click on What's Hot.

3 Click on the topic that interests you.

Conventions

Table 1 shows the conventions used in this manual.

Table 1: Conventions for this manual

Item	Example
Command names, keywords, command options, utility names, utility flags, and system procedure names	create database isql -U sp_configure
Database names	master, model, and tempdb databases
Datatypes	int, datetime, and varchar datatypes
File names and path names	*interfaces* *$SYBASE/bin*

Variables, or words that placeholders for values that you fill in	`select` *`column_name`*
	`from` *`table_name`*
	`where` *`search_conditions`*

Examples showing the use of Transact-SQL commands	`select * from publishers`

Item	Example
Examples of output from the computer	0736 New Age Books Boston MA 0877 Binnet & Hardley Washington DC 1389 Algodata Infosystems Berkeley CA
Parentheses – if shown in the syntax, type them as part of the command.	compute *row_aggregate* (*column_name*)
Comma – choose one or more options. Type commas as part of the command.	cash, check, credit
Vertical bar – choose only one option. Do not type the bars.	cash | check | credit
Curly braces around options separated by commas – you must choose at least one option. You may choose more than one. Do not type the braces.	{cash, check, credit}
Curly braces around options separated by vertical bars – you must choose only one option. Do not type the braces.	{cash | check | credit}
Brackets around options separated by commas – choosing an option is optional. You may choose more than one. Do not type the brackets.	[cash, check, credit]
Brackets around options separated by vertical bars – choosing an option is optional. You may choose only one. Do not type the brackets.	[cash | check | credit]
An ellipsis (...) means that you can repeat the preceding optional unit (between the square brackets) as many times as you like.	buy *toy* = *price* {cash | check | credit} [, *toy* = *price* {cash | check | credit}]...

In this example, you must specify at least one toy and enter its price. You must choose only one method of payment: cash, check, or credit. You may also buy additional toys. Separate each toy with a comma. You may put each item on a separate line for readability, but this is not required.	buy doll = 25.50 check,
	train = 60.00 credit,
	bicycle = 150.00 credit,
	paint set = 6.00 cash

Transact-SQL does not restrict the number of words you can put on a line or where to break a line. However, for readability, 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.

You can disregard case when typing Transact-SQL keywords. For example, Adaptive Server understands SELECT, Select, and select to be the same. In this manual, most examples are in lower case. Case sensitivity of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. For more information, see Chapter 19, "Configuring Character Sets, Sort Orders, and Languages," in the System Administration Guide.

Commands for both the C and Bourne shells are provided in this document, when they differ. The initialization file for the C shell is called *.cshrc*. The initialization file for the Bourne shell is called *.profile*. If you are using a different shell, such as the Korn shell, see your shell-specific documentation for the correct command syntax.

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 Introduction

This chapter provides an overview of Adaptive Server configuration and the steps you need to take to customize it for your use.

About Adaptive Server

Adaptive Server performs data management and transaction functions, independent of client applications and user interface functions. It also:

- Manages multiple databases and multiple users
- Keeps track of the data’s location on disks
- Maintains the mapping of logical data description to physical data storage
- Maintains data and procedure caches in memory

Adaptive Server uses these auxiliary programs to perform dedicated tasks:

- Backup Server – manages database load, dump, backup, and restoration activities.
- Adaptive Server Enterprise Monitor Historical Server – obtains performance data from Monitor Server and saves the data in files for use at a later time.

- XP Server – stores the extended stored procedures (ESPs) that allow Adaptive Server to run operating-system level commands.

Note The instructions in this book assume that Adaptive Server is installed and running unless noted otherwise.

System-specific issues

Adaptive Server runs on a variety of hardware and operating system platforms. System-specific issues do not affect the basic functionality of Adaptive Server, but there are differences among platform implementations. These differences may include:

- Changes to the operating system that enable or enhance Adaptive Server performance
- Operating system commands or utilities that simplify or automate routine system administration tasks
- Options for selecting database devices
- The structure of entries in the interfaces file
- Adaptive Server configuration

For more information about system-specific issues, see the installation documentation and release bulletin for your platform.

Definition of terms

The following terms are used in this book:

- *Unload* – to copy Adaptive Server products from the distribution media to your machine, using the Studio Installer.
- *Install* – to make the Adaptive Server products fully functional by configuring various attributes.
- *Upgrade* – to configure an existing Adaptive Server to a more recent Adaptive Server version level.

- *Server* – provides a service in client/server computing. Examples include Adaptive Server, Backup Server, Historical Server, Monitor Server, and XP Server.
- *Client* – requests a service in client/server computing. Sybase Central™, PowerDynamo™, PowerDesigner®, SQL Modeler™, and end-user applications are clients.

User roles

The Adaptive Server installation and setup process defines various user roles. Different user roles have different responsibilities and privileges. These user roles clarify the way Adaptive Server is to be integrated into your system:

- *Operating System Administrator* – the individual who maintains the operating system. This individual has superuser or "root" privileges.
- *System Administrator* –the individual in charge of Adaptive Server system administration, creating user accounts, assigning permissions on databases, and creating new databases. At installation time, the System Administrator's login name is "sa". The "sa" login is not a UNIX login. The "sa" login is specific to Adaptive Server and is used to log in to Adaptive Server using the isql command.
- *Sybase Login* – the "sybase" login is a UNIX login that owns all the Sybase installation directories and files, sets permissions on those directories and files, and performs the installation and upgrading of Adaptive Server.

Environment variables

It is crucial to the operation of Sybase products that the system environment variables be set correctly.

Environment variables are set in the user's environment either interactively or by including them in the user's *.login* and *.cshrc* files (for C shell) or *.profile* file (for Bourne shell). The installation instructions in this guide explain when to set these variables:

- DSLISTEN – defines the name Adaptive Server uses to listen for client connections if no name is given during the Adaptive Server start-up. If DSLISTEN is not set, and no name is given during the Adaptive Server start-up, the Adaptive Server name defaults to the server name given at installation.
- DSQUERY – defines the Adaptive Server name that client programs try to connect to if no Adaptive Server is specified with a command-line option. If DSQUERY is not set, and you do not supply the Adaptive Server name with a command-line option, clients attempt to connect to the server name given at installation.
- SYBASE – defines the path of the Sybase installation directory. The installation program sets up the variable SYBASE to point to the release directory specified during installation.
- SYBASE_ASE – defines the subdirectory directory of the Adaptive Server components.
- SYBASE_OCS – defines the subdirectory to which the Open Client™ is set.
- SYBASE_ FTS – defines the subdirectory for the full-text search components.
- SYBASE_SYSAM – points to the license-management software directory.
- LM_LICENSE_FILE – points to the *license.dat* file in the *SYBASE_SYSAM/licenses/* directory.
- PATH – specifies which directory path to search for executables. The Sybase executables are located in the installed components */bin* directory. When using the source files *SYBASE.csh* or *SYBASE.sh*, the following paths are prepended to PATH:

  ```
  $SYBASE/$SYBASE_ASE/bin:$SYBASE/$SYBASE_OCS/bin
  ```
- LD_LIBRARY_PATH – on Digital UNIX, HP-UX, and Sun Solaris platforms, this variable specifies which directory to search for executables. The Sybase executables are located in the installed component's */bin* directory.

  When using the source files: *SYBASE.csh* or *SYBASE.sh*, the following paths are prepended to the LD_LIBRARY_PATH environment variable:

```
$SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/lib:$SYBASE/SYBASE_FTS/lib,
etc.
```

- LIBPATH – on the IBM RS/6000 platform, this variable specifies which directory to search for executables. The Sybase executables are located in the installed component's */bin* directory.

  When using the source files: *SYBASE.csh* or *SYBASE.sh*, the following paths are prepended to the LIBPATH environment variable:

  ```
  $SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/lib:
  $SYBASE/SYBASE_FTS/lib, etc.
  ```

Sybase installation directory

The Sybase installation directory structure is created by the installation process. Adaptive Server is placed in the directory you indicate when you run the Studio Installer. Table 1-1 illustrates part of the Adaptive Server installation directory structure.

The Adaptive Server installation directory contains executable files and administrative tools, which are added as each product is installed.

Table 1-1: Installation directory for $SYBASE

Component directory	Subdirectory	Subdirectory
/ASE-12_0		
	/bin	
	/debugger	
	/init	/auditinit
		/bsrv
		/logs
		/sqlsrv
	/install	/spr
	/sample	/esp
		/JavaSql
		/server
	/scripts	
	/upgrade	
/charsets		
/collate	/unicode	
/config		
/FTS-12_0		
/installed		
/Installer	/bin	
	/lib	
/jConnect_docs	/docs	
/jConnect-4_2	/classes	/com /gateway /sample
	/devclasses	/com
	/docs	/en
	/gateway	
	/sample	
	/sp	
	/tools	
	/devclasses	
/jConnect-5_2	/classes	/gateway2 /sample2
	/devclasses	
	/docs	/en
	/gateway2	

Component directory	Subdirectory	Subdirectory
	/sample2	
	/sp	
	/tools	
/jre-1_1-NT386	/bin	
(This directory in installed on Digital UNIX, HP-UX and Sun Solaris platforms only.)	/lib	/security
/jutils-2_0	/cascade	/uk
	/jisql	/doc /HelpFiles
	/ribo	/doc
/locales	/english	
	/message	
	/unicode	
/OCS-12_0	/bin	
	/config	
	/include	
	/lib	
	/locales	
	/sample	
	/scripts	
/SQLRemote	/bin	
	/\scripts	
/SYSAM-1_0	/bin	
	/licenses	

Adaptive Server devices and system databases

Devices are files or portions of a disk that are used to store databases and database objects. You can initialize devices, using raw disk partitions (for production systems) or operating system files (for nonproduction uses).

Adaptive Server requires the following devices:

- master – to store system databases.
- sysprocsdev – to store system procedures.

The master and sysprocsdev devices are created when you create a new Adaptive Server.

The master device

The master device contains the following databases:

- master – controls the operation of Adaptive Server as a whole and stores information about all users, user databases, devices, objects, and system table entries. The master database is contained entirely on the master device and cannot be expanded onto any other device.
- model – provides a template for new user databases. The model database contains required system tables, which are copied into a new user database with the create database command.
- tempdb – the work area for Adaptive Server. Each time Adaptive Server is started the tempdb database is cleared and rebuilt from the model database.
- The sample databases are stored on the master device at installation, but should be moved to a user-defined device after installation. For more information, see "The sample databases" on page 9.

Note For recovery purposes, Sybase recommends that you do not create other system or user databases or user objects on the master device.

The sybsystemdb device

The sybsystemdb device stores the sybsystemdb database, which stores information about transactions in progress, and which is also used during recovery.

For instructions about creating the sybsystemdb device and database for Data Transfer Management (two-phase commit), see Chapter 9, "Upgrading Sybase Servers" in the Adaptive Server installation documentation for your platform.

The sysprocsdev device

The sybprocsdev devices stores the sybsystemprocs database, which contains most of the Sybase-supplied system procedures. System procedures are a collection of SQL statements and flow-of-control statements that perform system tasks, for example, sp_configure.

The system procedures that are needed during recovery situations are stored in the master database.

Note sysprocsdev is the default system name for this device. However, it is frequently referred to as the sybsystemprocs device, since it stores the sybsystemprocs database.

Optional devices and databases

The devices and databases described in the following sections are optional.

The sample databases

The sample databases are:

- pubs2 and pubs3 are provided as learning tools for Adaptive Server. pubs2 is used for most of the examples in the Adaptive Server documentation; other examples use the pubs3 database. Both are available in U.S. English versions of Adaptive Server.
- interpubs database contain French and German data.
- jpubs contains Japanese data.

For information about installing the sample databases, see Chapter 7, “Adding Optional Functionality to Adaptive Server”.

For information about the contents of the sample databases, see the *Transact-SQL User’s Guide*.

The sybsecurity device and database

The sybsecurity device is created as part of the auditing installation process. The sybsecurity device stores the sybsecurity database and the auditing system procedures with which you can configure auditing for your system.

The auditing system records system security information in an Adaptive Server audit trail. You can use this audit trail to monitor the use of Adaptive Server or system resources.

For instructions on configuring Adaptive Server for auditing, see Chapter 7, “Adding Optional Functionality to Adaptive Server”. For information about installing and using the auditing system, see Chapter 8, “Auditing,” in the *System Administration Guide*.

dbccdb database

The database consistency checker (dbcc) provides commands for checking the logical and physical consistency of a database. The dbccdb database stores the results of dbcc when dbcc checkstorage or dbcc check verifying are used.

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 information on installing and using dbccdb, see Chapter 25, “Checking Database Consistency,” in the *System Administration Guide*.

Client/server communication (the interfaces file)

Adaptive Server communicates with other Adaptive Servers, Open Server™ applications (such as Backup Server), and client software on your network. Clients can talk to one or more servers, and servers can communicate with other servers by remote procedure calls.

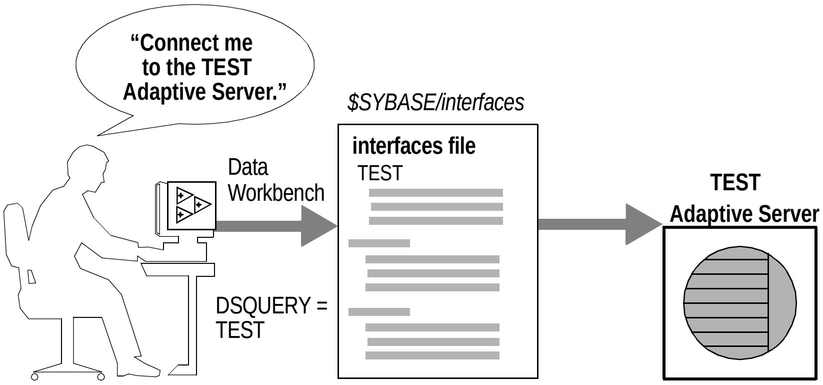
For Sybase products to interact with one another, each product needs to know where the others reside on the network. This information is stored in an interfaces file, named *interfaces* on UNIX platforms, located in the *$SYBASE* installation directory. After your Adaptive Server or client software is installed, it can connect with any server on the network that is listed in the interfaces file.

The interfaces file is like an address book. It lists the name and address of every known server. When you are using a client program, and you want to connect with a particular server, the client program looks up the server name in the interfaces file and connects to that server, as shown in Figure 1-1. You can supply the name of the server by using the DSQUERY environment variable.

On TCP/IP networks, the port number gives clients a way to identify the Adaptive Server to which they want to connect. It also tells Adaptive Server where to listen for incoming connection attempts from clients. Adaptive Server uses a single port for these two services (referred to as *query service* and *listener service*).

On SPX networks, the socket number gives clients and servers a way to identify each other.

Figure 1-1: Communicating with a server



During installation, you use the srvbuild utility to create a new server. The srvbuild process adds entries to the interfaces file for your new Adaptive Server, Backup Server, Monitor Server, and XP Server.

For instructions on using dsedit or dscp to modify interfaces file entries or to create new interfaces file entries for existing servers, see Chapter 3, "Configuring the Operating System for Adaptive Server."

Changing Adaptive Server configuration

On UNIX platforms, you use sp_configure to change the configuration of Adaptive Server. To change the configuration of several servers, enter the sp_configure information in a script. For details on using sp_configure, see the *System Administration Guide* and the *Adaptive Server Reference Manual*.

For information about configuring languages, character sets, and sort orders, see Chapter 5, "Customizing Localization for Adaptive Server."

For information about configuring Adaptive Server to use high availability features, see *Using Sybase Failover in a High Availability Environment.*

For information about configuring Adaptive Server for distributed transaction management (two-phase commit), see the *Distributed Transaction Management User's Guide*.

CHAPTER 2 Starting and Stopping Servers

This chapter describes how to start and stop Adaptive Server, Backup Server, Monitor Server, and XP Server.

Overview

Use the methods described in this chapter to start Adaptive Server, Backup Server, and Monitor Server after a shutdown for database maintenance, because of an operating system crash, or for other reasons.

XP Server is not started by the installation process. XP Server is started only when an XP command is issued through **isql**.

For information about starting the Full-Text Search server, see the *Standard Full-Text Search Specialty Data Store User's Guide*.

You can start Adaptive Server, Backup Server, and Monitor Server on UNIX either by:

- Entering commands from the UNIX command line, or
- Configuring your operating system so that the servers start automatically when the operating system starts.

Starting servers

You can start Adaptive Server, Backup Server, or Monitor Server from the UNIX command line by using *runserver* files and **startserver** commands. You can move the Adaptive Server shared memory files by adding a flag to the *runserver* file.

Using a runserver file

Each time a new Adaptive Server, Backup Server, or Monitor Server is installed, the **srvbuild** program creates a *runserver* file that contains the information required to restart that server. Runserver files are created in the *$SYBASE/$SYBASE_ASE/install* directory. For Adaptive Server, the runserver file is named *RUN_servername*, where *servername* is the name of the server.

For example, the runserver file name for an Adaptive Server named TEST is *RUN_TEST*.

For Backup Server and Monitor Server, the runserver files are named *RUN_servername_back* and *RUN_servername_mon*, respectively, where *servername* is the name of the server.

Warning! Do not delete the runserver file that is created in *$SYBASE/$SYBASE_ASE/install*. This file is used to restart servers when you customize your installation. If you need the runserver file in another location, make a copy of the original runserver file in the new location.

The runserver file for a server named SYBASE is named *RUN_SYBASE*. If the runserver file for your current Adaptive Server is named *RUNSERVER*, you must change the name to *RUN_SYBASE* during the upgrade process.

