SYBASE®

Configuration Guide

Adaptive Server® Enterprise

12.5.1

[UNIX]

DOCUMENT ID: DC35823-01-1251-04

LAST REVISED: March 2007

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

This manual, the *Configuration Guide*, provides instructions for performing specific configuration tasks for Sybase® Adaptive Server® Enterprise on UNIX platforms.

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

This manual covers the following topics:

- Instructions for reconfiguring certain attributes of your existing Sybase Adaptive Server Enterprise, Backup Server™, Adaptive Server Enterprise Monitor™, and XP Server™ to meet your specific needs.
- Creating network connections.
- Configuring optional functionality.
- Performing operating system administration tasks.
- Working with system administration issues that are relevant to Adaptive Server running on UNIX platforms. This manual supplements the *System Administration Guide* and the *Performance and Tuning Guide*.

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 manual is for System Administrators or other qualified installers who are familiar with their system's environment, networks, disk resources, and media devices.

How to use this book

This manual contains the following chapters:

- Chapter 1, "Introduction" is an overview of Adaptive Server and 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” describes how to set up your operating system to work with Adaptive Server.
- Chapter 4, “Adaptive Server Default Configuration” provides information about the default parameter settings of Adaptive Server.
- Chapter 5, “Setting Up Communications Across the Network” describes how to use the interfaces file to establish network connections for servers and clients.
- Chapter 6, “Using the Lightweight Directory Access Protocol as a Directory Service” provides information about using LDAP directory services to establish connections.
- Chapter 7, “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 8, “Logging Error Messages and Events” describes how to use the error logging features of Adaptive Server.
- Chapter 9, “Managing Adaptive Server Databases” provides information about the administration of Adaptive Server databases, including both routine tasks and performance and tuning considerations.
- Chapter 10, “Adding Optional Functionality to Adaptive Server” provides instructions for adding optional functionality, such as auditing and the sample databases.
- Chapter 11, “Managing Sybase Licenses for Adaptive Server: SySAM Overview,” discusses the Sybase license manager SySAM, how to manage Sybase licenses, Adaptive Server features that can be licensed, and how to add feature licenses when installing, upgrading or starting Adaptive Server.

Related documents

The Sybase Adaptive Server Enterprise documentation set consists of the following:

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

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

- The *Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5.1, the system changes added to support those features, and the changes that may affect your existing applications.
- *ASE Replicator User's Guide* – describes how to use the ASE Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.
- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as data types, functions, and stored procedures in the Adaptive Server database.

- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:
  - *Basics* – the basics for understanding and investigating performance questions in Adaptive Server.
  - *Locking* – describes how the various locking schemas can be used for improving performance in Adaptive Server.
  - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
  - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL® information:
  - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
  - *Commands* – Transact-SQL commands.
  - *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and `dbcc` stored procedures.
  - *Tables* – Transact-SQL system tables and `dbcc` tables.

- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
- *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Web Services User's Guide* – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.
- *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

Other sources of information

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

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

- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (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 the Technical Library.

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

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

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

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

2 Select Products from the navigation bar on the left.

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

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

5 Click a Certification Report title to display the report.

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

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

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

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

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

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

2. Select EBFs/Maintenance. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).
3. Select a product.
4. Specify a time frame and click Go.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

The following sections describe conventions used in this manual.

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

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

Table 1: Font and syntax conventions for this manual

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

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

Type parentheses as part of the command.	`compute` *`row_aggregate`* `(`*`column_name`*`)`
Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates “is defined as”.	`::=`
Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces.	`{cash, check, credit}`
Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.	`[cash | check | credit]`
The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command.	`cash, check, credit`

Element	Example
The pipe or vertical bar(|) means you may select only one of the options shown.	`cash | check | credit`

An ellipsis (...) means that you can *repeat* the last unit as many times as you like.	`buy thing = price [cash | check | credit]`
	`[, thing = price [cash | check | credit]]...`
	You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment.

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

  sp_dropdevice [*device_name*]

  For a command with more options:

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

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

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

  ```
  select * from publishers
  ```

- Examples of output from the computer appear as follows:

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

(3 rows affected)
```

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

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

Terms

The following terms appear repeatedly throughout this book. For more detailed information about these and other terms, see the *Glossary*.

- */work/sybase* – is given as an example of the Sybase installation directory.
- *Text editor* – refers to an ASCII text editor or any editor that can save files to text format.

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

Sybase Adaptive Server Enterprise for UNIX is a full-featured Adaptive Server that runs on the following platforms:

- HP Tru64 UNIX
- HP-UX
  - PA RISC
  - IPF/HP-UX
- IBM AIX
- Sun Solaris
- Linux
  - Linux 32
  - Linux Itanium
- Macintosh OS X
- Silicon Graphics IRIX

Note The instructions in this book assume that Adaptive Server is installed and running. For information about installing and starting Adaptive Server, as well as for an overview of Adaptive Server, see the Adaptive Server *Installation Guide* for your platform.

This chapter provides an overview of how to configure Adaptive Server 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.

Adaptive Server 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 Historical Server obtains performance data from Monitor Server and saves the data in files for use at a later time.

  Note Monitor Server and Historical Server are not available on Macintosh OS X.

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

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:

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

System-specific issues are described in this document. For more information about system-specific issues, see the *Installation Guide* and release bulletin for your platform.

System 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. This login is preferred, but not required.

Note Adaptive Server on Mac OS X does not require a “sybase” account or an account with root privileges. However, administrator privileges are required to update */Library/StartupItems*.

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 SYBASE.
- DSQUERY – defines the Adaptive Server name that client programs attempt 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 SYBASE.
- 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 of the Adaptive Server component.
- SYBASE_OCS – defines the subdirectory to which Open Client™ is set.

- SYBASE_ FTS – defines the subdirectory for the full-text search components.

Note Full-text search components are not available on Mac OS X.

- SYBASE_SYSAM – points to the license-management software directory.
- LM_LICENSE_FILE – points to the *license.dat* file, which is located in the *$SYBASE_SYSAM/licenses/* directory by default.
- PATH – specifies 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 – this variable specifies the directory to search for shared libraries. The Sybase shared libraries are located in the installed component's */lib* 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.
```

- DYLD_LIBRARY_PATH – this variable is specific to Macintosh OS X. It accomplishes the same tasks as LD_LIBRARY_PATH.
- LIBPATH – on the IBM RS/6000 platform, this variable specifies which directory to search for libraries.