Using the startserver command

To start a server from the command line, enter:

```
$SYBASE/$SYBASE_ASE/install/startserver [ -f runserverfile ]
```

where *$SYBASE/$SYBASE_ASE/install/startserver* is the full path name of the **startserver** utility, and *runserverfile* is the full path name of the runserver file (usually *$SYBASE/$SYBASE_ASE/install/RUN_servername*).

If your Adaptive Server is named SYBASE, the runserver file specification is optional.

You must have read and write permissions on the master device for an Adaptive Server to start it with **startserver**. For more information about **startserver**, see *Utility Programs for UNIX Platforms*.

Although **startserver** runs Adaptive Server in the background, it does not return to the operating system prompt after restarting. After Adaptive Server has been restarted, press Return to return to the operating system prompt.

Using the monserver command

Use the **monserver** command to start Monitor Server from the UNIX command line. For details, see the *Adaptive Server Enterprise Monitor Server User's Guide*.

Starting servers when the operating system restarts

This section describes how to configure your operating system for automatic restart of Adaptive Server and Backup Server.

On production systems, Adaptive Server, Backup Server, and Monitor Server should restart automatically when the UNIX operating system starts. This is accomplished by making an entry for the server in the operating system start-up script. Monitor Server must be listed *after* Adaptive Server in the start-up script.

For more information on starting Monitor Server at operating system restart, see the *Sybase Adaptive Server Enterprise Monitor Server User's Guide*.

For Digital UNIX

Before configuring your operating system for automatic restart:

1. Before starting the server, define the environment variable LD_LIBRARY_PATH.
2. Add *$SYBASE/product_name/lib* first to LD_LIBRARY_PATH. If */usr/shlib* comes before *$SYBASE/product_name/lib* in LD_LIBRARY_PATH, you may encounter unresolvable symbol errors.
3. Verify that all network resources are available before you start Adaptive Server or Backup Server. The server cannot start if the network is down.
4. Verify that the entry in the *rc* (run control) directory for restarting the server *follows* any commands for starting network operations.
5. Designate a period of time for your server to wait before starting, by inserting a **sleep** command at the beginning of the linked *runserver* file you created. Slow start-up of network operations can prevent servers from starting, even when the commands are in the correct order in the *rc* directory.

To configure your operating system for automatic restart of Adaptive Server and Backup Server:

1 Create a start-up script file like the following. The start-up file is a Bourne shell script. You must use Bourne shell conventions in this file, even if you use the C shell for other tasks.

```
#!/sbin/sh
# start a SYBASE Adaptive Server on system reboot
# set SYBASE to the path of the sybase directory
# set RUNSERVER to the name of the runserver file
#     for the Adaptive Server
# set BSERVER to the name of the runserver file
#     for the backup server
#
SYBASE=/sybase/rel1151
RUNSERVER=RUN_REL1151_XYZ
BSERVER=RUN_REL1151_XYZ_BACKUP
PATH=/sbin:/usr/sbin:/usr/bin
#
export PATH SYBASE RUNSERVER BSERVER
#
case "$1" in
’start’)
   echo "Starting Sybase Adaptive Server"
   nohup su sybase -c \
      "$SYBASE/$SYBASE_ASE/install/startserver -f
       $SYBASE/$SYBASE_ASE/install/$RUNSERVER >/dev/null &"
   echo "Starting Sybase Backup Server"
   nohup su sybase -c \
      "$SYBASE/$SYBASE_ASE/install/startserver -f
       $SYBASE/$SYBASE_ASE/install/$BSERVER >/dev/null &"
   ;;
*)
   echo "Usage:  $0 {start}"
   ;;
esac
```

2 Copy the script to the */sbin/init.d* directory, using syntax like the following:

```
cp script_name /sbin/init.d/script_name
```

3 Review the “initdefault” entry in the */etc/inittab* file to determine the default run level for your operating system. The default run level is usually 2 or 3.

4 Use the **ln** command to create a hard link from the runserver file to the appropriate run control (*rc*) directory, *rc#*, where # is the default run level obtained in step 3.

For example:

```
ln /sbin/init.d/script_name /sbin/rc#.d/S##script_name
```

When you type the command to create the link, add an uppercase "S" and a two-digit sequence number preceding the name of the script file. "S" indicates a start-up file. The sequence number is necessary because files in the *rc* directory are executed in numeric order. This file should be executed last, so use a number that comes after all existing numbers in ASCII sort order.

You can perform an **ls** command on the *rc* directory to view existing numbers. For example:

```
ls /sbin/rc3.d/S*
```

returns the following results:

```
/sbin/rc3.d/S10syslog
/sbin/rc3.d/S15nfs.server
/sbin/rc3.d/S21rfs
```

If your script is named **sybstart**, type a command like the following:

```
ln /sbin/init.d/sybstart /sbin/rc3.d/S99sybstart
```

In this example, you can use any number greater than 21 in place of 99.

For HP-UX

In HP-UX versions 10.0 and later, you cannot edit commands in the */etc/rc* file, so the System Administrator must create a script that executes when the operating system starts or shuts down.

For more information about **rc (1M)**, see the HP-UX manual page. You can model your script after the HP-UX template file */sbin/init.d/template*.

After creating the start-up and shutdown script, place it in the directory */sbin/init.d*. Execution scripts placed in this directory have symbolic links to directories */sbin/rcn.d* where *n* is the system run level. The linked scripts in */sbin/rcn.d* are used to control the sequencing order of the execution scripts.

For IBM RS/6000

On production systems, Adaptive Server can be restarted automatically whenever the UNIX operating system restarts. To have Adaptive Server restart automatically, place the **startserver** command in the */etc/inittab* file.

Following is a suggested format for an Adaptive Server start-up command to add to */etc/inittab*:

```
"sybase:2:wait:/release_directory/install/startserver -f \ RUN_servername
/dev/console 2>&1"
```

Where *release_directory* is the full path to the Sybase installation directory (specified as the SYBASE environment variable), and *RUN_servername* is the runserver file for the server you are starting.

Make sure that the entry for starting Adaptive Server *follows* any entry for */etc/rc.tcpip* and */etc/rc.nfs* in the */etc/inittab* file. All network resources must be available before you start Adaptive Server; otherwise Adaptive Server will not start. Also, your server will not start if the network is not up.

Slow start-up of network operations can also prevent Adaptive Server from starting, even when the commands in the */etc/rc.tcpip* file are in the correct order. You can designate a period of time for Adaptive Server to wait before starting by inserting a **sleep** command before the server start-up command in the runserver file. The **sleep** command is in the form:

```
sleep seconds_of_rest
```

Note To enable restart of a Backup Server when the operating system starts, add a Backup Server start-up command to */etc/inittab*. Use the path of the Backup Server runserver file in this command.

For Sun Solaris

Before you set Adaptive Server or Backup Server to start automatically, make sure that all network resources are available. Your server will not start if the network is not up. Be sure that the entry for restarting the server *follows* any commands in the *rc* directory for starting network operations. Slow start-up of network operations can prevent servers from starting, even if commands are in the correct order in the *rc* directory. You may want to designate a period of time for your server to wait before starting by inserting a **sleep** command at the beginning of the linked runserver file you created.

To configure your operating system for automatic restart of Adaptive Server and Backup Server:

1 Create a start-up script file with contents like the following:

```
$SYBASE/$SYBASE_ASE/install/startserver
   -f $SYBASE/$SYBASE_ASE/install/RUN_servername
```

where *$SYBASE/$SYBASE_ASE/install/startserver* is the full path name of the **startserver** utility, and *$SYBASE/$SYBASE_ASE/install/RUN_servername* is the full path name of the runserver file for the server.

2 Copy the script to the */etc/init.d* directory, using syntax like the following:

```
cp script_name /etc/init.d/script_name
```

3 Review the contents of the */etc/inittab* file to determine the default run level for your operating system. The **initdefault** entry specifies the default run level, which is usually 2 or 3.

4 Use the **ln** command to create a hard link from the runserver file to the appropriate *run control (rc)* directory, *rc#*, where *#* is the default run level obtained in step 2.

Use syntax like the following:

```
ln /etc/init.d/script_name
   /etc/rc#.d/S##script_name
```

When you type the command to create this link, add an uppercase “S” and a two-digit sequence number preceding the name of the script file. “S” indicates a start-up file. The sequence number is necessary because files in the *rc* directory are executed in order of their numbers. This file should be executed last, so you should use a sequence number that follows all existing numbers in ASCII order.

You can perform an **ls** command on the *rc* directory to view existing sequence numbers. For example:

```
ls /etc/rc3.d/S*
```

returns:

```
/etc/rc3.d/S10syslog
/etc/rc3.d/S15nfs.server
/etc/rc3.d/S21rfs
```

If your script is named **sybstart**, type the following:

```
ln /etc/init.d/sybstart /etc/rc3.d/S77sybstart
```

In this example, you can use any number greater than 21 in place of 77.

Stopping servers

Only the System Administrator has permission to issue a shutdown command. Using a shutdown command minimizes the amount of work that automatic recovery needs to do when the servers are restarted.

The preferred method of stopping Adaptive Server or Backup Server is to use the Transact-SQL **shutdown** command. For Monitor Server, use the **sms_shutdown** command.

Stopping Adaptive Server

To shut down Adaptive Server:

1. Use **isql** to log in to an Adaptive Server account with System Administrator privileges:

   ```
   isql -Usa -Ppassword -Sserver_name
   ```

2. Enter the following command to shut down the server:

   ```
   1> shutdown
   2> go
   ```

The default for the **shutdown** command is **with wait**. The **with wait** option allows Adaptive Server to finish executing SQL statements or procedures, perform a checkpoint in each database, disable new logins, and perform other shutdown tasks.

Issuing the **shutdown** command prints a message like this to the *sterr* file:

```
Server SHUTDOWN by request.The SQL Server is terminating
this process.
CT-LIBRARY error:
```

This is normal behavior. If the message indicates that Adaptive Server is waiting for processes to complete, and you need to stop Adaptive Server immediately, you can use the **shutdown with nowait** command. **shutdown with nowait** does not wait for currently executing statements to finish and does not perform checkpoints in every database.

Stopping Backup Server

To shut down a Backup Server:

1. Use **isql** to log in to a server with System Administrator privileges:

   ```
   isql -Usa -Ppassword -Sserver_name
   ```

2. Enter the following command to shut down the specified Backup Server:

   ```
   1> shutdown SYB_BACKUP
   2> go
   ```

After you shut down a Backup Server, you must wait at least 30 seconds before restarting it.

Issuing the **shutdown** command prints a message similar to the following to the *stderr* file:

```
Backup Server: 3.48.1.1: The Backup Server will go down
immediately.
Terminating sessions.
```

This is normal behavior. If a message indicates that Adaptive Server or Backup Server is waiting for processes to complete, and you need to stop Adaptive Server or Backup Server immediately, you can use the **shutdown with nowait** command. **shutdown with nowait** does not wait for currently executing statements to finish and does not perform checkpoints in every database.

Using **shutdown with nowait** for Backup Server can cause inconsistent or incomplete dumps and loads. Use this command only when necessary.

For more information on the **shutdown** command, see the *Adaptive Server Reference Manual*.

Stopping Monitor Server

You must shut down Monitor Server before restarting Adaptive Server. If Adaptive Server stops, make sure that Monitor Server is shut down before you restart Adaptive Server. Monitor Server must be shut down to release resources. Otherwise, Adaptive Server may not be able to allocate enough resources to restart.

If the Monitor Server heartbeat feature is in effect, Monitor Server automatically detects the termination of Adaptive Server within the specified period and shuts itself down. Therefore, before attempting to restart Adaptive Server after a shutdown, either wait for the automatic shutdown of Monitor Server to occur, or explicitly stop Monitor Server.

For more information on stopping Monitor Server, see the *Adaptive Server Enterprise Monitor Server User's Guide*.

Using the kill command

Warning! Use the kill command in Adaptive Server and Backup Server only as a last resort.

When possible, use the Transact-SQL **shutdown** or **shutdown with nowait** command. Do not use **kill** with the **-9** flag because it exits the server without running a checkpoint to ensure that all database changes are written to the database device. Adaptive Server may also exit without removing associated shared memory files and network handlers.

Because Adaptive Server and Backup Server are background processes, they can be killed from the operating system by their owner or by "root" with the UNIX **kill** command. The syntax is:

```
kill pid
```

where *pid* is the process identification of any **dataserver** or **backupserver** process, as determined by the **showserver** command. Killing one engine for a particular Adaptive Server kills all engines for that server.

If more than one Adaptive Server is running on the same system, you must be careful that the engine you kill is associated with the correct Adaptive Server. If your Adaptive Server is configured to use multiple engines (CPUs), each engine has an associated operating system process. The correct way to kill a multi-engine server is to specify the process ID for engine 0.

This `showserver` output shows the processes for a four-engine server:

```
showserver
```

UID	PID	PPID	C	STIME	TTY	TIME	COMD
jorge	3320	1	80	10:31:40	pts/4	302:15	dataserver -dteamster
jorge	3321	3320	80	10:31:45	pts/4	324:47	dataserver -ONLINE:1
jorge	3322	3320	80	10:31:45	pts/4	326:02	dataserver -ONLINE:2
jorge	3323	3320	80	10:31:45	pts/4	328:56	dataserver -ONLINE:3

This example shows four running **dataserver** processes with operating system process identifications (PID) 3320, 3321, 3322, and 3323. (**dataserver** is the executable form of the Adaptive Server program.)

Child engine processes for the dataserver have the **-ONLINE:** argument.

Each child engine has a parent process identification (PPID) that is equal to the process identification (PID) of the parent. In the example above, the PID of the parent server is 3320. The other three engines spawned by the parent process have the same PPID.

If the PPIDs appear to be unrelated, and there is more than one **dataserver** process, then more than one Adaptive Server is running on the system.

Shutdown and shared memory files

On Digital UNIX, HP-UX, and IBM RS/6000, when Adaptive Server starts, it creates the following files in *$SYBASE* to store information about the shared memory segments that it uses:

- *SERVER_NAME.krg* and
- *SERVER_NAME.srg0*.

On Sun Solaris, when Adaptive Server starts, it creates the *SERVER_NAME.krg* file in *$SYBASE* to store information about the shared memory segments that it uses. When Adaptive Server is shut down in a normal manner, the shared memory files are automatically removed. If Adaptive Server crashes or is stopped with the **kill -9** command, these files are not deleted. You need read and write permissions on these files to restart Adaptive Server after a crash or a **kill -9** command, because Adaptive Server must be able to overwrite the previously created shared memory files.

CHAPTER 3

Configuring the Operating System for Adaptive Server

This chapter discusses the operating system configuration settings that you can adjust after installing or upgrading Adaptive Server. Unless stated otherwise, the information pertains to all supported UNIX platforms.

Setting environment variables

It is crucial to the operation of Sybase products that the system environment variables are set correctly. Environment variables are set in the user's environment either interactively or by including them in the user's *.login* and *.cshrc* files (for C shell) or *.profile* file (for Bourne shell).

Only the file owner or the "root" user can edit a user's *.login*, *.cshrc*, or *.profile* file.

Table 3-1 describes the system environment variables for Adaptive Server.

To check the current value of environment variables on your system, at the operating system prompt enter:

```
setenv
```

Table 3-1: System environment variables

Variable	Sample value	Function
DSLISTEN	IGNATZ	Used at Adaptive Server start-up; enables the specified Adaptive Server to listen for input from front-end software.
DSQUERY	IGNATZ	Defines the Adaptive Server to which local client applications connect.
PATH	*/usr/bin:/etc:* */usr/sbin:* */usr/ucb:* */usr/bin/X11:/sbin*	Appends the full path to the Adaptive Server bin subdirectories.

DSLISTEN

The DSLISTEN environment variable defines the name used by Adaptive Server to listen for client connections, when no name is given during Adaptive Server start-up. If DSLISTEN is not set, and no name is given during start-up, the Adaptive Server name defaults to the server name given at installation.

DSQUERY

The DSQUERY environment variable defines the Adaptive Server name to which client programs attempt to connect when no Adaptive Server name is specified with a command-line option. If DSQUERY is not set, and you do not supply the Adaptive Server name with a command-line option, clients try to connect to the server name given at installation.

PATH

The PATH environment variable specifies which directory paths to search for executables. The Sybase executables are in the *bin* and *install* directories in the Sybase installation directory. The install program appends these paths to the current PATH environment variable.

Note The new executables are added at the *end* of the PATH variable. Verify that you do not have any old Sybase executables, instead of the new executables, in the PATH that will be invoked.

Using the stty setting

Setting the stty tostop option causes a background Adaptive Server to stop as soon as it tries to write to the terminal. To avoid this error, execute the following command before starting Adaptive Server:

```
stty -tostop
```

If you are redirecting all Adaptive Server output to files, you do not have to change the stty setting.

Restoring correct permissions

Sybase software files and directories are installed with the correct access permissions. If you notice that the permissions are no longer correct, you can restore the correct permissions with the script setperm_all, located in the *$SYBASE/install* directory.

File descriptors and user connections

The number of user connections used by Adaptive Server cannot exceed the number of file descriptors available to Adaptive Server on the operating system. When configuring user connections on Adaptive Server, the System Administrator should take into account the number of file descriptors available per process. Although most of the open file descriptors are available for user connections, a few are used by Adaptive Server for opening files and devices.