  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.
```

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 or operating system files.

Adaptive Server requires the following devices:

- master – to store system databases.
- sybsystemdb – to store information about distributed transaction.
- sysprocsdev – to store system procedures.

The master, sybsystemdb, 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 temporary databases in Adaptive Server. Adaptive Server supports multiple tempdbs. See "create temporary database" in the *T-SQL User's Guide*. Each time Adaptive Server is started, the tempdb database is cleared and rebuilt from the model database.
- The sample databases, pubs2 and pubs 3, are stored on the master device at installation, but should be moved to a user-defined device after installation.

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

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

The sybsystemdb database is required to support distributed transaction management (DTM) features. Before installation, make sure you have enough space available on the default segment to support sybsystemdb.

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 – 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 – contains French and German data.
- jpubs – contains Japanese data.

For information about installing the sample databases, see Chapter 10, “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 10, “Adding Optional Functionality to Adaptive Server.” For information about installing and using the auditing system, see Chapter 12, “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 is used.

dbcc checkstorage records consistency 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*.

sybmgmtdb database

sybmgmtdb is a Job Scheduler database.

All the job, schedule, and scheduled job information, and data needed by the Job Scheduler task for internal processing is stored in the sybmgmtdb database. Most access to data in the sybmgmtdb database is via stored procedures. The stored procedures make the data available to the GUI, the JS Agent and the command-line interface. Only the Job Scheduler task accesses data directly from the sybmgmtdb database.

For more information about sybmgmtdb and Job Scheduler, see *Job Scheduler User’s Guide*.

dsync option *on* by default for database device files

The dsync option ensures Adaptive Server can recover data from devices on the file systems. By default, Adaptive Server enables dsync for file system devices. However, dsync may cause performance degredation for file system devices that experience high write activity. The dsync option can be set or reset using the disk init, disk reinit, and sp_deviceattr commands.

Note The dsync option is ignored for raw devices.

When installing a new Adaptive Server, by default dsync is set on for all file system devices.

When upgrading from ASE 12.0 or earlier releases on a UNIX server that stored databases on UNIX file system devices, by default dsync is set:

- on for the master device
- off for all other devices

Immediately after upgrade, set the dsync option for the file system devices. See also sp_deviceattr and sp_helpdevice in the *Reference Manual*

Warning! Data corruption may occur if dsync option is not set immediately after upgrade.

Determining the location, type, and size of a database device

Adaptive Server requires several database devices. Table 1-1 below shows baseline values for each of the devices. See the release bulletin for any last-minute changes to these values.

Table 1-1: Adaptive Server database devices

Device	Purpose	Minimum size	Minimum recommended size
master	Stores system databases	14MB for 2K pages 27MB for 4K pages 50MB for 8K pages 105MB for 16K pages	30MB 60MB 120MB 240MB
sysprocsdev (also called the sybsystemprocs device)	Stores the sybsystemprocs database	112MB	120MB (plus any space for holding stored procedures that you have created)
sybsystemdb	Transaction processing	4MB	5 – 20MB
sybsecurity (optional)	Required for auditing	5MB	7MB; more for specialized auditing

For Solaris, HP-UX, IBM AIX, and Linux

For all databases you can use either a raw partition or a file.

Sybase Adaptive Server supports the database devices on NFS- and CIFS-mounted devices with Network Appliance Filers for storing data. Network appliance filers provide the same performance and data integrity as raw devices. There are no changes needed to the operating system or to Sybase Adaptive Server to use network appliance filers.

NFS mounted devices have been tested on Solaris, HP/UX, IBM AIX, Windows NT, Linux, and SGI.

Note Previous versions of Adaptive Server performed buffered writes to file devices, which does not ensure recoverability. Starting in version 12.0, the default for writes to file systems is unbuffered, so files are now acceptable as database devices.

For HP Tru64 UNIX

Do not run Adaptive Server on network-mounted devices, such as NFS- or RFS-mounted files, for these reasons:

- Network problems can cause serious data integrity problems because the network lies between Adaptive Server and the devices on which it writes.
- If the remote machine is down when you restart Adaptive Server, and database devices cannot be activated, automatic recovery fails.
- Adaptive Server activity is suspended when the remote machine or network is down.

For SGI

Do not run Adaptive Server on network-mounted devices, such as NFS-mounted files, for these reasons:

- Network problems can cause serious data integrity problems because the network lies between Adaptive Server and the devices on which it writes.
- If the remote machine is down when you restart Adaptive Server, and database devices cannot be activated, automatic recovery fails.
- Adaptive Server activity is suspended when the remote machine or network is down.

For all platforms

You may need to repartition the disk you choose. Contact your operating system administrator for assistance.

When preparing a raw partition device, follow these guidelines:

- Do not initialize a database device on the partition that contains your Sybase installation software. Doing so destroys all existing files on that partition.
- A raw partition designated for use by Sybase cannot be mounted for use by the operating system for any other purpose, such as for file systems or swap space.
- After a Sybase configuration utility or the disk init command has initialized a portion of a partition as a database device, the entire partition cannot be used for any other purpose. Any space left on the partition beyond the size specified for the device becomes inaccessible, unless you drop and re-create the device.
- For best performance, place the Sybase software and all disk devices, including the master device, on the same machine.
- To avoid any possibility of using a partition that contains the partition map, do not use cylinder 0.

v Choosing a raw partition

1 Determine which raw partitions are available.

2 Determine the sizes of the raw partitions.

3 From the list of available raw partitions, select a raw partition for each device, based on the size recommendations in Table 1-1 on page 9.

4 Verify with the operating System Administrator that the partition you have chosen is available.

For Silicon Graphics IRIX The System Administrator should mark this partition in */etc/fstab* as being of the rawdata datatype

5 Make sure the “sybase” user has read and write privileges to the raw partition.

Note For more information on , see your operating system documentation.

Client/server communication

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. Names and addresses of every known server are listed in a directory services file. This information can be stored in a directory services file two different ways:

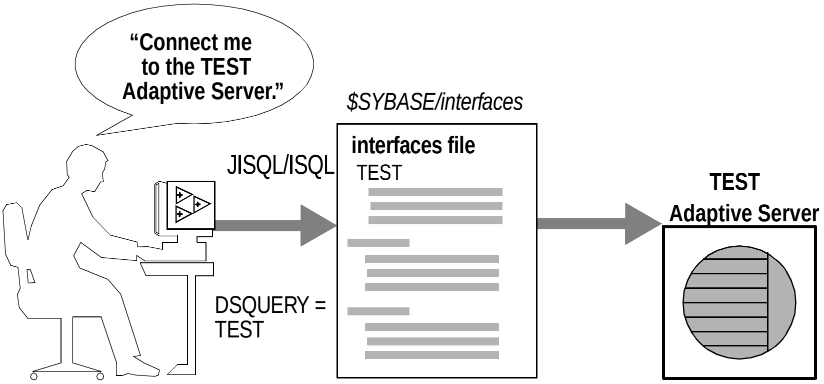
- In an interfaces file, named *interfaces* on UNIX platforms, located in the *$SYBASE* installation directory, or
- In an LDAP server

After your Adaptive Server or client software is installed, it can connect with any server on the network that is listed in the directory services.

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 directory services 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, Open Server, Backup Server, or Monitor Server to which they want to connect. It also tells the server where to listen for incoming connection attempts from clients. The server uses a single port for these two services (referred to as *query service* and *listener service*).

Figure 1-1: Communicating with a server using the interfaces file



During installation, you use the srvbuild utility to create and configure 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 how to modify existing interfaces file entries using dsedit and dscp or to create new interfaces file entries for existing servers, Chapter 6, "Using the Lightweight Directory Access Protocol as a Directory Service."

Changing Adaptive Server configuration

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 *Reference Manual*.

For information about configuring languages, character sets, and sort orders, see Chapter 7, "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 *Using Adaptive Server Distributed Transaction Management Features*.

Languages other than U.S. English

If you are running srvbuild in a language other than U.S. English, verify that any input you provide uses a character set that is supported by the us_english character set.

Note The srvbuild utility is not available on Mac OS X.

Note The us_english character set does not support accent marks, such as tildes (~) and umlauts (ü). This prevents srvbuild from supporting the character sets that use these characters.

For more information about languages, character sets, and sort orders, see the *Installation Guide* for your platform. For more information on srvbuild, see the *Utility Guide*.

Adaptive Server specifications

Database specifications		
Databases per Adaptive Server	32,767	Practical limit is approximately 100
Maximum database size	2^{43} (8TB)	Minus overhead for system databases
Minimum allowable sybsystemprocs database	120MB	Required for an upgrade
Maximum size of a database device (disk partition)	2^{24} (32GB)	If the Operating System supports file sizes up to 32GB, then Adaptive Server supports file system devices up to 32GB
Maximum number of database devices per server	256	
Maximum number of devices or device pieces per database	Unlimited	
Maximum number of segments per database	31	

Maximum number of login IDs per server	2147516416	
Maximum number of users per database	2146484223	
Maximum number of groups per database	1032193	
pubs2 database	2MB	Minimum required for a new installation
	4MB	Free space required for an upgrade
pubs3 database	2MB	Minimum required for a new installation
	4MB	Free space required for an upgrade
interpubs database	2MB	Minimum required for a new installation
	4MB	Free space required for an upgrade
jpubs database	2MB	Minimum required for a new installation
	4MB	Free space required for an upgrade

Table specifications

User objects per database	2^{31} – 100	
Indexes per table	250 (one clustered index)	
Rows per table	Limited by available storage	Maximum 2^{32}
Columns per composite index	31	
Creation of clustered index	1.2*(x + y) x = total data space in table, y = sum of space of all nonclustered indexes on table, and 20 percent overhead for logging	For sorted data, approximately 20 percent of the table size needed
Maximum size of object name	30	

Query specifications

Maximum number of tables participating in a query, for a query without a union, or for each side of a union in a query	64	Maximum of 50 user tables, including result tables, tables referenced by views (the view itself is not counted) correlations and self-joins; maximum of 14 worktables
Maximum number of tables in a “union” query	256	Includes up to 50 user tables and 14 work tables on every side of the union, for a total of up to 256 tables across all sides of the union.
Maximum number of databases participating in one transaction	Unlimited	Includes database where transaction began, all databases changed during transaction, and `tempdb`, if it is used for results or worktables
Practical number of databases participating in one query	16	Includes each occurrence of each database queried and `tempdb`, if it is used for results or worktables
Maximum number of tables with referential integrity constraints for a query	192	
Procedure specifications		
Number of buffers and procedure buffers	Configurable	Limited by amount of RAM and maximum size of shared memory segment
Minimum memory required per stored procedure	2K	
Maximum number of parameters per stored procedure	2048	

Adaptive Server extended-limit capabilities vary by type of table and the database logical page size. Table 1-2 lists the column and row limits for allpages-locked (APL) tables.

Table 1-2: Allpages-locked (APL) tables

Maximum APL table limits	Number of columns	Column size 2K page	Column size 4K page	Column size 8K page	Column size 16K page
Fixed-length column	1024	1960 bytes	4008 bytes	8104 bytes	16296 bytes
Variable-length column	254	1948 bytes	3988 bytes	8068 bytes	16228 bytes

Table 1-3 lists the column and row limits for data-only-locked (DOL) tables.

Table 1-3: Data row and data page tables.

Maximum DOL table limits	Number of columns	Column size 2K page	Column size 4K page	Column size 8K page	Column size 16K page
Fixed-length column	1024	1958 bytes	4006 bytes	8102 bytes	16294 bytes
Variable-length column	1024	1954 bytes	4002 bytes	8089 bytes	16290 bytes

Database space requirements depend upon the logical page size of the server. Table 1-4 lists the minimum size for each database.

Table 1-4: Database requirements for varying page sizes

Databases	2K page	4K page	8K page	16K page
Default database size	2MB	4MB	8MB	16MB
master database	6MB	12MB	24MB	48MB
model database	2MB	4MB	8MB	16MB
tempdb database	3MB	4MB	8MB	16MB

Larger logical page sizes can contain more data. Table 1-5 lists the maximum data for each logical page size.

Table 1-5: Data limits for tables according to page size

Tables	2K page	4K page	8K page	16K page
Maximum number of data bytes per text or image file	1800	3600	7650	16200
Bytes per index key	600	1250	2600	5300
User-visible row length DOL table	1958	4006	8102	16294
User-visible row length APL table	1960	4008	8104	16296

CHAPTER 2 Starting and Stopping Servers

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

Overview

The methods described in this chapter are used 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 by Adaptive Server when an XP command is issued through isql.

You can use Sybase Central to start and stop servers manually or automatically. For more information about Sybase Central, see "Chapter 1, Overview of System Administration," in the *System Administration Guide*.

Requirements for starting servers

To start a server, your user account must have:

- Anyone with access to the database servers (execute privileges) and database devices (read/write privileges).
- Access to Adaptive Server distribution files
- The system environment variables discussed in Chapter 1, "Introduction"

The installation program creates the *interfaces* file and system environment variables when you install servers on your computer.

Starting servers

You can start Adaptive Server, Backup Server, or Monitor Server from the command line by using *RUN_server_name* files and startserver commands. You can customize the starting options.

You can also customize the start up options by editing the *RUN_server_name*.

Server start-up parameters

The default start-up parameters are stored under *$SYBASE/SYBASE_ASE/install/RUN_server_name*, where *server_name* is the name of the server you installed.

Backup Server, Monitor Server, and Historical Server server names are appended with “_back”, “_mon”, and “_hs”, respectively.

Table 2-1 lists the default start-up parameters for Adaptive Server.

Table 2-1: Default Adaptive Server start-up parameters

Switch	Description
-d *$SYBASE/data/master.dat*	Location of the master device file
-s *server_name*	Name of the Adaptive Server
-e *$SYBASE/$SYBASE_ASE /install/errorlog*	Location and name of the error log file
-M *$SYBASE/$SYBASE_ASE*	Directory that stores shared memory files

Changing start-up parameters

You cannot change any of these default start-up parameters unless you directly edit the *$SYBASE/$SYBASE_ASE/install/RUN_ server_name* file. You can also specify additional start-up parameters within the *$SYBASE/$SYBASE_ASE/install/RUN_ server_name* file.

Additional start-up parameters include any valid server command line options listed for the databaserver and backupserver descriptions in the *Utility Guide*. For a description of start-up parameters for Monitor Server and Historical Server, see the Adaptive Server Enterprise Monitor documentation.

Using a *RUN_server_name* file

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

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

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

Warning! Do not delete the *RUN_server_name* 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 *RUN_server_name* file in another location, make a copy of the original *RUN_server_name* file in the new location.

The *RUN_server_name* file for a server named SYBASE is named *RUN_SYBASE*. If the *RUN_server_name* file for your current Adaptive Server is named *RUN_server_name*, 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 RUN_server_name file ]
```

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

If your Adaptive Server is named SYBASE, the *RUN_server_name* 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 the Adaptive Server *Utility Guide*.

Using the monserver and backupserver commands

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 *Monitor Server User's Guide*.

For HP Tru64 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 Designate a period of time for your server to wait before starting, by inserting a sleep command at the beginning of the linked *RUN_server_name* 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 */sbin/rc*.d* 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 RUN_server_name to the name of the RUN_server_name file
#     for the Adaptive Server
# set BSERVER to the name of the RUN_server_name file
#     for the backup server
#
SYBASE=/sybase/rel1251
RUN_server_name=RUN_REL1251_XYZ
BSERVER=RUN_REL1251_XYZ_BACKUP_back
PATH=/sbin:/usr/sbin:/usr/bin
#
export PATH SYBASE RUN_server_name BSERVER
#
case "$1" in
'start')
       echo "Starting Sybase Adaptive Server"
       nohup su sybase -c \
          "$SYBASE/$SYBASE_ASE/install/startserver -f
           $SYBASE/$SYBASE_ASE/install/$RUN_server_name >/dev/null &"
       echo "Starting Sybase Backup Server"
       nohup su sybase -c \
          "$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 *RUN_server_name* 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, after starting the network operations. 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, you can restart Adaptive Server automatically whenever the UNIX operating system restarts. To have Adaptive Server restart automatically, place the startserver command in the */etc/inittab* file.

Here 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 *RUN_server_name* 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 does not start. Also, your server does not start if the network is not running.

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 *RUN_server_name* 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 *RUN_server_name* file in this command.

For SGI

On production systems, Adaptive Server should start automatically when the UNIX operating system restarts. To do this, add an Adaptive Server start-up script as a file in the */etc/rc2.d* directory. For example, if the SYBASE installation is in */home/u/test* and your server is named TEST, add the following lines to a file in directory */etc/rc2.d*, according to the instructions in your operating system documentation:

```
SYBASE=/home/u/test
export SYBASE
set SYBASE -c "$SYBASE/$SYBASE_ASE/install/startserver
-f $SYBASE/$SYBASE_ASE/install/RUN_TEST &"
```

Note 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.

The first two lines set the SYBASE environment variable so that Adaptive Server is started from the *$SYBASE* directory. Substitute the path name on the first line for your *$SYBASE* directory as appropriate.

Note All network resources must be available before Adaptive Server is started. Adaptive Server does not start if the network is not running. Be sure that the entry for restarting Adaptive Server *follows* any commands for starting network operations in the *rc2.d* directory. Slow start-up of network operations may prevent Adaptive Server from starting, even if the commands in *rc2.d* are in the correct order.

For Sun Solaris and Linux

Before you set Adaptive Server or Backup Server to start automatically, make sure that all network resources are available. Your server does not start if the network is not running. 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 may 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 *RUN_server_name* 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 *RUN_server_name* 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 for Sun Solaris and 5 for Linux

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

Use syntax like the following for Sun Solaris:

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

For Linux:

Use ln -s to create symbolic links from the script name to the appropriate run control (*rc*) directory, *rc#* where # is the run level. Use syntax like the following:

```
ln -s /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 27 in place of 77.

For Linux:

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

```
ls /etc rc5.d/S*
```

returns:

```
/etc/rc5.d/S12syslog
/etc/rc5.d/S14nfslock
/etc/rc5.d/S27ypbind
```

and so on. If your script is named sybstart, type the following:

```
ln -s /etc/init.d/sybstart    /etc/rc5.d/S99sybstart
```

In this example, you can use any number greater than 27 in place of 99.

For Macintosh OS X

The boot sequence

During start-up, Mac OS X executes a boot sequence that readies the system for use. This section summarizes what happens during this sequence, and how to add automatic start-up of Adaptive Server.

BootROM

When the power to a Macintosh computer is turned on, the BootROM (which is part of the computer's hardware) has two primary responsibilities: to initialize system hardware, and to select an operating system to boot. BootROM has two components to help it carry out these functions:

- POST (Power-On Self Test) initializes some hardware interfaces and verifies that sufficient RAM is available and is usable.
- Open Firmware initializes the rest of the hardware, builds the initial device tree (a hierarchical representation of devices associated with the computer), and selects the operating system to use.

BootX

When BootROM (or the user) selects Mac OS X as the operating system to boot, control passes to BootX (located in */System/Library/CoreServices*). BootX's principal duty is to load the kernel environment. As it does this, BootX draws the "booting" image on the screen.

When loading the kernel environment, BootX first attempts to load a previously cached set of device drivers (called an mkext cache) for hardware that is involved in the boot process. If this cache is missing or corrupt, BootX searches */System/Library/Extensions* for drivers and other kernel extensions whose OSBundle Required property is set to a value appropriate to the type of boot (for example, local or network boot). See the kernel developer documentation for more information on the OSBundle Required key and the loading of device drivers during booting.

Once the kernel and all drivers necessary for booting are loaded, BootX starts the kernel's initialization procedure. At this point, enough drivers are loaded for the kernel to find the root device. Also from this point, Open Firmware is no longer accessible.

Once the root file system is mounted, the kextd daemon is started during system initialization (see “System Initialization,” below). kextd examines all the drivers available on the system and unloads any unnecessary drivers, freeing memory. From this point onward, kextd attempts to fulfill any requests for loading kernel extensions.

Kernel initialization

In this phase, the kernel initializes the Mach and BSD data structures and then initializes the I/O Kit. The I/O Kit links the loaded drivers into the kernel, using the device tree to determine which drivers to link. Once the kernel finds the root device, it roots BSD off of it.

Finally, the kernel starts the mach_init process, the first process in user space. The mach_init process is the Mach bootstrap port server, which enables Mach messaging.

System initialization

The mach_init process starts the BSD init process. BSD init, which has a process ID (PID) of 1, “owns” every other process on the system, and performs four principal tasks:

- Determines if the user wants single-user mode or is booting from a CD. If either of these conditions apply, an advisory prints and control is handed over to the user.
- Runs the system-initialization shell scripts */etc/rc.boot* and */etc/rc* which complete basic initialization tasks. */etc/rc* runs the SystemStarter program, which handles more specialized initialization tasks specified as “start-up items.” For details see “Start-up items” on page 31.
- Via the getty command, init launches the login window application, which displays the login window, validates entered user names and passwords, and completes the login procedures.
- As the parent of all processes, init handles all necessary cleanup of processes as they terminate.

The *rc.boot* and *rc* scripts

The *rc.boot* and *rc* Bourne shell scripts in */etc* perform basic initialization tasks. First the *rc.boot* script performs a file system consistency check (fscc) and synchronizes memory with the file system (sync). Then the *rc* script:

- Mounts the essential local file systems as defined in the various */etc/fstab* configuration files.
- Starts the device-driver loader (kextd).
- Starts the window server.
- Runs the update background process, which periodically flushes the file system cache.
- Creates the swap file for the virtual memory system and starts the dynamic pager.
- As the final step, the *rc* script starts the SystemStarter program to process the local and system start-up items. For more information, see “Start-up items,” below.

Start-up items

Start-up items are procedures run during the last phase of booting to prepare a Mac OS X system for normal operation. They consist of programs (including shell scripts) that perform tasks such as clearing away temporary files and starting system daemons. The system start-up items (that is, those provided by Apple) are located in */System/Library/StartupItems*. Do not modify the items in this directory. However, you can define your own start-up items; these custom start-up items are stored in */Library/StartupItems*.

The SystemStarter programs, which is the last thing run by the *rc* script, coordinates the execution of all start-up items. A start-up item is a folder containing at least two files:

- A program (typically a shell script) that takes the same name as the folder
- A configuration property list

The property list file for each start-up item is named *StartupParameters.plist*. The property list specifies the name of the start-up item and the dependencies for the start-up items at multiple levels of granularity. These values indicate the services that the start-up item provides, which services must be run before the start-up item, and which services the start-up item uses (if available). SystemStarter reads and processes the dependency information in all property lists and determines the order in which to run the scripts or programs in the folders.

Sybase-provided *StartupItems*

Included on the software distribution is a sample of a start-up item that can be used to automatically boot Adaptive Server during operating system restart. This start-up item, found in *$SYBASE/Install/StartupItems*, is called *SybaseAdaptiveServerEnterprise*. Copy this directory and its content to */Library/StartupItems*. The result should be a new directory, called */Library/StartupItems/SybaseAdaptiveServerEnterprise*, which contains a start-up script called *SybaseAdaptiveServerEnterprise*, plus a property list. This shell script runs a start-up utility called startserver. As installed, this script starts a server named SYBASE. If your server name is not SYBASE, then edit the script, replacing SYBASE with *hostname*_ASE, prior to copying everything to */Library/StartupItems*.

Starting XP Server after initial installation

If you install Adaptive Server and XP Server in the same build session, srvbuild automatically adds information about the XP Server to the sysservers table of Adaptive Server. If you install XP Server in a different build session than the Adaptive Server installation, you are prompted during the XP Server installation process to supply the name of the related Adaptive Server and the System Administrator’s name and password. This information is required by Adaptive Server to start XP Server.

If you do not supply this information during the installation process, XP Server cannot run. When you execute an extended stored procedure (ESP), you see an error message similar to the following:

```
Msg 11018, Level 16, State 1:
Procedure 'xp_cmdshell', Line 2
XP Server must be up for ESP to execute.
```

To add this required information to the sysservers table manually, enter:

```
sp_addserver servername_XP, NULL, servername_XP
```

where *servername* is the name of the Adaptive Server.

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:

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

The default for shutdown uses the with wait option, which 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 must stop Adaptive Server immediately, you can use shutdown with nowait which neither waist for currently executing statements to finish, nor performs checkpoints in every database.

Note Using the shutdown with nowait command is not recommended; use it only when necessary.

Stopping Backup Server

To shut down a Backup Server:

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

   ```
   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 must stop Adaptive Server or Backup Server immediately, use shutdown with nowait, which neither waits for currently executing statements to finish, nor performs 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 *Reference Manual*.

Stopping Monitor Server

Note Monitor Server is not available on Mac OS X.

To release resources, you must shut down Monitor Server before you restart Adaptive Server. If Adaptive Server stops, make sure that Monitor Server is shut down before you restart Adaptive Server. 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 *Monitor Server User's Guide*.

Using the kill command

Warning! Use the kill command to stop 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, 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

When Adaptive Server starts, it creates *SERVER_NAME.krg* file in *$SYBASE* to store information about shared memory segments that it uses.

If Adaptive Server is configured with a memory size that is greater than the MAXSHMSEGSIZE parameter in the operating system then Adaptive Server may create additional shared memory segments and for every additional shared memory segment that it creates, an additional file with *SERVER_NAME.srg[N]* (where *N* ranges from 0 – *N*), is created under *$SYBASE*.

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.

If Adaptive Server or Backup Server is killed abnormally it also leaves shared memory segments. Use the ipcs and ipcrm commands to identify and remove these shared memory segments that have “NATTACH” count as “0”.

See the UNIX man pages for more information about ipcs and ipcrm.

CHAPTER 3

Configuring the Operating System

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 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).

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

You may want to verify that the Adaptive Server environment variables in Table 3-1 are set correctly.

To check the current value of environment variables, enter the following command at the operating system prompt:

```
env
```

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

Table 3-1: System environment variables

Variable	Sample value	Function
DSLISTEN	TEST	Used at Adaptive Server start-up; enables the specified Adaptive Server to listen for input from front-end software.
DSQUERY	TEST	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.

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 *$SYBASE/$SYBASE_ASE/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 HP Tru64

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 HP Tru64 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 programatically, 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. The limit is 10,000 for HP-UX on a 32-bit system, and 60,000 on a 64-bit system.

To obtain the current values on the file descriptors use the Korn or Bourne shell ulimit command: `ulimit -n`

For AIX

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
```

For Linux

The number of file descriptors per process is limited to 10,000. You can set the number of file descriptors using ulimit.

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. The limit is 10,000.

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.

For Silicon Graphics IRIX

Two values must be changed:

- rlimit_nofile_cur – specifies the current number of file descriptors available to a process.
- rlimit_nofile_max – specifies the maximum value rlimit_nofile_cur may hold.

To change these values, enter:

```
#  /usr/sbin/systune -i
```

Updates are made to running systems and to */unix.install*.