For Digital UNIX

The number of file descriptors per process is determined by the operating system parameter *open_max*. The default value of *open_max* is 4096. For more information on setting *open_max*, see the Digital UNIX operating system documentation.

Note Adaptive Server can use a maximum of 1024 file descriptors, regardless of the value of *open_max*.

To obtain the current value of the *open_max* parameter, use the Korn or Bourne shell ulimit command:

```
ulimit -n
```

To obtain the number of current file descriptors, use the sysconf or getdtablesize function.

For HP-UX

The kernel parameters maxfiles and maxfiles_lim control the number of file descriptors available to any one process.

For IBM RS/6000

The number of file descriptors per process is determined by the operating system parameter *open_max*. The default value of *open_max* is 32767. Adaptive Server can use a maximum of 2000 file descriptors per engine, regardless of the value of *open_max*. For more information on setting *open_max*, see the AIX operating system documentation.

To obtain the current value of the *open_max* parameter:

- Use the Korn or Bourne shell ulimit command:

```
ulimit -n
```

To obtain the number of current file descriptors programmatically:

- Use the sysconf or getdtablesize function.

For Sun Solaris

For Sun Solaris you can set both “soft” and “hard” limits for file descriptors. The soft limit can be increased up to the hard limit by the user, but the hard limit can be increased only by someone with “root” permissions. The soft limit determines the number of open file descriptors available to an Adaptive Server engine.

Although most of the open file descriptors are available for user connections, a few are used by Adaptive Server engines for opening files and devices.

See the *System Administration Guide* for additional information on user connections.

Displaying current soft and hard limits

To display the current soft limit:

- For C shells, enter:

```
limit descriptors
```

- For Bourne shells, enter:

```
ulimit -n
```

To display the current hard limit:

- For C shells, enter:

```
limit -h descriptors
```

- For Bourne shells, enter:

```
ulimit -Hn
```

Increasing the soft limit

To increase the soft limit:

- For C shells, enter:

  ```
  limit descriptors n
  ```

- For Bourne shells, enter:

  ```
  ulimit -Sn new_value
  ```

  where *n* is the current value for the soft limit, and *new_value* is the value to which you want to increase the soft limit.

Note The preceding commands can be used in your *runserver* file to increase the hard and soft limits. Because the *runserver* file is a Bourne shell script, be sure to use the Bourne shell versions of these commands in the *runserver* file.

Increasing the hard limit

To increase the hard limit, use a program like the sample program shown in "Sample program" on page 30.

To set up the sample program to increase the hard limit:

- Create the ASCII file *file_name.c* (where *file_name* is the name you give the file), by using an ASCII text editor. Type the text shown in the sample.
- Compile the file:

  ```
  cc file_name.c -o program_name
  ```

  where *file_name* is the name of the source file you created, and *program_name* is the name you want to give the program.

- Change the program's permissions and ownership so that it will execute as "root":

  ```
  chmod 755 program_name
  chown root program_name
  ```

  where *program_name* is the name of the compiled program.

- The "root" user can use the program to start Adaptive Server with increased user connections by typing the following command at the operating system prompt:

  ```
  # program_name dataserver -d master_device_name
  ```

where *program_name* is the name of the compiled program, and *master_device_name* is the full path of Adaptive Server’s master device. Instead of typing the command at the operating system prompt, you can add *program_name* preceding the dataserver command line in the Adaptive Server *runserver* file.

Sample program

The following example shows the source code that you can use to increase the hard limit:

```
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/types.h>
/*
** define MAX_CONNECTIONS to a number less than
** 10000. The number defined will then become the
** maximum number of connections allowed by an Adaptive
** Server.
*/
#define MAX_CONNECTIONS 9999
extern int errno;

main(argc,argv)
char **argv;
{
   struct rlimit rlp;
   uid_t uid;

   rlp.rlim_cur = MAX_CONNECTIONS;
   rlp.rlim_max = MAX_CONNECTIONS;
[nbsp  ]   /* set the number of open file desriptors to MAX_CONNECTIONS */
   if (setrlimit (RLIMIT_NOFILE,&rlp) == -1)
   {
      perror("setrlimit");
      exit(1);
   }

[nbsp  ]   /* reset the user id to disable superuser privileges */
   uid = getuid();
   setuid(uid);
[nbsp  ]   /* run the program indicated as arguments to this program */
   execv(*++argv, argv);
}
```

For additional information on user connections, see the *System Administration Guide*.

Adjusting the client connection timeout period

Adaptive Server uses the KEEPALIVE option of the TCP/IP protocol to detect clients that are no longer active. When a connection to a client is inactive for a period of time (the *timeout period*), the operating system sends KEEPALIVE packets at regular intervals. If it does not receive a response from the client machine for any of these packets, the operating system notifies Adaptive Server that the client is no longer responding. Adaptive Server then terminates the client's connection.

The KEEPALIVE default timeout period is 2 hours (7,200,000 ms.). To display the current time value, use the command for your platform as shown in the following sections.

For Digital UNIX

To display the current timeout value, enter:

```
/usr/sbin/ndd -get /dev/tcp tcp_keepalive_interval
```

To reduce the timeout period to 15 minutes (900,000 ms.), enter:

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 900000
```

For HP-UX

To display the current timeout period, enter:

```
/usr/contrib/bin/nettune -l
```

Note The switch is a lower case "l", not a numeral one (1).

The tcp_keepstart parameter specifies the length of time (measured in seconds) to keep an idle connection active before the system checks to see if the connection died.

To change the timeout period, use the nettune -s command.

For IBM RS/6000

To display the current timeout value, enter:

```
/usr/sbin/no -o tcp_keepidle
```

The tcp_keepidle parameter specifies the length of time (measured in half seconds) to keep an idle connection active before the system checks to see if the connection died. The default is 14,400 half seconds (7200 seconds, or 2 hours).

IBM recommends a value of at least 15 minutes.

For Sun Solaris

To display the timeout value, enter:

```
/usr/sbin/ndd -get /dev/tcp tcp_keepalive_interval
```

To reduce the timeout period to 15 minutes (900,000 ms.), enter:

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 900000
```

Checking for hardware errors

The following types of hardware error messages indicate problems that may lead to database corruption:

- Disk read, write, or retry errors
- Timeouts
- System panics
- Memory problems of any type

For Digital UNIX

Check the */var/adm/messages* file on a regular basis. See the operating system documentation for more information on this file.

For HP-UX

Check the */var/adm/syslog/syslog.log* file on a regular basis. You can view the file directly, or you can use the HP-UX dmesg command. See your operating system documentation for more information.

For IBM RS/6000

The errpt command includes several flags for limiting the report to events that match certain criteria. Use the errpt command on a regular basis. If errors appear, use the diagnostic tool diag to check your memory and disks. Or use the System Management Interface Tool (SMIT) to run the errpt command. This command may produce a lot of output.

For Sun Solaris

Check the */var/adm/messages* file on a regular basis. If any of the types of hardware errors described in the beginning of this section appear, use the Sun Microsystems diagnostic tool, sundiag, to check memory and disks. See the operating system documentation for more information.

Monitoring the use of operating system resources

The *Performance and Tuning Guide* and the *System Administration Guide* discuss maintaining the optimal number of Adaptive Server engines for your workload and system configuration. To determine the optimal number, you need to monitor system and CPU usage.

For Digital UNIX

The following tools help monitor performance:

- The iostat command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.

- The ps command provides an accurate snapshot of accumulated CPU time and usage for individual processes. This can be very helpful in determining the load of the data server, engine, and processes.
- The time command can be useful in determining the various user, system, and real-time resources used over a complete run.

For details about these tools, see your operating system documentation.

For HP-UX

HP-UX supplies many tools to help monitor performance, which include:

- The sar command reports relative and absolute I/O throughput rates to each disk and controller.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.
- The ps command provides a snapshot of accumulated CPU time and usage for individual processes.
- The time command can be useful in determining the various user, system, and real-time resources used over a complete run.

For details about these tools, see your operating system documentation.

For IBM RS/6000

IBM RS/6000 supplies the following tools for monitoring performance:

- The iostat command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.
  - netstart -v displays Transmit/Receive Statistics. It is also used to determine if enough mbufs have been configured for network traffic.
  - no -a displays current network options. It is also used for tuning mbuf pools.
- The ps command provides a snapshot of accumulated CPU time and usage for individual processes.

- The time command determines the various user, system, and real-time resources during a complete run.

For details about these tools, see your operating system documentation.

For Sun Solaris

Sun Solaris supplies the following tools to help monitor performance:

- The iostat command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.
- The ps command gives you an accurate snapshot of accumulated CPU time and usage for individual processes. This can be very helpful in determining the data server-, engine-, and process-specific loading.
- The time command can be useful in determining the various user, system, and real-time resources used over a complete run.

For details about these tools, see your operating system documentation.

A sample C shell maintenance script

The following sample C shell script calls several isql scripts:

```
#!/bin/csh -f
if ( -e dbcc_mail.out)
then
    rm dbcc_mail.out
endif
foreach i (*.dbcc)
    isql -Usa -Ppassword < $i > dbcc_out
    if ( ’grep -c ’Msg 25[0-9][0-9]’ dbcc_out’ )
        then
        echo "There are errors in" $i >> dbcc_mail.out
        cat dbcc_out >> dbcc_mail.out
    else
        echo "Backing up " $i:r >> dbcc_mail.out
        isql -Usa -Ppassword < $i:r.backup
    endif
end
```

```
mail -s "Backup Report" jjones < dbcc_mail.out
```

The first set of scripts (one for each database with a file name appended with *.dbcc*) runs dbcc checkalloc and dbcc checkdb for each database and sends the messages to an output file called *dbcc_out*.

For example, the script master.dbcc runs dbcc to check the master database:

```
dbcc checkalloc (master)
go
dbcc checkdb (master)
go
```

The C shell script then runs the grep command to find 2500-level error messages in the dbcc output. The results of the grep command go into an output file called *dbcc_mail.out*.

Next, the script invokes an isql backup script for each database for which no 2500-level errors occurred and adds the “Backing up *database_name*” line to *dbcc_mail.out* . For example, the script master.backup backs up the master database:

```
use master
go
dump database master to master_dump
go
```

You may want to add appropriate dump transaction commands to your scripts.

If there are 2500-level error messages, the script does not back up the database. At the end of the script, *dbcc_mail.out* is mailed to the System Administrator “jjones,” who then has a record of fatal dbcc errors and successful backups.

You can tailor the sample shell and isql scripts to suit the needs of your installation.

To have the scripts execute automatically, edit the *crontab* file, and add an entry similar to this:

```
00 02 * * * /usr/u/sybase/dbcc_ck 2>&1
```

In this example, a C shell script called dbcc_ck executes daily at 2:00 a.m.

CHAPTER 4

Setting Up Communications Across a Network

This chapter provides information about how to set up communications across the network. Adaptive Server uses a configuration file called the “interfaces” file to provide the information Adaptive Server needs to allow servers and clients to communicate. On UNIX platforms the interfaces file is named *interfaces*. This chapter describes the interfaces file and provides specific configuration information for each supported UNIX platform.

Adaptive Servers and clients use different interfaces files. Interfaces files contain entries for all Adaptive Servers, backup servers, and other server products on the network.

Table 4-1 shows where to find more information on server and client interfaces file tasks and topics.

Table 4-1: Where to find interfaces file tasks and topics

Type of interfaces file	Task or topic	See
UNIX server or client	Adding entries for multiple Adaptive Server installations	Chapter 5, “Customizing Localization for Adaptive Server”
	Creating a master interfaces file for multiple installations	“Creating one interfaces file for multiple installations” on page 48

Type of interfaces file	Task or topic	See
	Configuring for multiple networks	"Configuring interfaces files for multiple networks" on page 49
	Reference information	"Understanding the format of the interfaces file" on page 42
PC-client	Configuring a client	*Installing Adaptive Server and OmniConnect on UNIX Platforms*
	Reference information and instructions for advanced tasks	*Open Client Supplement* for your PC client platform, or the appropriate Open Client documentation
Client platforms not listed	Configuring, reference information, and instructions for advanced tasks	*Open Client Supplement* for your PC client platform, or the appropriate Open Client documentation

What is an interfaces file?

An interfaces file contains information about the network locations of servers.

In the Sybase client/server environment, a client can connect with Adaptive Server if it knows where the server resides on the network and if the server supports the client's language or character set. When a client initiates a connection, it looks in its interfaces file for the network location of the target server.

The interfaces file lists the name and address of every server, including Backup Server. When you are using a client program, and you want to connect with a particular server, the client program looks up the server name in the interfaces file and connects to that server. You can also supply the name of the server by using the DSQUERY environment variable.

Servers also need network information. When a server starts up, it looks in its interfaces file to determine where to listen for client connection requests. In addition, Adaptive Server can take on a client role when it makes remote procedure calls to other Adaptive Servers.

Creating an interfaces file

The installation program, srvbuild, automatically creates an interfaces file for each server and client installation. The default location for the interfaces files is the Sybase installation directory (as determined by the value of the SYBASE environment variable).

Contents of an interfaces file

An interfaces file contains network information about all servers on your network, including Adaptive Server, Backup Server, XP Server, plus any other server applications such as Adaptive Server Enterprise Monitor™ Server, Replication Server® and Open Server.

The network information in the file includes the server name, network name or address of the host machine, and the port, object, or socket number (depending on the network protocol) on which the server listens for queries. See "Understanding the format of the interfaces file" on page 42 for the specific makeup of the interfaces file entry.

Each entry in an interfaces file can contain two types of lines:

- Master lines, which are used by server applications to listen for queries over the network. This information is called a *listener service*.
- Query lines, which are used by client applications to connect to servers over the network. This information is called a *query service*.

The network information contained in the master and query lines for a server is identical because a server listens for connection requests on the same port that clients use to request connections.

Differences between client and server interfaces files

A server needs both master and query lines in its interfaces file, since servers sometimes act as clients to other servers.

A client's interfaces file does not need a master line. It functions correctly with only a query line.

If your site has multiple installations

If you have more than one Adaptive Server installation, each server's interfaces file should contain information about all servers on the network.

If all of your server products are running on the same platform, you can create one master *interfaces* file and copy that file to each machine. For more information, see "Creating one interfaces file for multiple installations" on page 48.

If the host machine supports multiple networks, see "Configuring interfaces files for multiple networks" on page 49.

How Adaptive Server uses an interfaces file

Adaptive Server uses its interfaces file to listen for clients. To listen for queries, the server:

- Determines its own name by checking the value of the DSLISTEN environment variable
- Looks in the interfaces file for an entry whose name matches the DSLISTEN name and finds the corresponding master line
- Uses the network information provided by the master line to listen for queries

How a client uses an interfaces file

When a client, or a server acting as a client, connects to a server it:

- Determines the name of the server either programmatically or by referring to the DSQUERY environment variable. If the application user has not set DSQUERY, the runtime value for the server name defaults to the SYBASE environment variable.
- Looks in the interfaces file for an entry whose name matches the name of the server, stops at the first instance of that name, and reads the query line.
- Uses the network information provided by the query line to connect to the server. If the client cannot connect the first time, it makes additional attempts according to the delay and retry numbers indicated in the interfaces file. If no matching entry is found, an error message is written to the client’s standard error file. If multiple networks are supported, the client attempts to connect using the information in the second query entry for the server. See “Configuring for query port backup” on page 53.

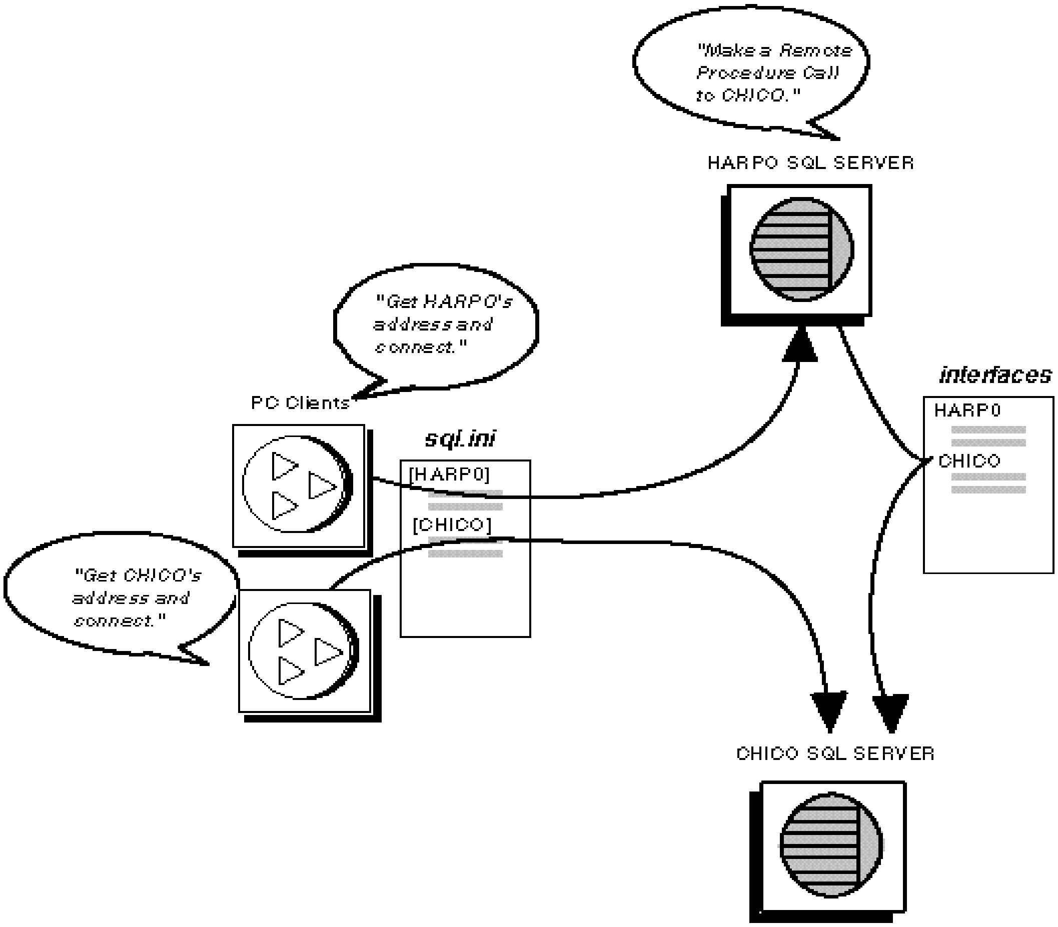
The Open Client documentation discusses client connections in much greater detail. See the *Open/Client Programmer’s Supplement* for your client platform or the appropriate Open/Client documentation.