```
systune-> rlimit_nofile_max 1024
rlimit_nofile_max = 2500 (0x9c4) ll
Do you really want to change rlimit_nofile_max to 1024
(0x400)? (y/n)  y
```

Restart the server for the change to rlimit_nofile_max to become effective.

For Macintosh OS X

Adaptive Server automatically sets the maximum file descriptors per process to the operating system maximum of 10,240. The ulimit command is not supported on Mac OS X, but you can use the sysctl command, or programatically via getrusage().

Adaptive Server can use a maximum of 10,24 file descriptors per process (engine), regardless of the operating system limit.

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

Increasing the hard limit

To increase the hard limit, use a program like the sample program shown in "Sample program" on page 43.

v **Setting up the sample program to increase the hard limit**

1 Create *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 in "Sample program" on page 43.

2 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.

3 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.

4 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 *RUN_server_name* file.

Sample program

Note This is an sample script; modify it as necessary.

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

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

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

Enabling asynchronous disk I/O

For HP Tru64 UNIX

Enable asynchronous disk I/O.

This step is recommended for HP Tru64 users.

HP Tru64 automatically enables asynchronous I/O but, by default, limits the number of concurrent I/O requests to 64. If you expect that you need additional requests for your Adaptive Server installation, add the following lines to the */etc/sysconfigtab* file:

```
rt:
aio-max-num=1024
```

To determine the current value of aio-maximum, enter:

```
/sbin/sysconfig -q -rt
```

For HP-UX

Enable asynchronous disk I/O.

This step is mandatory for HP users.

To improve I/O performance on character or raw and block devices, enable asynchronous I/O by installing the HP asynchronous I/O driver from SAM. For help with installing this driver, contact your operating system administrator or HP technical support.

Note Before executing the following instructions, shut down Adaptive Server (or SQL Server).

To enable asynchronous I/O:

1. From the SAM Kernel Configuration menu, choose Drivers and set the Pending State for *asyncdisk* to In by adding the driver.

   An alternate step could be done by adding the *asyncdsk* subsystem keyword to */stand/system*.

2. Rebuild the kernel, and reboot the system.

3. Execute the following commands using the userid root:

   ```
   #/etc/mknod/dev/async c 101 4
   #chmod 0660/dev/async
   #chown <uid> /dev/async
   #/etc/setprivgrp <ugrp> MLOCK
   ```

   Where:

<uid> is the identification (user ID)used by the user that is booting Adaptive Server.

<ugrp> is the user group for the *<uid>* userid.

4 At the UNIX prompt, execute the following statements as "root". The user ID of the user who is starting Adaptive Server and Backup Server must be the owner of the */dev/async* directory.

HP-UX 11.23

Enable asynchronous disk I/O.

This step is mandatory for HP users.

To improve I/O performance on character or raw block devices, enable asynchronous I/O by installing the HP asynchronous I/O driver from SAM. For help with installing this driver, contact your operating system administrator or HP technical support.

Note Before executing the following instructions, shut down Adaptive Server (or SQL Server).

1 To launch the Kernel Configuration use kcweb -F.

2 Select Modules.

3 Select asyncdsk and change the next boot to static.

4 Rebuild the kernel, and reboot the system.

5 Execute the following commands using the userid root:

```
#/etc/mknod/dev/async c 101 4
#chmod 0660/dev/async
#chown <uid> /dev/async
#/etc/setprivgrp <ugrp> MLOCK
```

Where:

<uid> is the identification (user ID)used by the user that is booting Adaptive Server.

<ugrp> is the user group for the *sybase* userid.

6 At the UNIX prompt, execute the following statements as "root". The user ID of the user who is starting Adaptive Server and Backup Server must be the owner of the */dev/async* directory.

HP-UX on Itanium

To verify that the async I/O is active on the Itanium machine:

1 Log in as root.

2 Issue the command:

```
# kcmodule -a l grep aync
```

Result: `asyncdsk static required`

or you can also use the following:

```
# ls /dev/asyn*
idev/async /dev/asyncdsk
```

For IBM AIX

Enable asynchronous disk I/O.

This step is mandatory for IBM users.

Enable asynchronous I/O by adjusting the kernel parameters, using the System Management Interface Tool (SMIT):

1 Enter “smit” at the UNIX prompt.

2 From the Devices menu, select Asynchronous I/O.

3 Select Change/Show Characteristics of Asynchronous I/O.

4 Enter the values from Table 3-2:

Table 3-2: SMIT parameters for asynchronous I/O

SMIT parameter	Values for RS/6000 AIX 4.3.3
MINIMUM number of servers	1
MAXIMUM number of servers	10
MAXIMUM number of REQUESTS I/O	4096
Server PRIORITY	39
ASYNC I/O STATE (to be configured at system restart)	Available

If your system uses more than seven disks at the same time for Asynchronous I/O, increase the *MAXIMUM number of servers* value by 1 for every active device.

Note Setting the minor device number on */dev/async* to 4 reduces or eliminates transient 605 errors in Adaptive Server.

If you are using LVM mirroring, set the minor device number to 0x0.

For Silicon Graphics IRIX

Silicon Graphics IRIX has two methods of executing asynchronous I/O:

- For asynchronous I/O Adaptive Server uses the functions found in *libC.so*. Customers enable asynchronous I/O by turning on the allow sql server asynchronous i/o flag in the *.cfg* file. This is commonly known as uaio.

  Prior to Adaptive Server 12.5, uaio used sproc()s. These were visible in the ps -elf listing.

  In Adaptive Server version 12.5 and later the asynchronous I/O functions now use pthreads rather than sproc()s; you no longer see the extra pocesses. The ps -T command shows the pthreads.

  Uaio functions are configurable using the environment variable AIO_THREADS to designate the number of pthreads per engine. The AIO_THREADS environment variable is set by the System Administrator.

  Uaio is implemented through POSIX threads (pthreads). Dataserver engines create the pthreads at start-up. These pthreads run in user space and interact with the operating system AIO library in kernel space.

  Although each engine can queue many I/O operations, the I/O requests are passed one at a time to the AIO library by each pthread. Increase I/O parallelism by increasing the number of pthreads.

  The value of AIO_THREADS depends on the number of database devices and the number of CPUs on the system.

  The default value for AIO_THREADS is 5. Although there is no maximum limit, Sybase recommends that you do not set AIO_THREADS to more than 75 per engine. As a general rule, there should be 1 pthread per 1.5 drives. For example, for a machine with 10 drives, set AIO_THREADS to 15.

  The number of processes for the AIO_THREADS environment variable is *n*+2 for each engine running, where *n* is the value of AIO_THREADS. For example, if AIO_THREADS is set to 5, the number of processes is 7. If two engines are running, the number of processes is 14.

  You must restart Adaptive Server for this environment variable to take effect.

- The SGI Database Accelerator package (DBA) moves asynchronous I/O from user-space into kernel space (commonly called KAIO). If you elect to use KAIO, you must install DBA (version 3.0 or higher) and set an environment variable in the server start-up script.

KAIO only applies to raw (character special) devices, but coexists with file system asynchronous I/O. Databases located on file system devices use uaio, while raw devices use KAIO.

To confirm that KAIO is in use issue ps -T. (Make sure that no file system devices are being used, because file system device use causes pthreads to display in the ps -T listing.)

Another way to confirm that KAIO is in use is to issue the command par -i -n syssgi on the running dataserver. If SGI_KAIO_READ or SGI_KAIO_WRITE display, then KAIO is active.

It is recommended that all devices be raw unless there is heavy use of tempdb. If there is heavy tempdb use, you should elect to use cooked devices for the buffering. You can also create an xlv and not build a file system so that KAIO can be used.

In addition to setting the allow sql server async i/o flag to true in the *.cfg* file, the RUN_SERVER file should set the environment variable __SGI_USE_DBA to one indicating that KAIO is being used. The following is an example RUN_SERVER file:

```
#!/bin/sh
# set up kaio
__SGI_USE_DBA=1;
export __SGI_USE_DBA
SYBASE=/usr/sybase
$SYBASE/bin/dataserver -d/dev/rdsk/dks0d6s1 \
-sSYBASE -e$SYBASE/install/errorlog_SYBASE \
-i$SYBASE
```

Determine the location, type, and size of each database device.

You must provide this information during the installation of Adaptive Server. Devices are operating system files or portions of a disk (called raw partitions) used to store databases and database objects.

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 HP Tru64

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 lowercase "L", not the number 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
```

For Macintosh OS X

To view the keepalive value, use sysctl, in */usr/sbin/sysctl*:

```
% sysctl net.inet.tcp.keepidle
```

Set the KEEPALIVE value, using sysctl:

```
% sudo sysctl -w net.inet.tcp.keepidle=n
```

where *n* is the keepalive interval.

The interval is the value of a timer that is counted down 2 times a second. Therefore, to set the keepalive interval to 15 minutes, use 60*15*2, or 1800.

Note You must have System Administration privileges to set keepalive. systole must be run under `'sudo'`.

For Linux

To display the timeout value, enter:

```
/sbin/sysctl -e net.ipv4.tcp_keepalive_time
```

To reduce the timeout period to 15 minutes (900 seconds,) enter:

```
/sbin/sysctl -w net.ipv4tcp_keepalive_time=900
```

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 HP Tru64

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 options 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 SGI

Check the */var/adm/SYSLOG* file on a regular basis. You can view the file directly. See the operating system documentation for more information.

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.

For Macintosh OS X

You can read operating system error messages using the command line utility /sbin/dmesg. You can also review messages by examining the file */var/log/system.log*.

For Linux

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

Monitoring the use of operating system resources

The *System Administration Guide* discusses maintaining the optimal number of Adaptive Server engines for your workload and system configuration. To determine the optimal number, monitor system and CPU usage.

For HP Tru64

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 dataserver, 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, some of which are described as follows:

- 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 buffers have been configured for network traffic.

    - no -a command 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 and Linux

Sun Solaris and Linux 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 dataserver-, 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.

For SGI

SGI supplies the following tools for monitoring performance:

- The osview command monitors various activities of the operating system and displays them on a full screen.
- The gr_osview command provides a graphical representation of system resources.
- The sar command reports usage of system resources on a line basis.
- The ps command provides a snapshot of accumulated CPU time and usage for individual processes.

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

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

For Macintosh OS X

Macintosh OS X includes a number of command line tools to assist you in evaluating the use of the operating system, including:

- top
- iostat
- netstat

Additional utilities are in the */Applications/Utilities* directory:

- CPU Monitor
- Process Viewer

A sample C shell maintenance script

Running dbcc checks and performing database backups protect the integrity and recoverability of your Adaptive Server databases. The following sample C shell script calls several isql scripts to help you do this:

```
#!/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
```

This example executes a C shell script called dbcc_ck every morning at 2:00 a.m.

Addressable Memory Limits

The following table lists the upper limits of addressable shared memory for Adaptive Server versions 12.0.x and 12.5.x:

Table 3-3: Addressable memory limits by platform

Platform	32-bit Adaptive Server	64-bit Adaptive Server
HP-UX 11.x (PA-RISC processor)	2.75 giga bytes	16 exabytes[1]
HP-UX 11.x (Itanium2 processor)	N/A	16 exabytes
HP Tru64 UNIX 5.x	N/A	16 exabytes
IBM AIX 5.x	2.75 gigabytes	16 exabytes
SGI IRIX 6.x	4 giga bytes	16 exabytes
Sun Solaris 8 (sparc processor)	3.78 gigabytes	16 exabytes
Sun Solaris 8 (Intel x86 processor)	3.75 gigabytes	N/A
Windows NT/Windows 2000[2]	2 gigabytes	N/A
Red Hat Enterprise Linux (Intel x86 processor)	2.7 gigabytes	N/A
Red Hat Enterprise Linux (Itanium2 processor)	N/A	16 exabytes
Mac OSX	2 gigabytes	N/A

[1] One exabyte equals 2^{60}, or 1024 PetaByte. 16 exabyte is a theoretical limit; in practice, the actual value is limited by the total memory available on the system. Adaptive Server has been tested with a maximum of 256 GB of shared memory.

[2] Starting Windows NT with the /3G option allows Adaptive Server to use up to 3 gigabytes of shared memory. For more information, see your Windows NT documentation.

Note The 12.5.x versions of Adaptive Server allocates memory differently than previous releases. This includes changes to existing memory-related configuration parameters and introduces new memory parameters. Review the new memory configuration parameters for Adaptive Server version 12.5.x before modifying the server or operating system memory parameters. For more information, see *What's New in Adaptive Server Enterprise?* and the *System Administration Guide* for details.

Note If you are running HP Tru64 UNIX, if the total shared memory size specified to the application is greater than 2 GB, you can specify a value of 2139095040 (or 0x7f800000) for the value ofthe shm_max attribute. This is the maximum value (2 GB minus 8 MB) that you can specify for the shm_max attribute and still share page table entries.

Each operating system has a default maximum shared-memory segment (for example, shm-max on HP Tru64). Make sure the operating system is configured to allow the allocation of a shared-memory segment at least as large as Adaptive Server's total memory (for Adaptive Sever version 12.0.x) or max memory (for Adaptive Server version 12.5.x). For more information, see the Adaptive Server Installation Guide.

CHAPTER 4

Adaptive Server Default Configuration

When you install or upgrade Adaptive Server, it includes some default parameter settings and a few of its auxiliary programs.

After installing and testing this “default” Adaptive Server, you can configure it to your system’s needs and install other optional features.

For information about configuring Adaptive Server, Backup Server, and Monitor Server see the *System Administration Guide*.

For information about configuring languages, character sets, and sort orders, as well as optional features, see the *System Administration Guide*.

Default settings

After installation, Adaptive Server default settings are as listed in Table 4-1. You may need to configure these settings to suit your computer and database needs.

Table 4-1: Defaults for Adaptive Server parameter settings

Item	Default value
Name	*Servername*
Transport Type	TCP/IP
Port number	4100 For Mac OS X –11222
Error log path	*$SYBASE/$SYBASE_ASE/install/servername.log*
Event logging	Not configured
International support (Localization)	
• Language	us_english

Item	**Default value**
• Character set	DUNIX – ISO 8859-1 HP – Roman8 IBM – ISO 8859-1 Sun – ISO 8859-1 Mac OS X – iso_1 Linux – iso_1
• Sort order	Linux – Binary ordering
Login security mode	Standard

Table 4-2 lists the default settings for the Backup Server, Monitor Server, and XP Server. For more information about these servers, see "Introduction" on page 1.

Table 4-2: Defaults for Backup, Monitor, and XP Servers

Server	**Item**	**Default value**
Backup Server	Name	*AdaptiveServername*_back
	Network support	(TCP/IP)
	Socket number	4200 For Mac OS X – 11223
	Error log path	*$SYBASE/$SYBASE_ASE/install/AdaptiveServername_back.log*
Monitor Server	Name	*AdaptiveServername*_mon
	Network support	(TCP/IP)
	Socket number	4300
	Error log path	*$SYBASE/$SYBASE_ASE/install/AdaptiveServername_back.log*
XP Server	Name	*$SYBASE/$SYBASE_ASE/ADAPIVESERVERNAME*_XP
	Network support	(TCP/IP)
	Socket number	4400 For Mac OS X – 11224
	Error log path	N/A

CHAPTER 5

Setting Up Communications Across the Network

Adaptive Server can communicate with other Adaptive Servers, Open Server applications, and client software across a network. Clients can communicate with one or more servers, and servers can communicate with other servers via remote procedure calls.

Directory services contains information about the network locations of servers. Directory services contain entries for all Adaptive Servers, Backup Servers, and other server products on the network.

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 directory services for the network location of the target server.

Directory services list the name and address of every server, including Backup Server, Monitor Server, and XP 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 directory services and connects to that server.

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.

Table 5-1 shows where to find more information on server and client interfaces file tasks and topics.

Table 5-1: 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 7, "Customizing Localization for Adaptive Server"
	Creating a master interfaces file for multiple installations	"Creating a master interfaces file" on page 70
	Configuring for multiple networks	"Configuring interfaces files for multiple networks" on page 71.
	Reference information	"Understanding the format of the interfaces file" on page 67.
PC-client	Configuring a client	*Installation Guide* for your platform
	Reference information and instructions for advanced tasks	*Open Client and Open Server Programmer's 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 and Open Server Programmer's Supplement* for your PC-client platform, or the appropriate Open Client documentation

How Adaptive Server determines which directory service entry to use

Adaptive Server uses directory services to determine the address at which it should listen for clients. When you start Adaptive Server, it performs the following steps:

1. It looks for the server name supplied in the command line -s option. If the server name is not supplied in the command line:
2. It determines its own name by checking the value of the DSLISTEN environment variable. If the DSLISTEN environment variable is not set, then it assumes that the server name is SYBASE.

3. Looks in directory services for an entry whose name matches the name found in the steps above.
4. It uses the network information provided by the directory services entry it has found to listen for client connections.

How a client uses directory services

When 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 directory services for an entry whose name matches the name of the server.
- Uses the network information provided by the directory services entry 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 directory services. 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 network address entry for the server.

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.

Creating a directory services entry

The installation program, srvbuild, automatically creates a directory services entry for each server installation. You can also use the following Sybase utilities to edit the network information in directory services:

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

For details on using these utilities, see the Adaptive Server *Utility Guide*.

Supported directory drivers

There are three supported drivers:

- interfaces driver
- Lightweight Directory Services driver.
- Cell Directory Service (CDS) provided by Distributed Computing Environment (DCE)

This remainder of this chapter describes the *interfaces* file and provides specific configuration information for each supported UNIX platform. For information about LDAP drivers, Cell Directory Services, and for a comparison between interfaces files and LDAP directory services, see the *Open Client/Server Configuration Guide* for your platform.

Contents of an interfaces file

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

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 67 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.

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 a master interfaces file" on page 70.

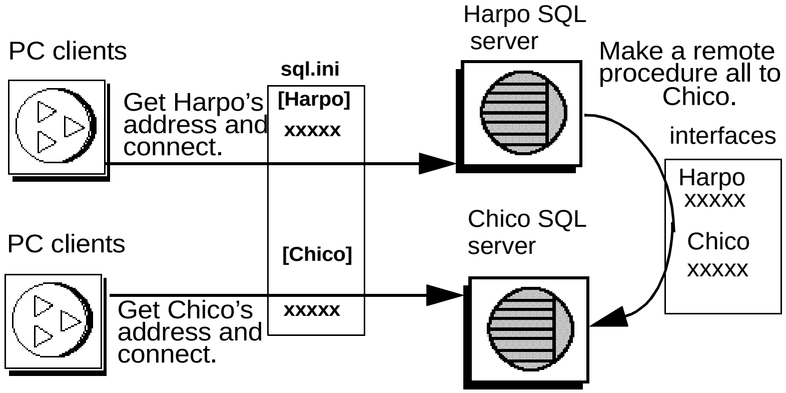
If the host machine supports multiple networks, see "Configuring interfaces files for multiple networks" on page 71.

Heterogeneous and homogeneous environments

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

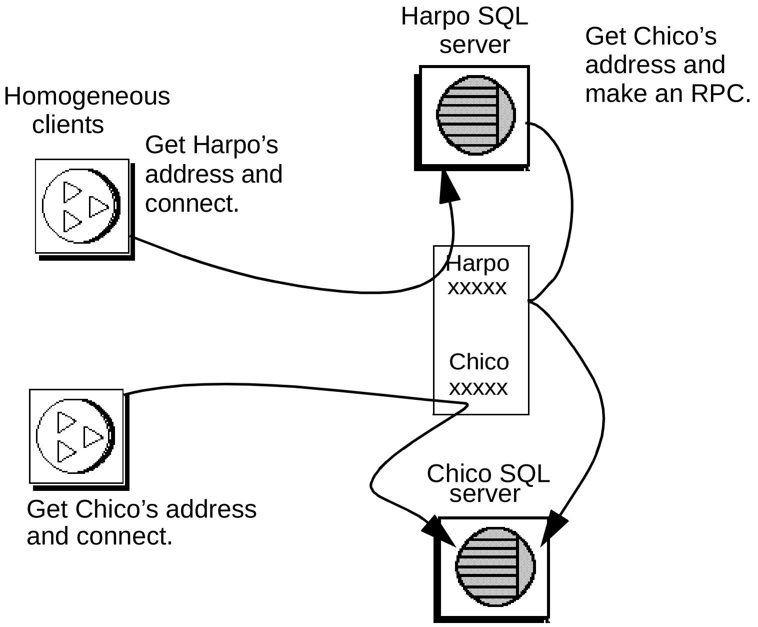
If the platforms are different, each platform may require a different format and configuration for its *interfaces* file. Figure 5-1 illustrates how a client PC uses network information in its interfaces file (*sql.ini*) to connect to Adaptive Server running under UNIX, and how Adaptive Server uses its *interfaces* file to connect to another server during a remote procedure call.

Figure 5-1: Establishing network connections in a heterogeneous environmen



If both a client and a server are running under UNIX, the same interfaces file is valid for both. Figure 5-2 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 5-2: Establishing network connections in a homogeneous environment


Harpo SQL server
Get Chico's address and make an RPC.
Homogeneous clients
Get Harpo's address and connect.
Harpo
xxxxx
Chico
xxxxx
Chico SQL server
Get Chico's address and connect.

Understanding the format of the *interfaces* file

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.

There are two interfaces file entry formats, TLI and TCP.

A TLI style entry looks like:

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

A TCP style entry looks like:

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

Components of an interfaces file entry

Table 5-2 describes the components of an interfaces file entry.

Table 5-2: 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 • query
api	Application programming interface available to the network.The supported value is tli.
protocol	Name of the network protocol. Protocol types available are: • TCP/IP, represented by the letters “tcp”
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`

Component	Value
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, the device 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 stream devices corresponding to different protocols in the TCP protocol suite. Choose the TCP streams device. Common TCP streams devices is */dev/tcp*.
address for a TLI protocol entry	Address consists of: • Address prefix, "\x" for TLI. • Network type, always 0002. • Port number converted to four digits, hexadecimal. Must be a unique 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 the port number. • IP network node address of the host machine converted to 8 digits, hexadecimal. • Trailing zeros, optional, 16 digits.
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.
ha_failover	An entry created in directory services or the interfaces file for high availability.
filter	Adaptive Server supports Secure Socket Layers (SSL) as a filter which is appended to the master and query lines of the directory services. SSL is the standard for securing the transmission of sensitive information.

Creating a master *interfaces* file

A master *interfaces* file contains entries for all Sybase servers on the network. It can be used with every server and client connected to the network. By distributing copies of a master *interfaces* file, you can ensure that all Sybase products on the network interact with one another.

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 can 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

Note For Mac OS X, you can use a utility called Server Discovery to add entries to the *interfaces* file. For information, see the *Quick Install for Adaptive Server on Mac OS X*.

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 Adaptive Server *Utility Guide*.

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. 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
```

When Adaptive Server restarts, 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 are 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 the same DSQUERY name for all clients. The interfaces files on the different machines contain different network names.
- Use different DSQUERY names for the clients. The interfaces files on all the machines are the same, but they contain multiple DSQUERY 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 looks like the following example on the "engineering" network:

```
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 interfaces file looks like the following example on the "corporate" network:

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_CORPNET 4559
<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 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 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 primary network is “corporate network” and backup is “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
<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 with the following message, the port number specified in the interfaces file may be in use:

```
00:00000:00002:2003/09/22 12:37:23.63 kernel  network name SERV_CORPNET, type
ether, port 4559, filter NONE
00:00000:00002:2003/09/22 12:37:23.65 kernel  ninit: bind, Address already in
use
00:00000:00002:2003/09/22 12:37:23.68 server  Error: 1602, Severity: 18, State:
2
00:00000:00002:2003/09/22 12:37:23.68 server  Unable to initialize network 0
00:00000:00002:2003/09/22 12:37:23.68 kernel  ninit: All master network
listeners have failed.  Shutting down.
00:00000:00002:2003/09/22 12:37:23.68 kernel  ueshutdown: exiting
00:00000:00016:2003/09/22 16:11:35.46 server  SQL Server shutdown by request.
```

v **Investigating 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 does 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 the Adaptive Server *Utility Guide*.

v **If a stale server process 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 the *Utility Guide*.