Heterogeneous and homogeneous environments

You can run Adaptive Server and clients on the same platform or on different platforms.

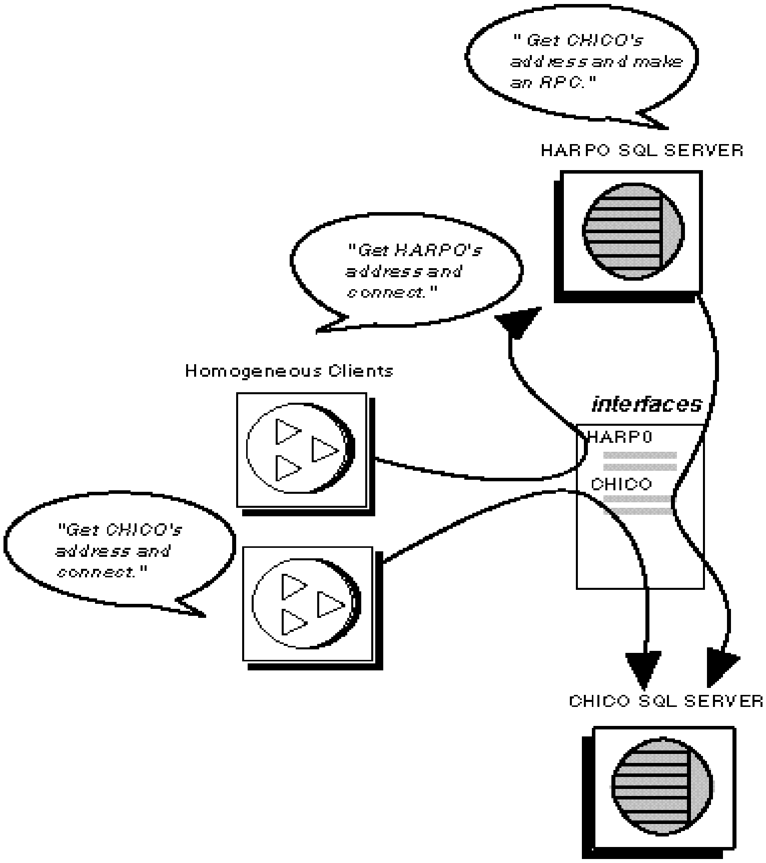
If the platforms are different, each platform may require a different format and configuration for its *interfaces* file. Figure 4-1 illustrates how a client PC uses network information in its interfaces file (*sql.ini*) to connect to Adaptive Server running under UNIX, and how Adaptive Server uses its *interfaces* file to connect to another server during a remote procedure call.

Figure 4-1: Establishing network connections in a heterogeneous environment


"Make a Remote Procedure Call to CHICO."
HARPO SQL SERVER
"Get HARPO's address and connect."
PC Clients
sql.ini
[HARPO]
[CHICO]
interfaces
HARPO
CHICO
"Get CHICO's address and connect."
CHICO SQL SERVER

If both a client and a server are running under UNIX, a copy of the same interfaces file is valid for both. Figure 4-2 illustrates how clients and Adaptive Servers running in a homogeneous environment can use copies of the interfaces file to establish connections. Because the two Adaptive Servers are running under the same operating system, they can use the same interfaces file or exact copies of the same file.

Figure 4-2: Establishing network connections in a homogeneous environment


" Get CHICO's address and make an RPC."
HARPO SQL SERVER
"Get HARPO's address and connect."
Homogeneous Clients
interfaces
HARPO
CHICO
"Get CHICO's address and connect."
CHICO SQL SERVER

Understanding the format of the interfaces file

When an Adaptive Server is installed, srvbuild creates an entry for it in the interfaces file.

Components of an interfaces file entry

Table 4-2 describes the components of an interfaces file entry.

Table 4-2: Components of an interfaces file

Component	Value
servername	Name of Adaptive Server or Backup Server. Requirements for a server name are: • The name cannot be more than 30 characters long. • The first character must be a letter (ASCII a through z, A through Z). • The characters that follow must be letters, numbers, or underscores (_).
retry_attempts (optional)	Number of times you want the client to try to connect to a server after initial failure. Default is 0.
delay_interval (optional)	Number of seconds between connection attempts. Default is 0.
service_type	Type of service defined by entry. Must be one of the following: • master (see "Contents of an interfaces file" on page 39) • query (see "Contents of an interfaces file" on page 39) • debug (connection for use by Sybase Technical Support only)
api	Application programming interface available to the network. The available interface file for UnixWare is TLI (Transport Layer Interface), represented by the letters "tli".
protocol	Name of the network protocol. Protocol types available are: • TCP/IP, represented by the letters "tcp" • SPX, represented by the letters "spx"
network	Name of the network; not currently used by Adaptive Server. srvbuild enters "ether" as a placeholder.
host	Network name or address of server's host machine. • For TCP/IP, use either the host name or Internet address. Maximum size of entry is 32 bytes. To determine the host name of a machine, log in to that machine and enter: `/bin/hostname`
machine	Network name or address of server's host machine. • You can use either the host name or Internet address. Maximum size of entry is 32 bytes. To determine the host name of a machine, log in to that machine and enter: `/bin/hostname`
device	The network device endpoint. • For TCP networks, it varies according to the vendor supplying the networking software. Check the vendor-specific documentation for the name of the device. Your network may provide multiple streams devices corresponding to different protocols in the TCP protocol suite. Choose the TCP streams device. Common TCP streams devices include */dev/tcp, /dev/ito,* and */dev/inet/tcp*. • For the IPX protocol, the value is always */dev/nspx.*

Component	Value
address for a TCP protocol entry	Address consists of: • Address prefix, “\x” for tli. • Network type, always 0002. • Port number converted to four digits, hexadecimal. Must be between 1025 and 65535. Must be unique. Check the */etc/services* file on each machine on the network to see what port numbers are in use. Enter the Adaptive Server port number in a new section of */etc/services* labeled “Sybase specific services”. You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using the port number. • IP network node address of the host machine converted to eight digits, hexadecimal. • Trailing zeros, optional, 16 digits.
address for an SPX protocol entry	Address consists of: • Address prefix, “\x” for tli • Network address, 8 digits • Node address, 12 digits • Socket number, 4 digits
port/object number	• For TCP/IP networks, port is a unique port number between 1025 and 65535. Check the */etc/services* file on each machine on the network to see what port numbers are in use. Enter the Adaptive Server port number in a new section of */etc/services* labeled “Sybase specific services”. You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using that number. • For DECnet networks, *object* is an object number between 128 and 253. The number must be unique to the node on which the server is running.
port	A unique port number between 1025 and 65535. Check the */etc/services* file on each machine on the network to see what port numbers are in use. Enter the Adaptive Server port number in a new section of */etc/services* labeled “Sybase specific services”. You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using that port number.

Format by platform

The following rules apply to the format of interfaces file entries:

- Each Adaptive Server has only one entry, although there may be multiple lines in the entry.

- Each line that follows the *servername* line must begin with a space or a character tab.
- Each element on the line must be separated by a single space.
- Each entry is separated by a blank line.
- You can add comments to an interfaces file by adding a pound sign (#) at the beginning of the line and a line break at the end.

Digital UNIX

```
servername retry_attempts delay_interval<newline>
 <tab>service_type protocol network host port/object_number<newline>
```

HP-UX

```
servername retry_attempts delay_interval<newline>
 <tab>service_type protocol network machine port<newline>
```

IBM RS/6000

```
servername retry_attempts delay_interval<newline>
 <tab>service_type protocol network machine port<newline>
```

Sun Solaris

```
servername retry_attempts delay_interval<newline>
 <tab>service_type api protocol device address<newline>
```

Creating interfaces files for SPX

If Novell Protocol Suite Version 3.0.1 is installed on your machine, you can create interfaces file entries to support the SPX/IPX protocol. Interfaces file entries for SPX have the following format:

```
servername retry_attempts delay_interval<newline>
 <tab>service_type api protocol device address<newline>
```

Note Each entry line that follows the *servername* line *must* begin with a <tab> character or a blank space. Use a single space between each component on an entry line.

The *servername*, *retry_attempts*, *delay_interval*, and *service_type* components have the same meaning in an SPX entry as in a TCP entry. Table 4-3 describes the other components.

Table 4-3: Remaining components in the interfaces file

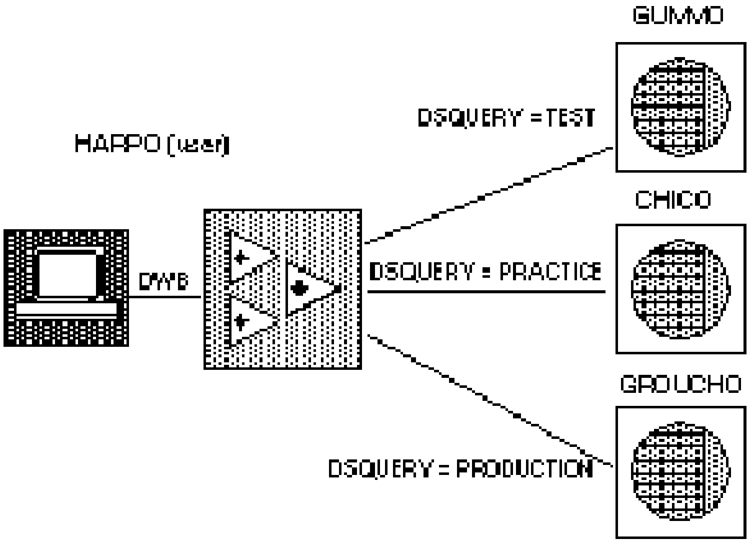
Component	Value
api	*api* is the application programming interface to the network. Since Adaptive Server uses the Transport Layer Interface (TLI) for SPX protocol connections, use the letters "tli".
protocol	*protocol* is the name of the network protocol. Use the letters "spx" to designate the SPX protocol.
device	*device* is the network device endpoint supplied for SPX networks. The device that is supplied with the Novell Protocol Suite network software is named */dev/nspx*.

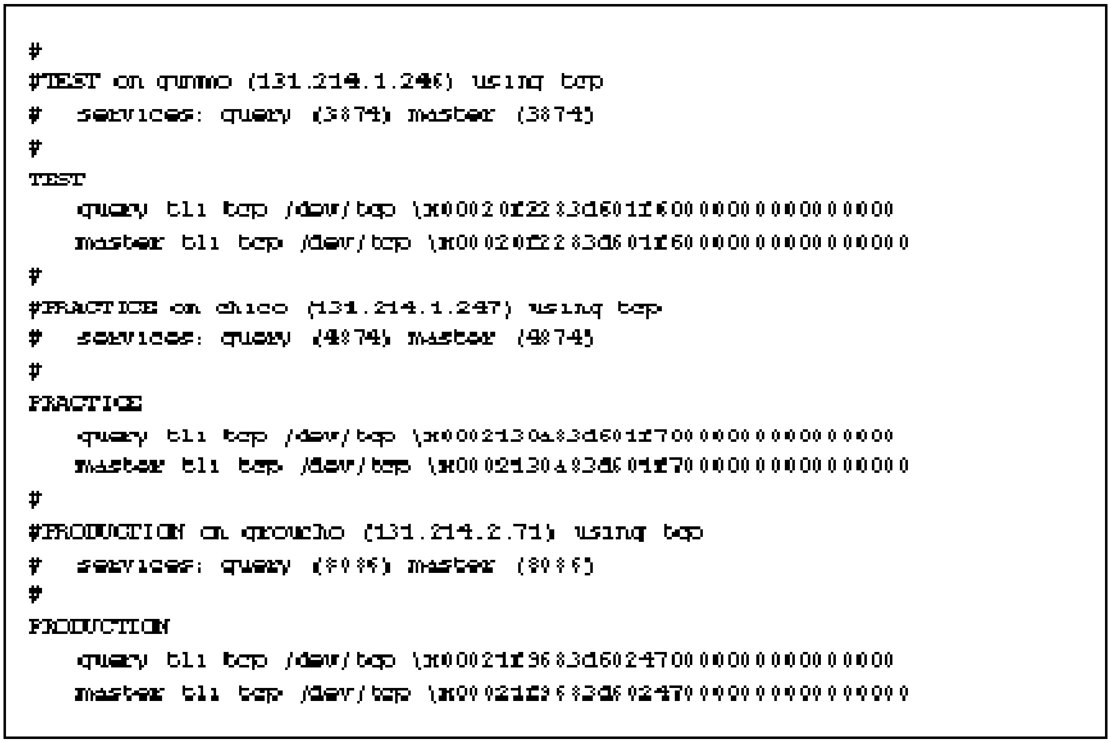
address	*address* is a composite entry generated from the network number, node, and socket number, defined as follows:
	• The first four bytes of the address represent the network number converted to hexadecimal.
	• The next six bytes represent the network node of the machine running Adaptive Server.
	• Both the network number and node are assigned by the network administrator of your system. You can check the */usr/netware/npsconfig* file to determine the network and node numbers for your machine.
	• The socket number is a hexadecimal number between 0x8000 and 0xFFFF that identifies the process on the machine.
	If you choose a socket number that is already in use on the machine where your Adaptive Server is located, you receive a message similar to the following when you attempt to restart your Adaptive Server:
	`ninit: t_bind, Incorrect permissions`
	`ninitconn_free: t_close, fd=5, Illegal transport fd`
	`ninit: All master network listeners have failed.`
	`Shutting down.`
	`ueshutdown: exiting`
	More than one Adaptive Server can have the same socket number, if each server is on a different machine. The *combination* of network number, node, and socket uniquely identifies the Adaptive Server.
	For example, if your machine has a network number of 4D160D14 and a node number of 000000000001, and you use socket number 86bb, the following address entry is added:
	`/x4D160D1400000000000186bb`

Interfaces file example

Figure 4-3 illustrates an environment with three UNIX Adaptive Servers and one UNIX client. The interfaces file shown is a master interfaces file that works for each server and the client on the four machines shown.

Figure 4-3: Interfaces file example – three servers, one client





```
#
#TEST on gummo (131.214.1.246) using tcp
#   services: query (3874) master (3874)
#
TEST
    query tli tcp /dev/tcp \x00020f2283d601f60000000000000000
    master tli tcp /dev/tcp \x00020f2283d601f60000000000000000
#
#PRACTICE on chico (131.214.1.247) using tcp
#   services: query (4874) master (4874)
#
PRACTICE
    query tli tcp /dev/tcp \x0002130a83d601f70000000000000000
    master tli tcp /dev/tcp \x0002130a83d601f70000000000000000
#
#PRODUCTION on groucho (131.214.2.71) using tcp
#   services: query (8086) master (8086)
#
PRODUCTION
    query tli tcp /dev/tcp \x00021f9683d602470000000000000000
    master tli tcp /dev/tcp \x00021f9683d602470000000000000000
```

Utilities for editing the interfaces file

Use the following Sybase utilities to edit the network information in the interfaces file:

- dsedit – an X-Windows GUI utility.
- dscp – a UNIX command-line utility.

For details on using these utilities, see the *Utilities Guide for UNIX Platforms*.

Creating one interfaces file for multiple installations

Distributing copies of one interfaces file (a master file) with entries for all Adaptive Servers is the easiest way to maintain consistency in the interfaces files in a homogeneous environment on a network.

You make all changes to one version of the file and then copy the updated master file to all appropriate Sybase directories.

You can make a master file in one of two ways:

- Using dsedit or dscp
- Using a text editor

Using dsedit or dscp to create a master interfaces file

You can use the dsedit or dscp utility to create a master interfaces file, which you can then distribute to all servers. If you are not an experienced Sybase user, you may find that using dsedit or dscp is easier than using a text editor. Using dsedit or dscp also ensures that your interfaces file is consistent in format.

To create a master interfaces file with dsedit or dscp:

1. Select the interfaces file that contains the most complete, up-to-date information.
2. Begin a dsedit or dscp session in your latest Sybase installation to edit this interfaces file.
3. Add entries for any Adaptive Servers or Backup Servers that are not listed in this file.

For details on using these utilities, see the *Utilities Guide for UNIX Platforms*.

Using a text editor to create a master interfaces file

To construct a single master interfaces file from several individual interfaces files:

1. Concatenate the individual interfaces files.
2. Make a copy of the file.
3. Use an ASCII text editor to modify the copy of the concatenated file.