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 cannot start because the port number may be in use by another process. Use the netstat command 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.

If you find a process using the same port number, you can do one of the following:

- Change the interfaces file to use a new port number for the XP Server.
- Stop the process using the port number allotted to XP Server.

Restart Adaptive Server, and execute the ESP that you attempted earlier. XP Server should start automatically.

CHAPTER 6

Using the Lightweight Directory Access Protocol as a Directory Service

For information on using Lightweight Directory Access Protocol (LDAP) as a directory service, see *Open Client/Server Configuration Guide*.

Note LDAP is not supported for Macintosh OS X.

Performance

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

Migrating from the interfaces file to LDAP

There is no direct upgrade path from the interfaces file to lightweight directory services. If you are upgrading a pre-12.5 server to an 12.5 and later version of Adaptive Server, you must first upgrade the server according to the directions in the *Installation Guide* for your platform.

If you have LDAP or other directory services defined in the *libtcl.cfg* file before configuring the server, the -i argument is not added to the *RUN_servername* file.

If you do not have LDAP or other directory services defined in the *libtcl.cfg*, the -i argument is added to the *RUN_servername* file.

Once you have completed the upgrade, you can configure your server to use the lightweight directory service.

1. Shut down the server. See Chapter 2, "Starting and Stopping Servers."
2. Using a standard ASCII text editor, edit the *$SYBASE/$SYBASE_OCS/config/libtcl.cfg* file to add or remove the directory service.
3. Use dsedit or dscp and add the server entry to directory service.
4. Using a standard ASCII text editor, edit *$SYBASE/$SYBASE_ASE/install/RUN_servername* file to add or remove the -i argument.
5. Restart the server from a command line by executing the file *$SYBASE/$SYBASE_ASE/install/RUN_servername*

   where *servername* is the name of the server.

CHAPTER 7

Customizing Localization for Adaptive Server

This chapter provides information about Sybase localization support for international installations, including configuring languages, character sets, and sort order. 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 – Adaptive Server includes language modules for:
    - Brazilian Portuguese
    - Chinese (Simplified)
    - French
    - German
    - Japanese
    - Korean
    - Polish
    - Spanish
    - Thai
- Translated documentation – translated documentation is available in:
    - Chinese (Simplified)
    - French
    - German
    - Japanese
    - Korean

Language modules

Adaptive Server stores its localized software messages in separate language modules.

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.

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

Default character sets for servers

The default character set is the character set in which data is encoded and stored on the Adaptive Server databases.

Changing the default language and character set

Warning! Make all changes to the 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 require additional steps. To change the character set or sort order after you have added data, see the *System Administration Guide*.

When you installed Adaptive Server, srvbuild displays a message box asking if you want to localize your Adaptive Server to a language other than us_english and if you want to use a character set or sort order other than the default.

srvbuild creates an Adaptive Server with the following defaults:

- us_english language
- iso_1 character set (on HP-UX platforms, use Roman8)
- Binary sort order

Valid language options depend on what language modules were unloaded from the distribution media onto your system.

All character sets are copied from the distribution media by default.

You can:

- Click No to accept the defaults.
- Click Yes to change the defaults.

  The sqlloc menu is displayed. sqlloc is the GUI utility used to change default languages, character sets, and sort orders.

By default, when Adaptive Server and Backup Server are installed on HP Tru64, IBM, and SUN Solaris systems, the installation installs the character set files for ISO 8859-1, which supports the Western European languages.

By default, when Adaptive Server and Backup Server are installed on HP systems, the installation installs the character set files for ROMAN8, which supports the Western European languages.

Note sqlloc is not available on Mac OS X.

Use the *charset* script to change the character set and sort order. Use the langinstall utility to install languages, and sp_modifylogin to specify language preferences.

Changing the default character set for servers

You can select any character set as the default on Adaptive Server, including character sets that are not the platform default character sets. Keep the following guidelines in mind when selecting a new default character set:

- To avoid conversion errors or overhead, determine the default character set based on the character set used by your clients.

  For example, if most of your clients use ISO 8859-1, you can minimize the amount of data conversion that has to occur by specifying ISO 8859-1.

- If your server is operating in a heterogeneous language environment, choose a character set that works with all the character sets needed. Often, this is Unicode (UTF-8).

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 can cause incorrect behavior. To change the character set or sort order after you have added data, see the *System Administration Guide*.

Supported character sets

The following language, scripts and character sets are supported by Adaptive Server:

- Arabic – see Table 7-1 on page 85.
- Baltic – see Table 7-2 on page 86.

- Chinese, Simplified – see Table 7-3 on page 86.
- Chinese, Traditional – see Table 7-4 on page 86
- Cyrillic – see Table 7-5 on page 86.
- Eastern European – see Table 7-6 on page 87.
- Greek – see Table 7-7 on page 87.
- Hebrew – see Table 7-8 on page 87.
- Japanese – see Table 7-9 on page 87.
- Korean – see Table 7-10 on page 87.
- Thai – see Table 7-11 on page 88.
- Turkish – see Table 7-12 on page 88.
- Unicode (which supports over 650 languages) – see Table 7-13 on page 88.
- Vietnamese – see Table 7-14 on page 88.
- Western European – see Table 7-15 on page 89.

The tables define each character set and indicate information on whether it requires Unilib conversion (Unilib Required column).

- Check mark (x) – the character set requires Unilib conversion.
- No check mark – the character set may use either the Unilib conversion or the built-in conversion.

For more information see “Character set conversion” on page 89.

Table 7-1 lists the Arabic character set:

Table 7-1: Arabic character sets

Character set	Unilib required	Description
cp864	X	PC Arabic
cp1256	X	Microsoft Windows Arabic
iso88596	X	ISO 8859-6 Latin/Arabic

Table 7-2 lists the Baltic character set:

Table 7-2: Baltic character sets

Character set	Unilib required	Description
cp1257	X	Microsoft Windows Baltic

Table 7-3 lists the simplified Chinese character set:

Table 7-3: Simplified Chinese character sets

Character set	Unilib required	Description
eucgb	X	EUC GB encoding = Simplified Chinese character sets
cp936	X	Microsoft Simplified Chinese character sets
gb18030	X	PRC 18030 standard

Table 7-4 lists the traditional Chinese character set:

Table 7-4: Traditional Chinese character set

Character set	Unilib required	Description
cp950	X	PC (Microsoft) Traditional Chinese
euccns	X	EUC CNS encoding = Traditional Chinese with extensions
big5	X	Big 5 Traditional Chinese
big5hk	X	Big 5 with HKSCS extensions

Table 7-5 lists the Cyrillic character set:

Table 7-5: Cyrillic character sets

Character set	Unilib required	Description
cp855		IBM PC Cyrillic
cp866		PC Russian
cp1251		Microsoft Windows 3.1 Cyrillic
iso88595		ISO 8859-5 Latin/Cyrillic
koi8		KOI-8 Cyrillic
mac_cyr		Macintosh Cyrillic

Table 7-6 lists the Eastern European character set:

Table 7-6: Eastern European character sets

Character set	Unilib required	Description
cp852		PC Eastern Europe
cp1250		Microsoft Windows 3.1 Eastern European
iso88592		ISO 8859-2 Latin-2
mac_ee		Macintosh Eastern European

Table 7-7 lists the Greek character set:

Table 7-7: Greek character sets

Character set	Unilib required	Description
cp869		IBM PC Greek
cp1253		MS Windows Greek
greek8		HP GREEK8
iso88597		ISO 8859-7 Latin/Greek
macgrk2		Macintosh Greek

Table 7-8 lists the Hebrew character set:

Table 7-8: Hebrew character sets

Character set	Unilib required	Description
cp1255	X	Microsoft Windows Hebrew
iso88598	X	ISO 8859-8 Hebrew

Table 7-9 lists the Japanese character set:

Table 7-9: Japanese character sets

Character set	Unilib required	Description
cp932	X	IBM J-DBCS:CP897 + CP301 (Shift-JIS)
deckanji		HP Tru64 JIS encoding
eucjis		EUC-JIS encoding
sjis		Shift-JIS (no extensions)

Table 7-10 lists the Korean character set:

Table 7-10: Korean character sets

Character set	Unilib required	Description
eucksc	X	EUC KSC Korean encoding = CP949

Table 7-11 lists the Thai character set:

Table 7-11: Thai client character sets

Character set	Unilib required	Description
tis620	X	TIS-620 Thai standard
cp874	X	Microsoft Windows Thai

Table 7-12 lists the Turkish character set:

Table 7-12: Turkish character sets

Character set	Unilib required	Description
cp857		IBM PC Turkish
cp1254		Microsoft Windows Turkish
iso88599		ISO 8859-9 Latin-5 Turkish
macturk		Macintosh Turkish
turkish8		HP TURKISH8

Table 7-13 lists the Unicode character set:

Table 7-13: Unicode character set

Character set	Unilib required	Description
utf8	X	Unicode UTF-8 encoding

Table 7-14 lists the Vietnamese character set:

Table 7-14: Vietnamese character set

Character set	Unilib required	Description
cp1258	X	Microsoft Windows Vietnamese

Table 7-15 lists the Western European character set:

Table 7-15: Western European character set

Character set	Unilib required	Description
ascii8	X	US ASCII, with 8-bit data, ISO 646
cp437		IBM CP437 - U.S. code set
cp850		IBM CP850 - European code set
cp860	X	PC Portuguese
cp858	X	cp850 with Euro support
cp1252	X	Microsoft Windows US (ANSI)
iso_1		ISO 8859-1 Latin-1
mac		Standard Macintosh coding
roman8		HP ROMAN8
iso15	X	ISO 8859-15 Latin-1 with Euro support
roman9	X	HP ROMAN8 with Euro support
mac_euro	X	Standard Macintosh coding with Euro support

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 must 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. For more information, see the *System Administration Guide*.

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 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, 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:

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

For more information about localization files, see “Localization directories” on page 94.

Table 7-16 describes the sort orders that you can specify at installation time or at a later time using the sqlloc utility in HP, and SUN Solaris systems, and the asecfg utility in IBM.

Table 7-16: 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.
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.
Spanish dictionary order, case sensitive	Case-sensitive Spanish dictionary sort order. Use with Spanish and for most Latin American locales.
Spanish dictionary order, case insensitive	Spanish case-insensitive dictionary sort order. Use with Spanish and for most Latin American locales.

Sort order name	Description
Spanish dictionary order case insensitive, accent insensitive	Spanish case-insensitive and accent-insensitive dictionary sort order. Use with Spanish and for most Latin American locales.
Scandinavian dictionary order, case sensitive	Case-sensitive dictionary sort order. Use with Scandinavian languages.
Scandinavian dictionary order, case insensitive, uppercase preference	Case-insensitive and accent-insensitive dictionary sorting, with uppercase preference. Use with Scandinavian languages.

To see the sort orders that are available, use sqlloc to display the sort orders for the character sets you plan to use. For more information on unicode sort orders for utf_8, see Chapter 7, “Configuring Character Sets, Sort Orders, and Languages” in the *System Administration Guide*.

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 into the Sybase installation directory to support the new language. For information about directories, see “Localization directories” on page 94.

Installing a new language module

A full Adaptive Server installation includes all the language components automatically. If you did not select a full install, you must 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 98.

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 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/language_name/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.dat* 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 includes some locale entries already defined 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

Before beginning the edit, make a copy of the original file, in case you have problems with the resulting edited version.

To edit the *locales.dat* file:

1. Open the *locales.dat* file copy in a text editor.
2. Find the section enclosed in brackets:
   - For Sun Solaris, *[sun_svr4]*
   - For HP, *[hp ux]*
   - For IBM, *[aix]*
   - For HP Tru64 *[axposf]*
   - For Mac OS X *[macosx]*

3 Make sure the section contains an entry for the language (*syb_language*) and character set (*syb_charset*) combination that you want to use.

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.

For example, if you want your Open Client messages to appear in French, and Adaptive Server is using the ROMAN8 character set, you would check the *locales.dat* entries for your platform and look for the following entry:

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

4 Add the required entry or modify an existing entry.

5 Save the changes, if any, and exit the text editor.

Changing the localization configuration

By default, the Adaptive Server and Backup Server configurations uses 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.

Adaptive Server localization

Each language uses about 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 Existing 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 Next, supply:

- the default language
- the default character set
- default sort order

6 Select any other languages you want to install. You may select only 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.

Backup Server localization

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

At the time you install the Backup Server, specify the language and default character set specified for Adaptive Server.

Configuring Adaptive Server for other character sets

To configure Adaptive Server with the character set and sort order for your language, complete the following steps. Your system messages appear in the default language, English.

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

To use charset, the server must be running and you must have System Administrator privileges. Use the *file name* of the sort order:

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

Replace *sort_order_file* with the name of the sort order file. See Table 7-17 on page 101. Replace *character_set* with the Sybase name for your character set. See Table 7-18 on page 102.

2 Use charset utility to load any additional character sets. See “charset utility” on page 104 for more about this utility.

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 Use the ID of the sort order to configure your server for the new character set and sort order.

```
1> sp_configure "default sortorder_id",
2> sort_order_id, "character_set"
3> go
```

Replace *sort_order_id* with the ID for your sort order. See Table 7-17 on page 101. Replace *character_set* with the Sybase name for your character set. See Table 7-18 on page 102.

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 starts, rebuilds all the system indexes, then shuts down. Restart a second time to bring the server up in a stable state.

Sort orders

Table 7-17 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 7-17: 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
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 for CP 855.	Dictionary order, case sensitive, accent sensitive	*rusdict.srt*	58
	Dictionary order, case insensitive, accent sensitive	*rusnocs.srt*	59

Language or script	Sort orders	File name	ID
Scandinavian	Dictionary order, case sensitive, accent sensitive	*scandict.srt*	47
These sort orders work only with CP 850.	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	Dictionary order, case sensitive, accent sensitive	*turdict.srt*	72
These sort orders work only with ISO 8859-9.	Dictionary order, case insensitive, accent insensitive	*turnoac.srt*	73
	Dictionary order, case insensitive, accent sensitive	*turnocs.srt*	74

Character sets

Table 7-18 lists the supported character sets and their Sybase name.

Table 7-18: Sybase character set names

Character sets	Sybase name
ASCII 8	acsii_8
Big 5	big5
Big 5HK	big5hk
CP 437	cp437
CP 850	cp850
CP 852	cp852
CP 855	cp855
CP 857	cp857
CP 858	cp858
CP 860	cp860
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

Character sets	Sybase name
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
GB 18030	gb18030
GREEK8	greek8
ISO 8859-1	iso_1
ISO 8859-2	iso88592
ISO 8859-5	iso88595
ISO 8859-6	iso88596
ISO 8859-7	iso88597
ISO 8859-8	iso88598
ISO 8859-9	iso88599
ISO 8859-15	iso5
Koi8	koi8
Macintosh Cyrnillic	mac_cyr
Macintosh Eastern European	mac_ee
Macintosh Greek	macgrk2
Macintosh Western European	mac
Macintosh with Euro support	mac_euro
Macintosh Turkish	macturk
ROMAN8	roman8
ROMAN9	roman9
Shift-JIS	sjis
TIS 620	tis620
TURKISH8	turkish8
UTF-8	utf8

charset utility

Use the charset utility to load character sets and sort orders into Adaptive Server. If you are using charset to load the default character set and sort order, this should be done only at the time of installation.

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

Syntax

```
charset
[ -U username ]
[ -P password ]
[ -S server ]
[ -I interfaces ]
[ -v version ]
sort_order
[ charset ]
```

Table 7-19: 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.
-S	Specifies the name of the server. If not specified, charset uses the DSQUERY environment variable to identify the server name. If there is no DSQUERY environment variable, 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	Causes the Sybase version string to be printed, then exits. Use with no other options specified.
sort_order	When charset is used to load the default character set and sort order, *sort_order* 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	Specifies the directory of the character set to be used by Adaptive Server.

CHAPTER 8

Logging Error Messages and Events

This chapter describes how to use the error logging features of Adaptive Server.

Adaptive Server error logging

Each time Adaptive Server starts, it writes information to a local error log file, called the Adaptive Server error log.

$SYBASE/install/errorlog

This file:

- Stores information about the success or failure of each start-up attempt.
- Logs error and informational messages generated by the server during its operations.
- Remains open until you stop the server process.
- Contains startup messages from Adaptive Server

Note When you want to make more disk space available by reducing the size of the error log, stop Adaptive Server before deleting logged messages. The log file cannot release its memory space until Adaptive Server has stopped.

Enabling and disabling error logging

Logging to the Adaptive Server error log is always enabled. However, when you create or modify a specific user-defined message, you can set it to be omitted from the log. See “Logging user-defined messages” on page 107.

Setting error log paths

The installation program sets the error log location in the Sybase installation directory when you configure a new Adaptive Server. Backup Server and Monitor Server each have their own error logs.

The default location for each server’s error log is:

- Adaptive Server: *$SYBASE/$SYBASE_ASE/install/error.log*
- Backup Server: *$SYBASE/$SYBASE_ASE/install/backup.log*
- Monitor Server: *$SYBASE/$SYBASE/install/ms.log*

At start-up, you can reset the name and location of the Adaptive Server error log file from the command line. Use the -e start-up parameter and value in the dataservercommand to start Adaptive Server.

Note Multiple Adaptive Servers cannot share the same error log. If you install multiple Adaptive Servers, specify a unique error log file name for each server.

Setting the Adaptive Server error log path

You can change the error log path by editing the *$SYBASE/$SYBASE_ASE/install/RUN_server_name* file.

For example, to change the error log path from *$SYBASE/$SYBASE_ASE/bin/dataserver -d/Devices/ASE_2K.dat -sASE_2K -i/ASE_125 -e/$SYBASE/$SYBASE_ASE/install/ASE_2K.log -M/ASE_125* to the *$SYBASE* directory, type:

```
$SYBASE/ASE-12_5/bin/dataserver -d/Devices/ASE_2K.dat
-sASE_2K -i/ASE_125 -e/$SYBASE/ASE_2K.log -M/ASE_125
```

For information about using the *RUN_server_name* file, see “Starting and Stopping Servers” on page 19.

Managing messages

When event logging is enabled, you can manage its functions in the following ways:

- Use sp_addmessage or sp_altermessage to control whether a specific user-defined message is logged in the Adaptive Server error log.

  For the complete syntax for sp_addmessage and sp_altermessage, see the *Reference Manual*.

- Use configuration parameters to specify whether auditing events are logged. Auditing events pertain to a user's success, log audit logon success, or failure, log audit logon failure, in logging in to Adaptive Server.

Logging user-defined messages

You can specify whether a user-defined message is logged to the Adaptive Server error log. Adaptive Server lets you make this determination for:

- New messages (sp_addmessage).
- Existing messages (sp_altermessage).

For more information about these commands and their parameters, see sp_addmessage and sp_altermessage in the *Reference Manual*.

New messages

Include the with_log option in sp_addmessage when you add a new user-defined message to sysusermessages. This parameter sets the Adaptive Server to log the message each time that the message appears.

Existing messages

Include the with_log option in sp_altermessage to change an existing user-defined message. This parameter alters the reporting status of that message:

- TRUE – to enable logging.
- FALSE – to disable logging.

Logging auditing events

By default, Adaptive Server does not log auditing events. However, you can use sp_configure parameters to specify whether Adaptive Server is to log auditing events, such as logins, to the Adaptive Server error log.

Possible parameters and values are:

- log audit logon success at 1 – to enable logging of successful Adaptive Server logins:

  ```
  sp_configure "log audit logon success", 1
  ```

- log audit logon failure at 1 – to enable logging of unsuccessful Adaptive Server logins:

  ```
  sp_configure "log audit logon failure", 1
  ```

- Either parameter at 0 – to disable logging of that message type:

  ```
  sp_configure "log audit logon success", 0
  sp_configure "log audit logon failure", 0
  ```

For more information about sp_configure, see the *System Administration Guide*.

CHAPTER 9

Managing Adaptive Server Databases

The administration of Adaptive Server databases includes both routine tasks and performance and tuning considerations.

- The *System Administration Guide* discusses most of the administrative tasks in detail.
- The *Performance and Tuning Guide* provides in-depth explanations of performance issues.

Managing database devices

The term **database device** refers to a disk or a portion of a disk that stores Adaptive Server databases and database objects.

Device requirements

The size and number of Adaptive Server devices depend on the following constraints:

- The maximum device size is 32GB.
- Each database can have as many as 128 devices.
- The maximum database size is 4TB.

When you install Adaptive Server, the program creates a *.dat* file in the */data* directory of the Sybase installation directory. To use a *.dat* file as a database device, you can either use the default */sybase/data* directory, or create a device and a directory in which to store it. Mac OS X places the devices in */Applications/Sybase/System/device*.

Creating *.dat* files for database devices

If you choose to create a new device, use the disk init command to specify the drive, path, and file name of the database device.

v **Creating a database device:**

1 If the */devices* directory does not exist, create it from the command prompt:

```
    mkdir devices
```

2 Start isql and connect to Adaptive Server using the “sa” account:

```
$SYBASE/$SYBASE_OCS/bin> isql
```

3 Create the device using a disk init statement similar to the following example:

```
disk init name = "user_device1",
physname = "/work/data/device1", size = 2048
```

The previous example creates a 4MB device, as measured in 2K pages, without an actual device number. To use a specific number, run sp_helpdevice to determine the number of an available device, and enter that number.

For more information about sp_helpdevice and disk init command, see the *System Administration Guide* and the *Reference Manual*.

For more information about device files, see the *Performance and Tuning Guide*.

CHAPTER 10

Adding Optional Functionality to Adaptive Server

This chapter provides instructions for adding optional functionality to Adaptive Server:

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

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 consists 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. 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

During auditing configuration, 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.

Note auditinit is not available on Mac OS X.

Pre-installation tasks for auditing devices

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 *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

v **Configuring 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.

10. Restart Adaptive Server for the changes to take effect.

v Creating 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.