Note When you manually edit an interfaces file, be sure that, for each entry, each line following the first line begins with a <tab> character.

The following elements must be correct and unique in the resulting file:

- *servername*– each server entry in the interfaces file must be unique. During the srvbuild session, you had the choice of entering a server name or accepting the default server name, SYBASE, for those servers. If you accepted the default name, find any duplicate SYBASE entries in your merged file, and rename them.
- A combination of the host machine’s network name or address and Adaptive Server’s port or object number.
- If the original interfaces file was created when there was only one machine on the network, its entries may have the word “loghost” in place of the machine name (address). If *loghost* is present, replace it with the machine name.

Configuring interfaces files for multiple networks

On some platforms, Adaptive Server can accommodate multiple networks. This allows Adaptive Server to listen for clients over multiple network interfaces. You must add an entry for each network interface to the interfaces file.

Configuring the server for multiple network handlers

To configure multiple network listeners:

1. Define a unique host name for each network interface in your operating system’s host database.

2 In your interfaces file, use a text editor to add copies of the “master” line for your Adaptive Server, one for each additional interface you want the server to listen on.

3 Include a unique host name on each line to configure a network handler for each network interface.

4 Port numbers within the interface need not be the same, but they can be. They fall under the same rules for naming and numeric range as the primary network interface.

Sample interfaces files for multiple network handlers

The following example shows an interfaces file for an Adaptive Server with two network interfaces, each connected to a different network. The server host machine’s network address is 131.214.1.246 on Network 1 and 131.214.1.247 on Network 2.

```
# PRODUCTION server with two network listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tli tcp /dev/tcp \x0002130a83d601f60000000000000000
 <tab>master tli tcp /dev/tcp \x0002130a83d601f70000000000000000
 <tab>query tli tcp /dev/tcp \x0002130a83d601f60000000000000000
```

The following example shows an interfaces file for an Adaptive Server with two network interfaces, each connected to a different network. The server host machine is connected to a TCP/IP network, on which it is called SERV_CORPNET, and a DECnet network, on which it is called SERV_ENGNET.

```
# PRODUCTION server with two network listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tcp ether SERV_CORPNET 6464
 <tab>master decnet ether SERV_ENGNET 150
 <tab>query tcp ether zeppo 6464
```

The following example shows an interfaces file for an Adaptive Server with two network interfaces. The server host machine is known as SERV_CORPNET on the corporate network and SERV_ENGNET on the engineering network.

```
# PRODUCTION server with two network listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tcp ether SERV_CORPNET 4559
 <tab>master tcp ether SERV_ENGNET 5479
 <tab>query tcp ether SERV_CORPNET 4559
```

The next time Adaptive Server is restarted, it spawns a network handler process for each master line in the entry that corresponds to the server's DSLISTEN value. Connections made on each interface will be handled equally, as peers.

Configuring the client connections

When an Adaptive Server client scans the interfaces file for a server name, the client uses the first "query" entry it encounters for the server's entry. This makes configuring clients to use multiple network connections less straightforward than configuring the server ports. You have two choices:

- Use different DSQUERY names for the clients. The interfaces files on all the machines are the same, but they contain multiple DSQUERY names.
- Use the same DSQUERY name for all clients. The interfaces files on the different machines contain different network names.

Using one network-independent DSQUERY name

If uniform client DSQUERY naming is important, you can mask the necessary difference in the network addressing of the clients in the interfaces file. You can install separate Sybase installation directories and distinct interfaces files on client file servers on each network to allow users to connect to the correct network address. Instead of altering the DSQUERY name the clients use, you maintain one DSQUERY name for all clients, on all networks, and alter each network's interfaces file accordingly.

This method assumes that:

- You have complete control over what the Sybase installation clients see on each network.
- The interfaces file (at the very least) is *not* shared or copied among Sybase installations on different networks.

The interfaces file would look like the following example on the "engineering" network:

```
#PRODUCTION server on network2
 PRODUCTION<tab>3<tab>3<newline>
 <tab>query tli tcp /dev/tcp \x0002130a83d601f70000000000000000
 <tab>master tli tcp /dev/tcp \x0002130a83d601f60000000000000000
 <tab>master tli tcp /dev/tcp \x0002130a83d601f70000000000000000#PRODUCTION
server network2<newline>
 PRODUCTION<tab>3<tab>3<newline>
```

```
<tab>query decnet ether CHICO 150
<tab>master tcp ether ZEPPO 6464
<tab>master decnet ether CHICO 150# PRODUCTION server network2<newline>
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_ENGNET 5470
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

The “query” line in each file name is different, depending on the network to be used.

The full “master” entry is present in both files. This is allowed because only Adaptive Server will use the “master” lines. Assuming that the server host machine can see both networks (both host names are interchangeable), it does not matter which interfaces file is used for Adaptive Server start-up.

Using different DSQUERY names

To use different DSQUERY names for each network listener:

1. Choose an additional server name.

   You can concatenate the original server name and the network name. For example, if your server is named PRODUCTION, you could choose the names PRODUCTION_network1 and PRODUCTION_network2.

2. Do one of the following:

   - For PC clients, use sqledit to create multiple *sql.ini* file entries for the server, one for each network. In the following example, you create one entry for PRODUCTION_network 1 and one for PRODUCTION_network2. For more information, see the Open Client documentation for your client platform.

   - For UNIX clients, you can edit the *interfaces* files with an ASCII text editor. From the server’s *interfaces* files, copy the server name line and the “master” line for each network into the client interfaces file. Add the appropriate server name for each entry, and change “master” to “query.”

     Clients on each network must use the DSQUERY value that corresponds to the network the client is on. In the following example, either PRODUCTION_network1 or PRODUCTION_network2 can be used.

```
# Client entry for PRODUCTION on network1
 PRODUCTION_network1<tab>3<tab>3<newline>
 <tab>query tli tcp /dev/tcp \x0002130a83d601f60000000000000000
 # Client entry for PRODUCTION on network2
 PRODUCTION_network2<tab>3<tab>3<newline>
 <tab>query tli tcp /dev/tcp \x0002130a83d601f70000000000000000#
Client entry for PRODUCTION on network1
 PRODUCTION_network1<tab>3<tab>3<newline>
 <tab>query tcp ether zeppo 6464
 # Client entry for PRODUCTION on network2
 PRODUCTION_network2<tab>3<tab>3<newline>
 <tab>query decnet ether chico 150# Client entry for PRODUCTION on
network1
 PRODUCTION_network1<tab>3<tab>3<newline>
 <tab>query tcp ether serv_corpnet 4559
 # Client entry for PRODUCTION on network2
 PRODUCTION_network2<tab>3<tab>3<newline>
 <tab>query tcp ether serv_engnet 5479
```

Configuring for query port backup

Another use for multiple network interfaces is to provide a backup in case of network failure. If a client is connected to a server via two networks, the client can establish a connection via the second network if the first one goes down.

To configure Adaptive Server for query port backup:

1. Install multiple "master" and "query" lines in a server entry in the interfaces file.
2. Adaptive Server listens for connections at both ports. Clients looking for a host name and a port number for a connection to an Adaptive Server will try the port on each "query" line in order, until they establish a connection.

   The following example shows how to configure a backup network that will be used only if the normal connection fails. The second network is named "network2".

```
# PRODUCTION server with two network listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tli tcp /dev/tcp \x0002130a83d601f60000000000000000
 <tab>master tli tcp /dev/tcp \x0002130a83d601f70000000000000000
 <tab>query tli tcp /dev/tcp \x0002130a83d601f60000000000000000
 <tab>query tli tcp /dev/tcp
\x0002130a83d601f70000000000000000#PRODUCTION server with two network
listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>query tcp ether ZEPPO 6464
 <tab>query decnet ether CHICO 150
 <tab>master tcp ether ZEPPO 6464
 <tab>master decnet ether CHICO 150# PRODUCTION server with two network
listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tcp ether SERV_CORPNET 4559
 <tab>master tcp ether SERV_ENGNET 5479
 <tab>query tcp ether SERV_CORPNET 4559
 <tab>query tcp ether SERV_ENGNET 5479
```

3 Configure PC-client interfaces files with the appropriate multiple “query” entries, as described in the Open Client documentation. For client interfaces files in a homogeneous environment, you can copy the entire interfaces file entry for the Adaptive Server into the client interfaces file.

4 A connection on the secondary port occurs only if the corporate network is disabled, or if the corporate network interface on the host machine fails or is shut down due to a network-related failure.

Troubleshooting

This section describes how to correct some common situations that may cause a server to not start.

Server fails to start

If a server fails to start, the port number specified in the interfaces file may be in use.

❖ **To investigate the port assignment**

1 Look in the interfaces file to identify the port number assigned to the server.

2 Determine whether another process is using the same port number by entering:

```
netstat -a
```

If the port number is presented as a local address in the netstat output, you cannot use that port for the server. Another process is already using that port.

3 To verify that the server port is in use, start the server manually.

The server will not start if its assigned port number is already in use.

For information on starting servers manually, see the installation documentation for your platform and *Utility Programs for UNIX Platforms*.

❖ **If a stale serverprocess is retaining use of the port number**

1. Do one of the following:

   - Use the operating system kill command to terminate the process.
   - Use another port number for the server by modifying the interfaces file.

2. Start the server manually to confirm that the port number is available.

   For information on starting servers manually, see the installation documentation for your platform and *Utility Programs for UNIX Platforms*.

3. If the server that failed to start was an XP Server, kill the server that you started manually, then start Adaptive Server.

   XP Server should start automatically.

Error when executing an ESP

If you attempt to execute an ESP (extended stored procedure), you may see the following error:

```
00:00000:00008:1997/09/10 12:52:53.03 kernel  XP Server failed to start. Try
bringing up XP Server manually. Check SQL Server documentation for more
information on how to bring XP Server up.
```

XP Server could not start because the port number is in use by another process. Use the procedures described in the previous section to determine if the port number specified for XP Server is in use.

If you find no processes using the same port number:

1. Restart Adaptive Server.
2. Execute the ESP that you attempted earlier.

   XP Server should start automatically.

CHAPTER 5

Customizing Localization for Adaptive Server

This chapter discusses Sybase localization support for international installations, including information about configuring languages, character sets, and sort order. It also includes information for reconfiguring localization. For more information, see the *System Administration Guide*.

Overview of localization support

Localization is the process of setting up an application to run in a particular language or country environment, including translated system messages and correct formats for date, time, and currency. Adaptive Server supports localization for international customers and for customers with heterogeneous environments.

This support includes:

- Data processing support – Adaptive Server comes with character-set and sort-order definition files it uses to process the characters used in different languages.

  Sybase provides support for the major languages in:

  - Western Europe
  - Eastern Europe
  - Middle East

    - Latin America
    - Asia
- Translated system messages are available in:
    - Brazilian Portuguese
    - Chinese (Simplified)
    - French
    - German
    - Japanese
    - Korean
    - Spanish
- Translated documentation is available in:
    - Chinese (Simplified)
    - French
    - German
    - Japanese
    - Korean
    - Spanish

Language modules

Adaptive Server stores its localized software messages in separate language modules, each of which includes the following:

- Character sets
- Sort orders
- Messages

When you install a language module, the installation program loads the messages, character-set, and sort-order files that support the new language in the correct locations.

Note The Adaptive Server language modules do not contain error messages for utilities such as bcp, isq, and defncopy. These error messages are contained in the Client-Library™ language modules. For information on running these utilities with a language other than U.S. English or a character set other than ISO 8859-1, see *Utility Programs for UNIX Platforms*.

When you install Adaptive Server and Backup Server, system messages in English are installed by default.

Character-set conversion

Backup Server passes messages to Adaptive Server in the client's language and in the Adaptive Server character set. Adaptive Server then converts the messages and issues them in the client's language and character set. Keep the following requirements in mind when selecting a character set:

- In a heterogeneous environment, Adaptive Server and Backup Server may need to communicate with clients running on different platforms and using different character sets. To maintain data integrity, the server converts the code between the character sets.
- To use the built-in conversion, you need to install the character-set definition files on the server for all the character sets being used by your clients. Built-in conversion support is available for many character sets.
- Unilib™ conversion support is available for all character sets supported by Sybase. To enable Unilib conversion, you must use sp_configure and turn enable unicode conversions on. See the *System Administration Guide* for more information.

If either Adaptive Server or Backup Server does not support a client's language or character set, that server issues a warning message. Errors also occur when the Backup Server character set is not compatible with the Adaptive Server character set.

Character-set conversion is supported only between character sets for the same language or between character sets in the same language group.

For example, automatic character-set conversion is supported between the character sets for the Western European languages: ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252, ISO 8859-1, ISO 8859-15, Macintosh Roman and ROMAN8. Similarly, conversion is supported between the character sets for Japanese: CP 932, EUC-JIS, Shift-JIS, and DEC-Kanji.

However, code conversion is not supported between any of the Western European language character sets and the Japanese character sets. For more information about supported conversions, see the *System Administration Guide*.

Conversions between server and client

If Adaptive Server does not support the client's language or character set, the client can connect with the server, but no character conversions will occur.

When a localized client application connects to Adaptive Server, the server checks to see if it supports the client's language and character set.

- If Adaptive Server supports the language, it automatically performs all character-set conversions and displays its messages in the client's language and character set.
- If Adaptive Server does not support the language, it uses the user's default language or Adaptive Server's default language.
- If Adaptive Server does not support the character set, it issues a warning to the client, turns conversion off, and sets the language to U.S. English.

Sort orders

Each character set comes with one or more sort orders (collating sequences), which are located in the sort-order definition files (*.srt* files). These files accompany the character-set definition files and can be found in the same directory.

You can select a sort order for your data according to the needs at your site. However, keep in mind that the server can support only one sort order at a time, so select a sort order that will work for all of your clients.

Warning! Make all changes to the default character set and sort order for a new Adaptive Server before creating any user databases or making any changes to the Sybase-supplied databases. Changing the character set and sort order after data or data structures have been added to Adaptive Server may cause incorrect behavior. To change the character set or sort order after you have added data, see the *System Administration Guide*.

Available sort orders

The sort order determines the collating sequence Adaptive Server uses to order, compare, and index character data. Each character set comes with one or more sort orders.

Sort orders are located in sort order definition files (*.srt* files) that accompany your character-set definition files.

Note Available sort orders vary according to the character set installed on Adaptive Server.

You can see the available sort orders for your character set by looking in the *.srt* file for your language. Sort orders are stored in the following path:

$SYBASE/charsets/<charset_name>/.srt*

For more information about localization files, see “Localization directories” on page 68.

Table 5-1 describes the sort orders that you can specify at installation time or at a later time using the sqlloc utility.

Table 5-1: Sort orders available in Adaptive Server

Sort order name	Description
Binary order	Sorts all data according to numeric byte values for that character set. Binary order sorts all ASCII uppercase letters before lowercase letters. Accented or ideographic (multibyte) characters sort in their respective standards order, which may be arbitrary. All character sets have binary order as the default. If binary order does not meet your needs, you can specify one of the other sort orders either at installation or at a later time by using the sqlloc utility.

Sort order name	**Description**
Dictionary order case sensitive, accent sensitive	Case sensitive. Sorts each uppercase letter before its lowercase counterpart, including accented characters. Recognizes the various accented forms of a letter and sorts them after the associated unaccented letter.
Dictionary order case insensitive, accent sensitive	Case-insensitive dictionary sort order. Uppercase letters are equivalent to their lowercase counterparts and are intermingled in sorting results.
Dictionary order case insensitive accent insensitive	Case-insensitive dictionary sort order. Diacritical marks are ignored.
Dictionary order case insensitive with preference	Case-insensitive dictionary sort order, with case preference for collating purposes. A word written with uppercase letters is equivalent to the same word written with lowercase letters. Uppercase and lowercase letters are distinguished only when you use an order by clause. The order by clause sorts uppercase letters before it sorts lowercase. **Note** Do not select this sort order unless your installation requires that uppercase letters be sorted before lowercase letters in otherwise equivalent strings for order by clauses. Using this sort order may reduce performance in large tables when the columns specified in an order by clause match the key of the table's clustered index.
Alternate dictionary order case sensitive	Case-sensitive alternate dictionary sort order with lowercase variants sorted before uppercase. Use with several of the Western European languages.
Alternate dictionary order case insensitive accent insensitive	Case-insensitive and accent-insensitive alternate dictionary sort order. Use with several of the Western European languages.
Alternate dictionary order case insensitive uppercase preference	Case-insensitive alternate dictionary sort order with uppercase preference. Use with several of the Western European languages.

Use sqlloc to display the available sort orders for the character sets you plan to use.

Gathering information about collation

The following functions enable you to perform collation operations in your applications:

- sortkey – generates binary values that you can use to order query results based on collation behavior.

- compare – compares two character strings based on alternate collation rules.

These functions can be nested and used anywhere that an expression is allowed, including stored procedures and triggers.

Ordering query results based on collation behavior

The sortkey function generates values that can be used to order results based on collation behavior. This allows you to work with character collation behaviors beyond the default set of Latin-character-based dictionary sort orders and case or accent sensitivity. The return value is a *varbinary* datatype value that contains coded collation information for the input string that is retained from the sortkey function.

For example, you can store the values returned by sortkey in a column with the source character string. When you want to retrieve the character data in the desired order, the select statement only needs to include an order by clause on the columns that contain the results of running sortkey.

The sortkey function guarantees that the values it returns for a given set of collation criteria work for the binary comparisons that are performed on varbinary datatypes.

Syntax for sortkey

The syntax is:

sortkey (*char_expression* [, {*collation_name* | *collation_ID* }])

where:

- *char_expression* is one of the following:
  - Character type (*char*, *varchar*, *nchar*, or *nvarchar*)
  - Character variable, or
  - Constant character expression, enclosed in single or double quotation marks
- *collation_name* is a quoted string or a character variable that specifies the collation to use.
- *collation_ID* is an integer constant or a variable that specifies the collation to use.

There are two types of collation tables, built-in and external. You can use either the collation name or the collation ID to specify a built-in table. You must use the collation name to specify external tables. Table 5-2 lists the valid values for *collation_name* and *collation_ID.*

Table 5-2: Collation names and IDs

Description	Collation name	Collation ID
Binary sort	binary	50
Default Unicode multilingual	default	0
CP 850 Alternative: no accent	altnoacc	39
CP 850 Alternative: lower case first	altdict	45
CP 850 Alternative: no case preference	altnocsp	46
CP 850 Scandinavian dictionary	scandict	47
CP 850 Scandinavian no case preference	scannocp	48
GB Pinyin	gbpinyin	n/a
Latin-1 English, French, German dictionary	dict	51
Latin-1 English, French, German no case	nocase	52
Latin-1 English, French, German no case preference	nocasep	53
Latin-1 English, French, German no accent	noaccent	54
Latin-1 Spanish dictionary	espdict	55
Latin-1 Spanish no case	espnocs	56
Latin-1 Spanish no accent	espnoac	57
ISO 8859-5 Cyrillic dictionary	cyrdict	n/a
ISO 8859-5 Russian dictionary	rusdict	n/a
ISO 8859-9 Turkish dictionary	turdict	n/a
Shift-JIS binary order	sjisbin	n/a
Thai dictionary	thaidict	1

Rules for using sortkey

The following rules apply to using the sortkey function:

- *char_expression* must be composed of characters that are encoded in the server's default character set.
- *char_expression* can be an empty string. If it is an empty string:
  - sortkey returns a zero-length varbinary value, and
  - Adaptive Server stores a blank for the empty string.

  An empty string has a different collation value than an NULL string from a database column.
- If *char_expression* is NULL, sortkey returns a NULL value.

- If you do not specify a value for *collation_name* or *collation_ID*, sortkey assumes binary collation.

Note sortkey can generate up to 6 bytes of collation information for each input character. Therefore, the result from using sortkey may exceed the 255-byte length limit of the varbinary datatype. If this happens, the result is truncated to fit. Truncation removes result bytes for each input character until the result string is less than 255 bytes. If this occurs, a warning message is issued, but the query or transaction that contained the sortkey function continues to work.

Comparing strings based on alternate collation rules

The compare function allows you to directly compare two character strings based on alternate collation rules.

Syntax for compare

The syntax is:

```
compare (char_expression1, char_expression2
[, {collation_name | collation_ID}])
```

where:

- *char_expression1* is the character expression you want to compare to *char_expression2*.

  char_expression2 is the character expression against which you want to compare *char_expression1*.

  char_expression1 and *char_expression2* can be one of:

  - Character type (*char*, *varchar*, *nchar*, or *nvarchar*)
  - Character variable
  - Constant character expression, enclosed in single or double quotation marks

- *collation_name* can be a quoted string or a character variable that specifies the collation to use.
- *collation_ID* is an integer constant or a variable that specifies the collation to use.

The compare function returns the following values, based on the collation rules that you chose:

- 1 – indicates that *char_expression1* is greater than *char_expression2*.

- 0 – indicates that *char_expression1* is equal to *char_expression2*.
- -1 – indicates that *char_expression1* is less than *char_expression2*.

Rules for using compare

The following rules apply to using the compare function:

- Both *char_expression1* and *char_expression2* must be characters that are encoded in the server's default character set.
- Either *char_expression1* or *char_expression2*, or both, can be empty strings:
  - If *char_expression2* is empty, the function returns 1.
  - If both strings are empty, then they are equal, and the function returns a 0 value.
  - If *char_expression1* is empty, the function returns a -1.

  The compare function does not equate empty strings and strings containing only spaces, as Adaptive Server does. compare uses the sortkey function to generate collation keys for comparison. Therefore, a truly empty string, a string with one space, or a string with two spaces do not compare equally.
- If either *char_expression1* or *char_expression2* is NULL, then the result will be NULL.
- If you do not specify a value for *collation_name*, compare assumes binary collation.
- If you do not specify a value for *collation_ID*, compare assumes binary collation.

Language modules

If you want Adaptive Server error messages to be displayed in a language other than U.S. English (us_english), you must install the appropriate language module.

When you install a new language module, installation automatically loads the language, character set and sort order files into the Sybase installation directory to support the new language. For information about directories, see "Localization directories" on page 68.

Installing a new language module

A full install of Adaptive Server installs all the language components automatically. If you did not select a full install, you need to install additional language modules manually.

To install a new language module:

1. Load the language module software from the distribution media. You must load this software into the same directory in which you loaded Adaptive Server.
2. Reconfigure the language and, if necessary, the character set and sort order for Adaptive Server. For instructions, see “Changing the localization configuration” on page 71.

Message languages

For messages, U.S. English is installed as the default language in Adaptive Server. The following rules apply to language modules:

- During Adaptive Server installation or reconfiguration, you can specify a default language other than U.S. English. However, you must have installed the language module for the language you specify.
- If your clients require Adaptive Server messages in a language other than U.S. English, you must load the language module for those languages. Then, you can configure Adaptive Server to the language used by your clients.
- If Adaptive Server does not support messages in a client’s language, these clients receive messages in the Server default language.

  For example, if your client’s language is Latin, the Spanish language module is installed, and Spanish is specified as the Adaptive Server default language, the client receives messages in Spanish.

Localization

By default, the Adaptive Server and Backup Server configurations use the English locale settings, which include:

- Character-set definition files for Western European character sets

- Sort-order definition files for Western European character sets
- U.S. English system message files

During the installation process or through reconfiguration, you can specify a different language, character set, and sort order.

Localization directories

Sybase localization configuration involves the following directories:

- *locales*
- *charsets*

The table below illustrates the structure of the localization files. It does not show a complete list of all the files.

%SYBASE% or *$SYBASE/*	*charsets*	*charset_name*	**.srt* files
		charset_name...	*charset.loc*
		unicode	**.uct* files
	locales	*language_name*	*charset_name*
		language_name...	*charset_name...*
		locales.dat	
		message	*language_name*
			language_name...

About the locales directory

The *$SYBASE/locales* directory contains a subdirectory for each available language. Each language subdirectory contains a subdirectory for each character set available with that language.

- The *.loc* files in these subdirectories enable Adaptive Server or Backup Server to report errors in a specific language, encoded in a specific character set.

  There are a variety of *.loc* files in each subdirectory. Most of these files contain translated error messages for a specific product or utility.

- The *common.loc* file in each subdirectory contains localized information, such as local date, time, and currency formatting, that is used by all products.
- The *locales.dat* file contains entries that associate platform-specific locale names with Sybase language and character-set combinations.

About the charsets directory

The files in *$SYBASE/charsets/charset_name* contain information related to each particular character set, such as the definition of the character set and any sort orders available for that character set.

About the locales.dat file

You can edit the *locales.dat* file to:

- Change the default language or character set for a platform, or
- Add new associations between platform locale names and Sybase language and character-set names.

Format of locales.dat file entries

Each entry in the *locales.dat* file links a platform-specific locale definition to a Sybase language and character-set combination. Each entry has the following format:

```
locale = platform_locale, syb_language, syb_charset
```

where:

- *platform_locale* is the platform-specific keyword for a locale. For acceptable values, see your operating system documentation.

  When the locale being defined is the default for the site, *platform_locale* is “default”.
- *syb_language* is the name of the language directory to be used from within *$SYBASE/locales/language_name*.
- *syb_charset* is the character-set name that determines the character-set conversion method and identifies the directory location of the message files for clients from within *$SYBASE/locales/charset_name*.

For example, the following entry specifies that the default locale uses us_english for the language and iso_1 for the character set:

```
locale = default, us_english, iso_1
```

How client applications use locales.dat

Client applications use the *locales.dat* file to identify the language and character set to use. The connection process follows these steps:

1. When a client application starts, it checks the operating system locale setting and then checks the *locales.date* file to see if that setting is appropriate for Adaptive Server. For example, a locale entry for French can look like the following:

   ```
   locale = fr_FR, french, iso_1
   ```

2. When the client connects to Adaptive Server, the language and character-set information is passed to Adaptive Server in the login record.
3. Adaptive Server then uses:
   - The character-set information, for example, iso_1, to identify the client's character set and verify whether it can convert character data to this character set
   - The language (in the preceding example, French) and character-set information to see if it has messages in the client's language

Note Adaptive Server software comes with some locale entries predefined in the *locales.dat* file. If these entries do not meet your needs, you can either modify them or add new locale entries.

Editing the locales.dat file

Note Before making any changes, create a backup copy of the original *locales.dat* file.

To edit *locales.dat*:

1. Open *locales.dat* in an ASCII text editor such as vi or notepad.
2. Find the appropriate section for your platform:
   - Digital UNIX – **[axposf]**
   - HP-UX – **[hp ux]**
   - IBM RS/6000 – **[aix]**
   - Sun Solaris – **[sun_svr4]**
3. Make sure the section contains an entry for the language (*syb_language*) and character-set (*syb_charset*) combination that you want to use. If no entry exists, create one. If the existing entry is not correct, modify it. Save the changes and exit the text editor.

For example, if you want your Open Client messages to appear in French, and Adaptive Server is using the ROMAN8 character set, look for the following entry:

```
locale = fr_FR, french, roman8
```

Note The value for *platform_locale* must match the value required by your operating system. If the locales definitions in your system configuration files do not match the Sybase locale definitions, your applications will not run properly.

Changing the localization configuration

By default, the Adaptive Server and Backup Server configurations use the English locale settings localization, which include:

- Character-set definition files for Western European character sets
- Sort order definition files for Western European character sets
- us_english system message files

During the installation process and through reconfiguration, you can specify a different language, character set, and sort order.

For Adaptive Server

Each language uses approximately 2MB of database space per module. If necessary, use the alter database command to increase the size of the master database before adding another language.

Note If you want to install more than one language on Adaptive Server, and the master database is not large enough to manage more than one language, the transaction log may become too full. You can expand the master database only on the master device. For more information, see the *System Administration Guide*.

1 To configure localization for Adaptive Server on the server, start sqlloc:

```
$SYBASE/$SYBASE_ASE/bin/sqlloc
```

2 Select Localize an exiting server.

3 From the Adaptive Server selection window, select the server.

4 Supply the user name and password. The user must have "sa" privileges.

5 Supply the following:

- Default language
- Default character set
- Default sort order

6 Select any other languages you want to install. You may only select languages that are supported by the default character set.

The Add and Remove Languages window lists all Sybase supported languages.

7 The Localization Summary window summarizes the configuration options you selected. Click OK to confirm your selections.

The Status Output window notifies you upon completing the installation.

For Backup Server

When you install Backup Server, you specify the language and default character set specified for Adaptive Server. Your system messages appear in the default language, English.

Configuring Adaptive Server for non-default character sets

When you select a Backup Server to configure, Server Config displays the Configure Backup Server dialog box.

To configure Adaptive Server with the character set and sort order for your language:

1 Use the charset utility to load the default character set and sort order:

```
$SYBASE/SYBASE_ASE/bin/charset -Usa -Ppassword
-Sserver_name  sort_order_file character_set
```

To use charset, the server must be running and you must have System Administrator privileges.

Replace *sort_order_file* with the name of the sort order file. See Table 5-3 on page 73. Replace *character_set* with the Sybase name for your character set. See Table 5-4 on page 75.

2 Use charset to load additional character sets. For details, see “Using the charset utility” on page 76.

If you plan to use the Adaptive Server built-in character-set conversions, you must load the character-set definition files for all the characters set on your client platforms. If you are using the Unilib character-set conversions, you do not need to do this.

3 Using isql, log in to your server as “sa” and select the master database.

```
1> use master
2> go
```

4 Configure your server for the new character set and sort order:

```
1> sp_configure “default sort_order_id”,
2> sort_order_id, “character_set”
3> go
```

Replace *sort_order_id* with the ID for your sort order. See Table 5-3 on page 73. Replace *character_set* with the Sybase name for your character set. See Table 5-4 on page 75.

5 Shut down the server to start the reconfiguration process.

6 Use your normal process on your UNIX system to reboot the server, usually by invoking one of the *RUN_xxx* scripts from *$SYBASE/$SYBASE_ASE/install*.

7 The server boots, rebuilds all the system indexes, then shuts down. Reboot a second time to bring the server up in a stable state.

Sort orders

Table 5-3 describes the available sort orders. If your language does not appear, then there is no language-specific sort order for your language—use a binary sort order.

Table 5-3: Available sort orders

Language or script	Sort orders	File name	ID
All languages	Binary order	*binary.srt*	50
Cyrillic	Dictionary order, case sensitive, accent sensitive	*cyrdict.srt*	63
	Dictionary order, case sensitive, accent sensitive	*cyrnocs.srt*	64

Language or script	Sort orders	File name	ID
English French German (These sort orders work with all Western European character sets.)	Dictionary order, case sensitive, accent sensitive	*dictiona.srt*	51
	Dictionary order, case insensitive, accent sensitive	*nocase.srt*	52
	Dictionary order, case sensitive, accent sensitive, with preference	*nocasepr.srt*	53
	Dictionary order, case insensitive, accent insensitive	*noaccent.srt*	54
English French German (These sort orders work only with CP 850.)	Alternate dictionary order, case sensitive	*altdict.srt*	45
	Alternate dictionary order, case sensitive, accent insensitive	*altnoacc.srt*	39
	Alternate dictionary order, case sensitive, with preference	*altnocsp.srt*	46
Greek (This sort order works only with ISO 8859-7.)	Dictionary order, case sensitive, accent sensitive	*elldict.srt*	65
Hungarian (These sort orders work only with ISO 8859-2.)	Dictionary order, case sensitive, accent sensitive	*hundict.srt*	69
	Dictionary order, case insensitive, accent sensitive	*hunnoac.srt*	70
	Dictionary order, case insensitive, accent insensitive	*hunnocs.srt*	71
Russian (This sort order works with all Cyrillic character sets except CP 855.)	Dictionary order, case sensitive, accent sensitive	*rusdict.srt*	58
	Dictionary order, case insensitive, accent sensitive	*rusnocs.srt*	59
Scandinavian (These sort orders work only with CP 850.)	Dictionary order, case sensitive, accent sensitive	*scandict.srt*	47
	Dictionary order, case insensitive, with preference	*scannocp.srt*	48
Spanish	Dictionary order, case sensitive, accent sensitive	*espdict.srt*	55
	Dictionary order, case insensitive, accent sensitive	*espnocs.srt*	56
	Dictionary order, case insensitive, accent insensitive	*espnoac.srt*	57
Thai	Dictionary order	*dictionary.srt*	51
Turkish (These sort orders work only with ISO 8859-9.)	Dictionary order, case sensitive, accent sensitive	*turdict.srt*	72
	Dictionary order, case insensitive, accent insensitive	*turnoac.srt*	73
	Dictionary order, case insensitive, accent sensitive	*turnocs.srt*	74

Character sets

Table 5-4 lists the supported character sets and their Sybase name.

Table 5-4: Sybase character-set names

Character sets	Sybase name
ASCII 8	acsii_8
Big 5	big5
CP 437	cp437
CP 850	cp850
CP 852	cp852
CP 855	cp855
CP 857	cp857
CP 860	cp860
CP 863	cp863
CP 864	cp864
CP 866	cp866
CP 869	cp869
CP 874	cp874
CP 932	cp932
CP 936	cp936
CP 950	cp950
CP 1250	cp1250
CP 1251	cp1251
CP 1252	cp1252
CP 1253	cp1253
CP 1254	cp1254
CP 1255	cp1255
CP 1256	cp1256
CP 1257	cp1257
CP 1258	cp1258
DEC Kanji	deckanji
EUC-CNS	euccns
EUC-GB	eucgb
EUC-JIS	eucjis
EUC-KSC	eucksc
GREEK8	greek8
ISO 8859-1	iso_1
ISO 8859-2	iso88592
ISO 8859-5	iso88595
ISO 8859-6	iso88596
ISO 8859-7	iso88597

Character sets	Sybase name
ISO 8859-8	iso88598
ISO 8859-9	iso88599
ISO 8859-15	iso885915
Koi8	koi8
Macintosh Cyrillic	mac_cyr
Macintosh Central European	mac_ee
Macintosh Greek	macgrk2
Macintosh Roman	mac
Macintosh Turkish	macturk
ROMAN8	roman8
Shift-JIS	sjis
TIS 620	tis620
TURKISH8	turkish8
UTF-8	utf8

Using the charset utility

Use the charset utility to load character sets and sort orders into Adaptive Server. Use charset to load the default character set and sort order at installation time only.

To change the default character set and sort order of Adaptive Server, see the *System Administration Guide*.

Syntax

```
charset [sort_order_file | charset.loc] charset_directory
```

Usage

```
charset
[ -U username ]
[ -P password ]
[ -S server ]
[ -I interfaces ]
[ -v version ]
[sort_order_file | charset.loc ]
[ charset_directory ]
```

Table 5-5: Keywords and options for charsets

Keywords and options	Description
-U	If you are not already logged in to your operating system as “sa”, you must specify “-Usa” or “/username = sa” in the command line.
-P	Specifies the “sa” password on the command line. If not specified, the user is prompted for the “sa” password.

Keywords and options	Description
-S	Specifies the name of the server. If not specified, charset uses the DSQUERY environment variable to identify the server name. If DSQUERY is not set, charset attempts to connect to a server named “SYBASE.”
-I	Specifies the interfaces file to use. If not specified, charset uses the interfaces file in the SYBASE directory.
-v	Prints the Sybase version string, then exits. Do not use with other options.
sort_order_file | *charset.loc*	When charset is used to load the default character set and sort order, *sort_order_file* is a mandatory parameter specifying the name of the sort order file to be used by Adaptive Server. When loading additional character sets, use *charset.loc* to indicate the name of the character-set files.
charset_directory	Specifies the directory of the character set to be used by Adaptive Server.

CHAPTER 6

Licensing Adaptive Server Features

This chapter describes how to add new licenses to existing servers using Sybase Software Asset Management (SySAM).

What is SySAM?

SySAM is a licensing mechanism that:

- Provides System Administrators the ability to monitor their site's use of Sybase products and optional features, and
- Grants access to select Sybase products and features.

Adaptive Server 12.0 uses SySAM to license optional features.

Adaptive Server features licensed through SySAM

Table 6-1 describes the Adaptive Server features that are licensed using SySAM.

Table 6-1: Licensed Adaptive Server features

Feature name	License name	Description
Adaptive Server	ASE_SERVER	The basic Adaptive Server Enterprise product, without optional features.
High Availability	ASE_HA	Adaptive Server failover capabilities for high availability environments.

Feature name	License name	Description
Distributed Transaction Management	ASE_DTM	Distributed transaction management features, including the Adaptive Server version 12.0 XA interface.
Java in Adaptive Server	ASE_JAVA	Java and XML classes in Adaptive Server databases.
Advanced Security Mechanisms	ASE_ASM	Network-based authentication and encryption using DCE and CyberSafe.

How does SySAM work?

SySAM "checks out" licensed features to users, and checks them back in when they are no longer needed. The basic components of SySAM are:

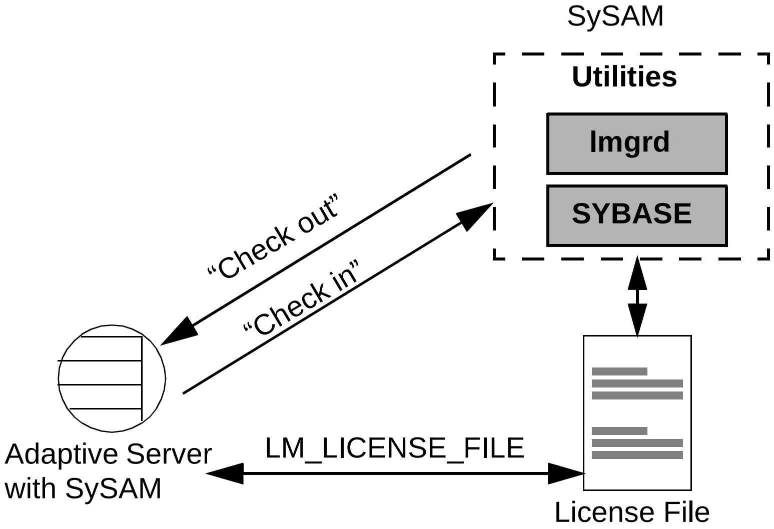
- One or more Adaptive Server products with SySAM support
- One or more license files
- Asset management software, which consists of a license management daemon and a Sybase daemon

Adaptive Server check-out procedure

When you boot Adaptive Server 12.0, it looks for a license file named *$SYBASE/$SYBASE_SYSAM/licenses/license.dat*. If the file does not exist, Adaptive Server looks for the file specified in the environment variable LM_LICENSE_FILE.

The license file contains a pointer to the SySAM software, or to a remote machine on which SySAM is running. SySAM consists of two daemons—the license management daemon, lmgrd, and the SYBASE daemon—and utilities to manage licensing activities, lmutil. The daemons handle requests to check in or check out licensed features, as shown in Figure 6-1.

Figure 6-1: SySAM components



Using information in the license file, Adaptive Server connects to SySAM and attempts to check out a license for the basic Adaptive Server product, ASE_SERVER. If the ASE_SERVER license is checked out successfully, Adaptive Server continues to boot and attempts to check out any optional features enabled in *server_name.cfg*.

If Adaptive Server is configured to use optional features, such as distributed transaction management or high availability, it attempts to check out licenses for those features during the boot-up process. If a license is unavailable for an optional feature, Adaptive Server still starts, but the feature cannot be used.

Types of SySAM systems

You can arrange the basic components of SySAM for a single standalone Adaptive Server, or for a collection of servers on a network.

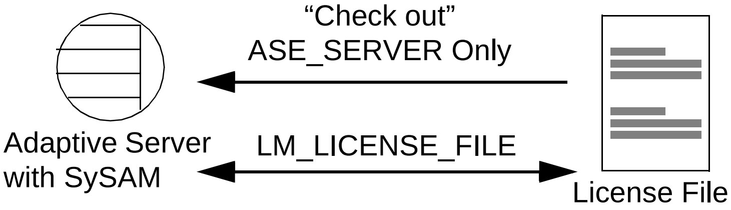
Standalone system

In standalone systems, SySAM components reside on the same computer that runs Adaptive Server. Standalone SySAM systems can license either Adaptive Server alone or Adaptive Server with optional features.

Standalone system with no optional features

In a standalone installation where no optional Adaptive Server features are used, you do not need to run the asset management software. See Figure 6-2. In this system, the license file contains all license information for ASE_SERVER. Adaptive Server obtains the basic license from the local license file and *does not* attempt to contact the license daemon software. No optional features can be used in this configuration.

Figure 6-2: Standalone system without optional features



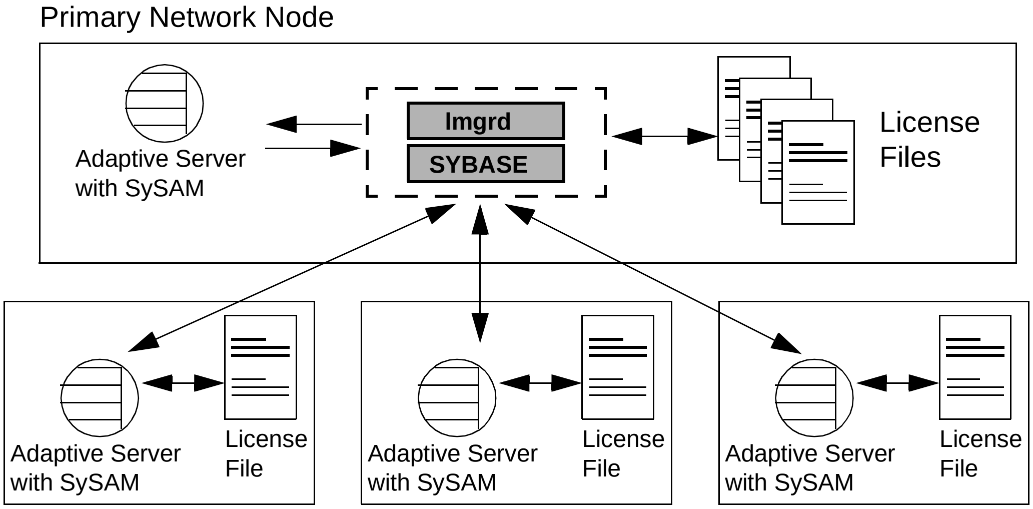
Standalone system with optional features

In a standalone system where you have licensed optional Adaptive Server features, *all* of the SySAM components shown in Figure 6-1 on page 81 reside on a single machine. The license file points directly to the asset management software, which runs on the same machine as Adaptive Server.

Network system

In a network system, a single machine called the **primary network node** runs SySAM. The software on the primary network node handles check in/check out requests from all other Adaptive Server instances in the system, as shown in Figure 6-3.

Figure 6-3: Network system


Primary Network Node
Adaptive Server with SySAM
lmgrd
SYBASE
License Files
Adaptive Server with SySAM
License File
Adaptive Server with SySAM
License File
Adaptive Server with SySAM
License File

In a network system, the license files for remote Adaptive Server installations contain only pointers to the primary network node. Servers check out individual licenses using the license files and asset management software that reside on the primary network node.

Setting up Adaptive Server with SySAM

The steps required to enable SySAM to track the functionality of Adaptive Server depend on the type of asset management system you want to use and on the Adaptive Server features you have purchased. Table 6-2 guides you through the setup process.

Table 6-2: Steps for setting up Adaptive Server for use with SySAM

If you want to use:	And you have purchased:	See:
Standalone asset management	• Adaptive Server basic product, but • No additional licensed features	Enabling SySAM software
	• Adaptive Server basic product, and • One or more licensed Adaptive Server features	Enabling SySAM software and Starting SySAM
Network asset management	• Adaptive Server basic product, with or without additional, licensed features	Enabling SySAM software and Starting SySAM

Enabling SySAM software

SySAM is installed with Adaptive Server.

If you licensed the Sybase Adaptive Server and other registered components when you installed them, SySAM is already enabled.

You must manually enable SySAM if:

- You did not register licensed products during Adaptive Server installation, or
- You add additional licensed products after Adaptive Server is installed.

Starting SySAM

If you are using a networked software asset management system, you must run the software on the primary network node before installing additional servers.

If you did not use SySAM to license Sybase server products during the unload process, use one of the following methods to start the asset management software manually:

- Use the UNIX command:

```
$SYBASE/$SYBASE_SYSAM/bin/lmgrd -c $LM_LICENSE_FILE &
```

- Run the *startd.sh* script in the *$SYBASE/$SYBASE_SYSAM/bin* directory.

Verifying that SySAM is running

When the lmgrd daemon starts, it automatically starts up the SYBASE daemon.

To verify that the license management software is running on the system, enter:

```
$SYBASE/SYSAM-1_0/bin/lmutil lmstat -c
```

Make sure that both lmgrd and SYBASE are running before you continue the installation or start Adaptive Server.

Adding servers in a network asset management system

To prepare for installing additional Adaptive Servers with SySAM in a network asset management system:

1. Make sure you have installed Adaptive Server and the asset management software on the primary network node. See Chapter 3, “Unloading Server Products from Distribution Media” in the installation guide for your platform.

2. Make sure that the asset management software is running on the primary network node. See “Starting SySAM” on page 84.
3. Make sure you have the printed Sybase License Certificates that contain your Adaptive Server and options key information.
4. Collect the following information:
   - The host name of the primary network node
   - An unused port number on the primary network server node
   - The remote directory of the network server node on which the asset management software has been installed

Install the additional servers as instructed in the Adaptive Server installation documentation.

Adding a license for an existing server

If you did not license Adaptive Server at the time of installation, you can add a license for it.

To add a license for a basic Adaptive Server installation:

1. Make sure the lmgrd and Sybase daemons are running:

   ```
   $SYBASE/$SYBASE_SYSAM/bin/lmutil lmstat -c
   ```

2. If the license daemon is not running, start lmgrd:

   ```
   lmgrd -c $SYBASE/$SYBASE_SYSAM/license.dat \
       -l $SYBASE$SYBASE_SYSAM/log/lmgrd.log&
   ```

3. Run lmgr:

   ```
   $SYBASE/SYSAM-1_0/bin/lmgrd
   ```

4. Provide the license information:
   - Order Number
   - Feature Name
   - Feature Count
   - Software Version
   - Authorization Code

5 After the license has been added, run the license management utility lmutil lmreread in the *$SYBASE/SYSAM-1_0/bin* directory:

```
lmutil lmreread
```

The new license is appended to the license file.

Adding licences for new features after installation

After completing the initial installation of Adaptive Server, if you install additional components that require licenses, you need to update the license information.

To add additional licenses:

1 Log on to the primary network node (server) that hosts the SySAM services.

2 Make sure the lmgrd and Sybase daemons are running:

```
$SYBASE/$SYBASE_SYSAM/bin/lmutil lmstat -c
```

3 If the license daemon is not running, manually start lmgrd:

```
$SYBASE/$SYBASE_SYSAM/bin/lmgrd -c $SYBASE/$SYSAM/licenses/license.dat
    \-l $SYBASE/$SYSAM/lmgrd.log&
```

4 Launch the license manager:

```
$SYBASE/SYSAM-1_0/bin/lmgr
```

5 When prompted: “Do you have Sybase Software Asset Management Certificates to register?” click Yes.

6 The SySAM License Manager screen prompts you for:

- Order Number
- Feature Name
- Feature Count
- Software Version
- Authorization Code

Click More until you have entered all licenses. Click Done.

7 Do the following:

a Verify that SySAM is running:

```
$SYBASE/$SYBASE_SYSAM/bin/lmutil lmstat -c
```

b Shut down the server to which you are adding the new license.

8 Run the license management utility lmutil lmreread from the *$SYBASE/$SYSAM/bin/* directory.

```
$SYBASE/$SYBASE_SYSAM/bin/lmutil lmreread
```

The new license is appended to the end of the *license.dat* file.

Starting Adaptive Server with optional features

When you start Adaptive Server, it automatically tries to check out a license for itself (ASE_SERVER) and a license for each enabled optional feature.

You enable or disable optional features by using sp_configure. Table 6-3 lists the configuration parameters used to enable optional features.

Table 6-3: Configuration parameters for optional features

Feature	License name	Configuration parameters
High Availability	ASE_HA	enable HA
Distributed Transaction Management	ASE_DTM	enable xact coordination enable DTM **Note** You must enable both xact coordination and DTM to use coordination services.
Java in Adaptive Server	ASE_JAVA	enable java
Advanced Security Mechanisms (This feature is only available on Sun Solaris.)	ASE_ASM	use security services

By default, the configuration parameters for optional features are set to 0 (off). To enable an optional feature, use sp_configure to set its configuration parameter to 1, and reboot Adaptive Server. During the boot sequence, Adaptive Server attempts to check out licenses for all enabled features.

For information about configuring Adaptive Server as a companion server in a high availability system, see *Using Sybase Failover in a High Availability System*.

For information about configuring Adaptive Server with distributed transaction management, see *Using Adaptive Server Distributed Transaction Management Features.*

For information about configuring the other optional features, see Chapter 7, "Adding Optional Functionality to Adaptive Server".

Troubleshooting SySAM

When you boot Adaptive Server, problems acquiring licenses or contacting the asset management software appear in the Adaptive Server error log file, *$SYBASE/$SYBASE_ASE/$SYBASE_SYSAM/log/lmgrd.log*.

Table 6-4 describes the SySAM error messages and suggests ways to correct common problems.

Table 6-4: SySAM error messages

Message	Description	Action
Use license file *filename*.	This message identifies the license file that Adaptive Server uses.	This is a normal start-up message; no action is required.
Feature *feature_name* is not licensed.	There is no license available for the specified optional feature.	You must purchase and install a valid license for the feature before it can be used. For information, see the installation guide for your platform.
There is no valid license for ASE server product. Server is booting with all the optional features disabled.	Adaptive Server cannot find a valid license file, or it cannot contact the asset management software to check out an ASE_SERVER license.	When Adaptive Server cannot check out an ASE_SERVER license, the server boots but does not enable any optional features. If you have purchased an ASE_SERVER license, make sure it has been installed. Also, make sure the license file exists and has been defined in LM_LICENSE_FILE.
License manager initialization fails.	The asset management software did not initialize.	See "Starting SySAM" on page 84.
Feature *feature_name* is over drafted *n* times.	All licenses for the specified feature have already been checked out by other servers.	Adaptive Server enables the optional feature, even though all available licenses have been used by other servers. Contact your authorized Sybase reseller to purchase an additional license for the feature.

Message	Description	Action
Warning: ASE_DTM does not have a valid license and therefore is not initialized. Warning: ASE_HA does not have a valid license and therefore is not initialized. Warning: there is no valid license for ASE_ASM and therefore feature ASM is not initialized. Warning: there is no valid license for ASE_JAVA and therefore feature JAVA is not initialized.	There is no license available for the configured feature.	Adaptive Server does not enable the optional feature until it can obtain a valid license. Purchase and install a license for the feature. Contact your authorized Sybase reseller.

Authorization code input error

When you purchase licenses for Adaptive Server optional features, you are issued a Sybase Software Asset Management Certificate that includes the following information for each product:

- Order number
- Feature name
- Feature count
- Software version
- Authorization code
- Product description

SySAM builds the license file, *license.dat*, from this information. It appends new licenses to the end of the file.

The authorization code is case sensitive. If you make a mistake while entering the authorization code, correct it by accessing the license file with an ASCII text editor, making the necessary changes, and saving the file. The file is located in *$SYBASE/$SYBASE_SYSAM/licenses/license.dat*.

Warning! Tampering with any portion of the license file other than the authorization code invalidates the license.

The format of a license file entry is:

```
INCREMENT ASE_FEATURE SYBASE nn.n license_type nnnn
nnnnnnnnnnnn SN=nnnnn OVERDRAFT=nnnnn ck=nnnnn
```

where:

- ASE_*FEATURE* is the feature license name, such as ASE_SERVER, ASE_JAVA, etc.
- *nn.n* is the version number.
- *license_type* is the type of license, such as PERMANENT.
- *nnnn* is the feature count.
- *nnnnnnnnnnnn* is the authorization code.
- SN=*nnnnn* is the order number.
- OVERDRAFT=*nnnnn* is the maximum licenses that can be checked out.
- ck=*nnnnn* is the number of licenses checked out.

Here is a sample license file:

```
SERVER server1 ANY 4100

VENDOR SYBASE
$SYBASE/$SYBASE_SYSAM/bin/SYBASEUSE_SERVER

INCREMENT ASE_SERVER SYBASE 12.0 PERMANENT 1000
123456789123 SN=10001 OVERDRAFT=10000 ck=0

INCREMENT ASE_JAVA SYBASE 12.0 PERMANENT 1000
123456789123 SN=10001 OVERDRAFT=10000 ck=0

INCREMENT ASE_DTM SYBASE 12.0 PERMANENT 1000
123456789123 SN=10001 OVERDRAFT=10000 ck=0

INCREMENT ASE_HA SYBASE 12.0 PERMANENT 1000
123456789123 SN=10001 OVERDRAFT=10000 ck=0
```

CHAPTER 7

Adding Optional Functionality to Adaptive Server

This chapter provides instructions for adding the following optional functionality for Adaptive Server:

- Sample databases – you can use French, German, Japanese, and U.S. English databases to practice most examples given in the *Transact-SQL User's Guide*.
- Auditing – you can detect access of the system and misuse of resources by using auditing to track security-related system activity in an audit trail.

After you have installed the Sybase products on your system, see the product documentation for configuration and administration issues.

Adding sample databases

This section describes how to install the U.S. English and international language sample databases. For installation instructions specific to each sample database, see:

- "Installing pubs2 and pubs3" on page 92
- "Installing interpubs" on page 94
- "Installing jpubs" on page 95

Installing pubs2 and pubs3

The **sample databases** pubs2 and pubs3 contain information about a fictional business. You can use this information to learn about Adaptive Server without affecting essential data.

Table 7-1 lists the scripts that you can use to install the sample databases.

Table 7-1: Sample database scripts

Script	Description
installpubs2	Installs the pubs2 sample database. pubs2 contains data that represents a publishing operation. Use this database to test your server connections and to learn Transact-SQL. Most of the examples in the Adaptive Server documentation query the pubs2 database.
installpubs3	Installs the pubs3 sample database. This updated version of pubs2 uses referential integrity. In addition, its tables are slightly different than the tables used in pubs2. Where noted, the Adaptive Server documentation uses the pubs3 database in its examples.
installpix2	Installs the image data that is used with the pubs2 database.

Note The master device size should be at least 30MB to install the full pubs2 database, including the image data. Run the installpix2 script after you run installpubs2.

Default devices for sample databases

The *$SYBASE/scripts* directory contains scripts for installing the us_english sample database, foreign language sample databases, and the image data associated with the U.S. English pubs2 sample database.

Each sample database requires 3MB on your database device. By default, the sample database installation scripts install the sample databases on the device that is designated as the default database device, which is the master device.

If you have not used sp_diskdefault to change the status of the master device or to specify another default device, the scripts install the sample databases on the master device. Sybase does not recommend this because it uses valuable space that is needed for system tables.

To avoid installing sample databases on the master device, do one of the following:

- Use sp_diskdefault to specify a default device other than the master device. For information on sp_diskdefault, see the *Adaptive Server Reference Manual*.
- Modify each sample database installation script to specify a different device.

Running the database scripts

To run the *installpubs2*, *installpix2*, and *installpubs3* scripts:

1 Start Adaptive Server.

2 Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device for pubs2 and pubs3. You will need to provide this information later.

3 Make a copy of the original installpubs2 and installpubs3 scripts. Be sure you can access the copies in case you have problems with the edited scripts.

4 Use a text editor to edit the script, if necessary, to specify a default device other than the master device, or use sp_diskdefault.

5 Go to the *scripts* directory in your Sybase installation directory (*$SYBASE/$SYBASE_ASE*), and use the following syntax to run the script:

```
isql -Usa -P -Sserver_name -iscript_name
```

where *server_name* represents the destination server for the database and *script_name* is the full path to and file name of the script to run.

For example, to install pubs2 on a server named VIOLIN, enter:

```
isql -Usa -P -SVIOLIN
-i $SYBASE/$SYBASE_ASE/scripts/installpubs2
```

6 To install the image data associated with pubs2, run:

```
isql -Usa -Ppassword -Sservername\
<$SYBASE/ASE-12_0/scripts/installpix2
```

Note The image data requires extra space—there are six pictures; two each in the PICT, TIFF, and Sun raster file formats. Run the images script only if you want to use or test the image datatype. Sybase does not supply any tools for displaying image data. You must use appropriate screen graphics tools to display the images after you have extracted them from the database.

Installing interpubs

interpubs is a database similar to pubs2 that contains French and German data. This data contains 8-bit characters and is available for use at Adaptive Server installations that use the ISO 8859-1 (iso_1) or Roman8 (for HP-UX) character set. To display the French and German data correctly, you must set up your terminal to display 8-bit characters.

To install interpubs:

1. Verify that iso_1 or Roman8 is installed as the default character set or as an additional character set.
2. Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device for the interpubs database. You will need to provide this information later.
3. Make a copy of the original installintpubs script. Be sure you can access this copy, in case you experience problems with the edited script.
4. Use a text editor to edit the script, if necessary, to specify a default device other than the master device, or use sp_diskdefault.
5. Execute the script, using the -J flag to ensure that the database is installed with the correct character set:

   For HP:

   ```
   isql -Usa -Ppassword -Sservername -Jroman8 \
    <$SYBASE/$SYBASE_ASE/scripts/roman8/installintpubs
   ```

   For Digital UNIX, IBM, and Sun Solaris:

   ```
   isql -Usa -Ppassword -Sservername -Jiso_1 \
    <$SYBASE/$SYBASE_ASE/scripts/iso_1/installintpubs
   ```

For more information on the -J option in isql, see the *Utility Programs for UNIX Platforms*.

Installing jpubs

If you installed the Japanese Language Module with your Adaptive Server, *$SYBASE/scripts* contains the installjpubs script for installing the jpubs database. jpubs is similar to pubs2 except it contains Japanese data. installjpubs uses either the EUC-JIS (eucjis) or the Shift-JIS (sjis) character set.

To ensure that your system can display Japanese data correctly:

- On HP-UX, verify that you are running HP Native Language I/O .
- On Sun Solaris, verify that you are running SunJLE.

To install the jpubs database:

1. Set your terminal to display 8-bit characters.
2. Verify that the EUC-JIS or Shift-JIS character set (depending on which characters you want for *jpubs*) is installed as Adaptive Server's default character set or as an additional character set.
3. Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device for the jpubs database. You will need to provide this information later.
4. Make a copy of the original installjpubs script. Be sure that you can access this copy, in case you experience problems with the edited script.
5. Use a text editor to edit the script, if necessary, to specify a default device other than the master device, or use sp_diskdefault. See "Default devices for sample databases" on page 92.
6. Execute the installjpubs script, using the -J flag to ensure that the database is installed with the correct character set:

   ```
   isql -Usa -Ppassword -Sservername -Jeucjis \
    < $SYBASE/$SYBASE_ASE/scripts/eucjis/installjpubs
   ```

   or:

   ```
   isql -Usa -Ppassword -Sservername -Jsjis \
    < $SYBASE/$SYBASE_ASE/scripts/sjis/installjpubs
   ```

For more information about the -J option in isql, see *Utility Programs for UNIX Platforms*.

Note Since you may want to refresh or make new copies of the sample databases, back up the original and edited versions of the installation scripts and store the copies in a safe place.

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 a wide range of privileges, including permissions to select, insert, update, and delete user tables. For more information about the “guest” user and “guest” permissions, see the *System Administration Guide*.

If possible, and if space allows, give each new user a clean copy of the sample databases so that she or he is not confused by other users’ changes.

If space is a problem, you can instruct the user 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.

Adding auditing

Auditing is an important part of security in a database management system. Security-related system activity is recorded in an audit trail, which can be used to detect penetration of the system and misuse of resources. By examining the audit trail, the System Security Officer can inspect patterns of access to objects in databases and can monitor the activity of specific users. Audit records can be traced to specific users, enabling the audit system to act as a deterrent to users who are attempting to misuse the system.

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 audit data.

Audit system devices and databases

The audit system includes several components. The main components are:

- The sybsecurity device and the sybsecurity database, which stores audit information
- The audit trail, which is composed of several audit devices and tables that you determine at configuration time
- The syslogs transaction log device, which stores transaction logs

The sybsecurity device and database

The sybsecurity device stores the sybsecurity database. The sybsecurity database is created as part of the auditing configuration process. It contains all the system tables in the model database as well as a system table for keeping track of server-wide auditing options and system tables for the audit trail.

Tables and devices for the audit trail

Adaptive Server stores the audit trail in system tables, named sysaudits_01 through sysaudits_08. For example, if you have two audit tables, they are named sysaudits_01 and sysaudits_02. At any given time, only *one* of the audit tables 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.

When you configure Adaptive Server for auditing, you determine the number of audit tables for your installation. You can specify up to eight system tables (sysaudits_01 through sysaudits_08). Plan to use at least two or three system tables for the audit trail and to put each system table on its own device, separate from the master device. If you do this, you can use a threshold procedure that archives the current audit table automatically, before it fills up and switches to a new, empty table for subsequent audit records.

Device for syslogs systems table

When you configure for auditing, you must specify a separate device for the syslogs system table, which contains the transaction log. The syslogs table, which exists in every database, contains a log of transactions that are executed in the database.

Overview of audit installation

There are two methods for installing auditing for the first time in Adaptive Server:

- Use the installsecurity script. For information, see the *System Administration Guide*.

- Use the auditinit utility. Tasks that you must perform before installing auditing and instructions on using the auditinit utility follow.

Pre-installation tasks

Determine the location of the raw devices for the sybsecurity, syslogs, and sysaudits table devices. You will need to provide this information later.

Sybase recommends that you:

- Configure your system with the minimum number of auditing devices you require—you must configure at least three devices. You can add more auditing devices later with sp_addaudittable. For information, see the *Adaptive Server Reference Manual*.
- Install auditing tables and devices in a one-to-one ratio. Tables that share the same device will share the same upper threshold limit. These tables cannot be used sequentially when a device fills up, because they both reside on the same device.
- Install each auditing table on its own device. This enables you to set up a smoothly running auditing system with no loss of auditing records. With two auditing tables, when one fills up, you can switch to the other. With a third auditing table, if one device fails, the System Security Officer can install a new threshold procedure that changes the device rotation to skip the broken device until the device is repaired.
- Make the device larger than the table. When you use only three auditing tables and devices, the size of the table and the size of the device can be similar, because you can obtain more auditing capacity by adding more auditing tables and devices (up to eight). When you are working toward the upper table and device limit (six to eight), you may want to make the device considerably larger than the table. Then, you can expand the table size later towards the upper size of the device when a larger auditing capacity is desired, and few or no device additions are available.

Installing auditing

To configure Adaptive Server for auditing:

1. Log in to your machine using your Sybase System Administrator (“sa”) user account.

2 Start auditinit at the UNIX prompt:

```
$SYBASE/$SYBASE_ASE/install/auditinit
```

auditinit displays the following menu:

```
AUDITINIT
1. Release directory:  /usr/u/sybase
2. Configure a Server product
```

3 Select Configure a Server Product.

4 Select Adaptive Server.

5 Select Configure an Existing Sybase Server.

6 Select the server to configure.

7 Provide the SA Password for the server you selected.

8 From the Sybase Server Configuration screen, select Configure Auditing.

As you proceed through the menus in auditinit, you can change any default values that appear. As you finish each menu, press CTRL+A to accept the defaults or changed values and move to the next menu.

```
CONFIGURE AUDITING
1. Configure auditing:  no
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name    Table name    Size

Device for the audit datbase transaction log:
Logical name    Physical name    Segment name    Table name    Size
```

9 From the Configure Auditing screen, select Configure Auditing.

auditinit redisplays the Configure Auditing menu with the value “yes” displayed for Configure Auditing.

Creating a device for an audit table

To create a device for an audit table:

1 From the Configure Auditing screen, select Add a Device for Audit Table(s).

auditinit displays the following menu:

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
2. Logical name of the device:
3. Size of the device (Meg):
4. Device size for auditing:
```

2 Select Sybsecurity Physical Device Name.

3 Enter the *full path* of the physical device (raw partition) that you located in “Pre-installation tasks” on page 98.

```
Enter the physical name of the device to use for the
audit database (default is “ ”):

/dev/path_to_partition
```

where *path_to_partition* is the path to the raw partition for the device.

If you specify an operating system file, the following warning appears:

```
WARNING: '/secret1/sybase_dr/install/aud1.dat' is a
regular file which is not recommended for a Server
device.
```

4 Press Return to acknowledge the warning.

auditinit redisplays the Add/Change a New Device for Auditing menu, which displays the physical name of the device:

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device
name:  /secret1/sybase_dr/install/aud1.dat
2. Logical name of the device:
3. Size of the device:
4. Device size for auditing:
```

5 Proceed through the remaining items on this menu.

Note The Size of the Device value must be equal to or greater than the Device Size for Auditing value. The Device Size for Auditing must be equal to the device size. If you are following Sybase auditing guidelines, you do not need to change the value displayed in Device Size for Auditing.

6 Press CTRL+A to accept the settings. auditinit returns to the Configure Auditing menu and displays the device you have created.

```
CONFIGURE AUDITING
 1.  Configure auditing:  yes
 2.  Add a device for audit table(s)
 3.  Add a device for the audit database transaction log
 4.  Delete a device entry
 5.  Change a device entry

List of devices for the audit tables:
Logical name   Physical name    Segment name         Table name    Size

6.Audit_01' secret1/sybase_dr/install/aud1.dat’ sysaudits_01 5
```

7 To add multiple audit devices, repeat steps 1–6.

You can add as many as eight devices. Sybase recommends adding three or more audit table devices.

After adding a device, auditinit returns to the Configure Auditing menu and displays all the devices you have created.

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

6. Audit_01'   /secret1/sybase_dr/install/aud1.dat’ sysaudits_01   5
7. Audit_02'   /secret1/sybase_dr/install/aud2.dat' sysaudits_02   5
```

Creating a device for the audit database transaction log

To create a device for the audit database transaction log:

1 From the Configure Auditing menu, select Add a Device for the Audit Database Transaction Log.

auditinit displays the Add/Change a New Device for Auditing menu.

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.  sybsecurity physical device name:
2.  Logical name of the device:
3.  Size of the new device (Meg):
4.  Device size for auditing:
```

2 Select Sybsecurity Physical Device Name.

auditinit prompts for the physical name and supplies you with a default, if available:

```
Enter the physical name of the device to use for the
sybsecurity database (default is''):
 /dev/path_to_partition
```

where *path_to_partition* is the path to the raw partition for the device.

3 Enter the full path name of a physical device.

If you enter an operating system file name, the following warning appears:

```
WARNING: '/secret1/sybase_dr/install/audlog' is a
regular file, which is not recommended for a Server
device.
```

4 Press Return to acknowledge this warning.

auditinit displays the Add/Change a New Device for Auditing menu and the value you selected for the physical name of the device.

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.sybsecurity physical device name:
      /secret1/sybase_dr/install/auditlog.dat
2.Logical name of the device:
3.Size of the device:
4.Device size for auditing:
```

5 Proceed through the remaining items on this menu. As you do so, be aware of the following:

- Sybase recommends a minimum size of 2MB for the size of the transaction log.
- auditinit displays the size in both Size of the Device and in Device Size for Auditing in the Add/Change a New Device for Auditing menu.
- The Device Size for Auditing default value is equal to the size of the device, based on the assumption that you may want to devote the entire device to log for the auditing task. If you want to use only a subset of the device, you can edit the Size of the Device value.

6 Press Ctrl-a to accept the settings displayed in the Add/Change a New Device for Auditing menu.

auditinit returns to the Configure Auditing menu and displays all the devices you have created.

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
```

```
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

6. Audit_01'   /secret1/sybase_  dr/install/aud1.dat’ sysaudits_01   5
7. Audit_02'   /secret1/sybase_  dr/install/aud2.dat' sysaudits_02   5
8. auditlog    /secret1/.../auditlog.dat logsegment   syslogs        2
```

7 When you are ready to execute the audit configuration, press Ctrl-a. auditinit returns you to the Sybase Server Configuration screen.

8 Press Ctrl-a again. auditinit prompts with:

```
Execute the Sybase Server Configuration now?
```

9 Enter “y” (yes).

auditinit executes the tasks to install auditing. When the installation completes successfully, the following messages are displayed:

```
Running task: install auditing capabilities.
....................Done
Auditing capability installed.
Task succeeded: install auditing capabilities.
Configuration completed successfully.
Press <return> to continue.
```

Enabling auditing

After auditing is installed, no auditing occurs until a System Security Officer enables auditing with sp_configure. For more information, see the *System Administration Guide*.

Deleting a device entry

To delete a device entry:

1 Select Delete a Device Entry from the Configure Auditing menu.

2 Enter the number of the device to delete.

3 Press RETURN.

Changing a device entry

To change a device entry:

1 Select Change a Device Entry from the Configure Auditing menu.

2 Enter the number of the device to change.

auditinit displays the Add/Change a New Device for Auditing menu with information on the device you selected:

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
```

```
1. sybsecurity physical device name:
      /secret1/sybase_dr/install/audlog
2. Logical name of the device: aud.log
3. size of the new device (Meg): 5
4. Device size for auditing:5
```

3 Select each remaining entry you want to change.

4 Press CTRL+A to save the new entries.

Index