To create a device for an audit table:

1 Enter the *full path* of the physical device (raw partition) that you located in “Pre-installation tasks for auditing devices” on page 113.

```
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.
```

2 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:
```

3 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.

4 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
```

5 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
```

v Creating 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*.

v Deleting 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.

v Changing 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.

Installing online help for Transact-SQL syntax

This section provides instructions for installing online help for Transact-SQL syntax.

Online syntax help: *sp_syntax*

The *$SYBASE/$SYBASE_ASE/scripts* directory contains scripts for installing the syntax help database, sybsyntax. You can retrieve this data with sp_syntax. For more information on sp_syntax, see the *Reference Manual*.

The *scripts* directory contains one or more of the sp_syntax scripts shown in Table 10-1, depending on which Sybase products are included with your server:

Table 10-1: sp_syntax installation scripts

Script	Product
ins_syn_cl	Open Client Client-Library™
ins_syn_esql	Embedded SQL™
ins_syn_os	Open Server
ins_syn_sql	Transact-SQL

All Adaptive Server installations include the ins_syn_sql script. This script includes syntax information for Transact-SQL, the system procedures, and the Sybase utilities. When you execute this script, you install the SQL portion of the sybsyntax database.

You can install any of these scripts, depending on the need for Sybase information on your server. The first script you execute creates the sybsyntax database and the needed tables and indexes. Any scripts that you execute after the first one add to the existing information in the database. If you execute a script that was executed previously, the previously installed rows of information are deleted from the table in the database and then reinstalled.

Warning! The *ins_syn_cl* and *ins_syn_os* scripts conflict. If you execute both scripts, errors occur.

Default device for the *sybsyntax* database

The sybsyntax database requires 2MB on your database device. By default, the sybsyntax installation scripts install the sybsyntax database on the device that is designated as the default database device.

If you have not used sp_diskdefault to change the status of the master device (which is installed as the default disk) or to specify another default device, the scripts install sybsyntax on the master device. Sybase does not recommend this configuration because sybsyntax uses valuable space, which is best left available for future expansion of the master database.

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

Use sp_diskdefault to specify a default device other than the master device. For information about sp_diskdefault, see the *Reference Manual*.

- Modify each sybsyntax installation script that you plan to execute to specify a different device, as explained in the following section.

Installing *sybsyntax*

For each sybsyntax installation script you want to execute:

1. Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device where you plan to store the sybsyntax database. You will need to provide this information later.
2. Make a copy of the original script. Be sure you can access this copy, in case you experience problems with the edited script.

3 Use a text editor to edit the script, if necessary, to change the default device from the master device to the device created in step 1. For information on the default device, see “Default device for the sybsyntax database” on page 120.

- Comment out the following section, which specifies the default device:

```
/* create the database, if it does not exist */
 if not exists (select name from sysdatabases
 where name = "sybsyntax")
 begin
     /* create the sybsyntax table if it doesn’t exist */
     /* is the space left on the default database
     devices > size of model? */
     if (select sum (high-low +1) from sysdevices where status
     & 1 = 1) - (select sum(size) from sysusages, sysdevices
         where vstart >= sysdevices.low
         and vstart <= sysdevices.high
         and sysdevices.status &1 = 1) >
         (select sum(sysusages.size) from sysusages
         where dbid = 3)
     begin
         create database sybsyntax
     end
     else
     begin
         print "There is not enough room on the default
         devices to create the sybsyntax database."
     return
     end
 end
```

- After you have commented out this entire section, add a line like this to the script:

```
create database sybsyntax on device_name
```

where *device_name* is the name of the device where you want to install sybsyntax.

4 Execute the script with a command like the following:

```
isql -Usa -Ppassword -Sservername <
$SYBASE/$SYBASE_ASE/scripts/ins_syn_sql
```

where *sa* is the user ID of the System Administrator, *password* is the System Administrator’s password, and *servername* is the Adaptive Server where you plan to install the database.

If you have set the DSQUERY environment variable to the *servername*, you can replace the server name with DSQUERY.

5 To ensure that you have installed the sybsyntax database and that it is working correctly, use isql to log in to the server on which you installed the database, and execute sp_syntax. For example:

```
isql -Usa -Ppassword -Sservername

1> sp_syntax "select"
2> go
```

Adaptive Server displays a list of commands that contain the word or word fragment "select."

CHAPTER 11

Managing Sybase Licenses for Adaptive Server: SySAM Overview

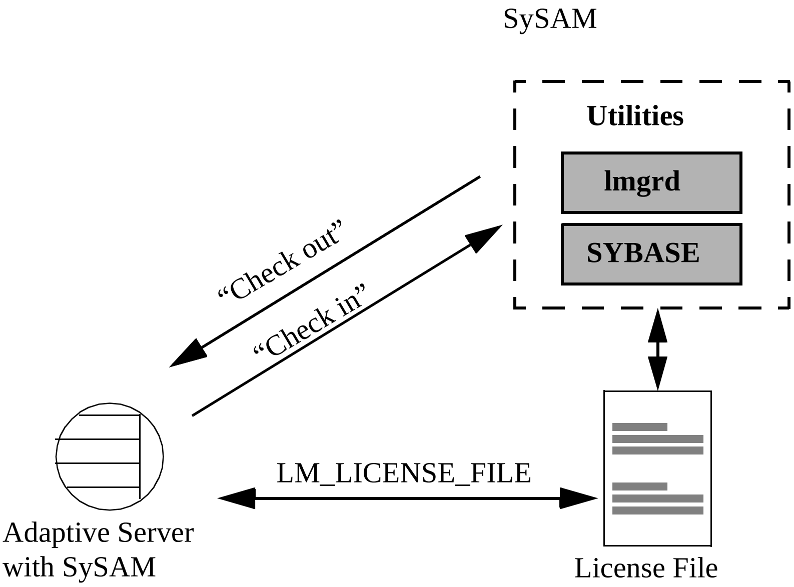
How SySAM works

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 features
- One or more license files
- The SySAM software, which consists of a license management daemon and a Sybase daemon

Figure 11-1 shows the relationship between these components.

Figure 11-1: SySAM components



Adaptive Server check-out procedure

When you start Adaptive Server, it attempts to locate the license file *license.dat* located in *$SYBASE/$SYBASE_SYSAM/licenses*. If the file does not exist, Adaptive Server looks for the license file specified in the environment variable LM_LICENSE_FILE.

The license file contains a pointer to the SySAM software on the primary server known as the **license host**, or to a remote license host where SySAM is running. SySAM consists of a utility, lmutil, to manage licensing activities, and two daemons—the license management daemon, lmgrd, and the SYBASE daemon. The daemons handle requests to check in or check out licensed features, as shown in Figure 11-1.

Using information in the license file, Adaptive Server connects to SySAM and attempts to check out a license (ASE_SERVER) for the base Adaptive Server product. If the ASE_SERVER license is checked out successfully, Adaptive Server continues to start 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 (ASE_DTM) or high availability (ASE_HA), it attempts to check out licenses for those features during the start-up or when the feature is enabled. If a license is not available for an optional feature, Adaptive Server still starts, but the feature cannot be used.

Compatibility with other licensed products, on Silicon Graphics IRIX

If LM_LICENSE_FILE is set at the environment level, it overrides the default locations where the license manager looks for runtime licenses. Other software on your machine (the MIPSPro C compiler and ProDev Workshop Pro, for example) use the same license manager; therefore, setting LM_LICENSE_FILE prevents this software from finding its license.

To avoid this, remove LM_LICENSE_FILE from the InstallShield Installer generated scripts, *SYBASE.sh* and *SYBASE.csh*. Instead, set LM_LICENSE_FILE in your start-up script.

For more information see the *FLEXlm End User Manual* in the online insight documentation.

Adaptive Server features licensed through SySAM

Table 11-1 lists all of the optional features available for Adaptive Server. Some of these feature options have been replaced by feature packages and are no longer available separately. They are listed here for completeness as those upgrading from earlier versions may still have earlier option license.

The Feature name column lists the value of the Feature name field on the Sybase Software Asset Management (SySAM) certificate.

Note Not all features are available on every platform.

Table 11-1: Licensed Adaptive Server features

Feature	Feature name	Description
Adaptive Server Enterprise Edition	ASE_SERVER	The basic Adaptive Server product, without any options

Feature	Feature name	Description
Adaptive Server Small Business Edition	ASE_SBE	The Small Business Edition of Adaptive Server
Adaptive Server Developer’s Edition	ASE_DEV	The Developer’s Edition of Adaptive Server.
High availability	ASE_HA	High availability supports server failover and recovery.
Distributed transaction management	ASE_DTM	Distributed transaction management support for XA and Microsoft DTC protocols.
Java in Adaptive Server	ASE_JAVA	Java and Java-based XML support in Adaptive Server. No longer available; see XML Management package below.
Advanced security mechanisms	ASE_ASM	Network-based authentication and encryption using DCE and CyberSafe. No longer available; see Secruity and Directory Services package below.
Enterprise JavaBean Server	ASE_EJB	A transaction server provides the framework for creating, deploying, and managing middle-tier business logic in the form of EJBs in a multi-tier environment. No longer available; see XML Management package below.
Enhanced Full-Text Search	ASE_EFTS	Enhanced Full-Text search capacity for Adaptive Server.
External file support	ASE_XFS	Allows access to data external to the database in file systems. No longer available; see Content Management package below.
LDAP directory services	ASE_DIRS	Lightweight Directory services. No longer available; see Security and Directory Services package below.
Enterprise performance monitoring	ASE_XRAY	A graphical performance monitoring and diagnostic tool for Adaptive Server.

Feature	Feature name	Description
Web services	ASE_WEBSERVICES	Allows customers to access web services in Adaptive Server utilizing the Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) protocols.
XML support	ASE_XML	XML is a markup language and a subset of SGML (Standardized General Markup Language). XML is more complete and disciplined, and it allows you to define your own application-oriented markup tags. These properties make XML particularly suitable for data interchange.

In addition to individual option features there are several packages, which combine optional features. The only difference between options and packages is that a package requires that the package definition be included in the license file. The package definition should be included in the *license.dat* file automatically by the installation process The package definitions are also contained in a file named *sybpkg.dat*. This file is in the same directory as the *license.dat* file.

Table 11-2: Licensed Adaptive Server packages

Package	Feature name	Description
Security and Directory Services package	ASE_SECDIR	Enables the DCE or Kerberos library interfaces, SSL-based communication, row-level storage security, and LDAP services. Includes ASE_ASM and ASE_DIRS features.
XML Management package	ASE_XMLMGMT	Enables Java features, EJB support, and native XML support in Adaptive Server. Includes ASE_JAVA, ASE_EJB, and ASE_XML features.
Content Management package	ASE_CONTMGT	Allows use of file system data with Adaptive Server. Includes ASE_XFS feature.

Package	Feature name	Description
eBusiness package	ASE_EBIZ	A bundled package that contains Security and Directory Services package, XML Management package, and Content Management package.

Starting Adaptive Server with optional features

You can use sp_configure parameters to turn on or off optional features. Table 11-3 lists the parameters used to configure optional features.

Table 11-3: Configuration parameters for optional features

Feature name	License name	Configuration parameter
High availability	ASE_HA	enable HA
Distributed transaction management	ASE_DTM	enable DTM
Java in Adaptive Server	ASE_JAVA	enable java
Advanced security mechanisms	ASE_ASM	use security services enable ssl
Enterprise JavaBean 32-bit platforms only	ASE_EJB	enable enterprise java beans
LDAP directory services	ASE_DIRS	n/a
Enhanced full-text search	ASE_EFTS	enable full-text search
External file system	ASE_XFS	enable file access
Web Services	ASE_SEBSERVICES	enable web services
XML support	ASE_XML	enable xml services

Note Adaptive Server's support for distributed transaction management protocols: XA and Microsoft DTC protocols, requires configuration parameter enable DTM to be set and also a license to be available.

Distributed transaction management support through Adaptive Server Transaction Coordinator requires configuration parameter enable xact coordination to be set. However, this feature is available with no license requirements.

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 restart Adaptive Server, if directed to do so.

Installation and configuration information for licensable features

Some features require additional preparation before you can use them. For more information, see Table 11-3.

- 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, see *Using Adaptive Server Distributed Transaction Management Features*.
- For information about configuring Adaptive Server full-text search capabilities, see the *Full-Text Search Specialty Data Store User's Guide*.
- For information about configuring the EJB Server, see the *EJB Server User's Guide*.
- For information about Java in Adaptive Server, see *Java in Adaptive Server Enterprise*.
- For information about Web Services in Adaptive Server, see *Web Services User's Guide*.
- For information about XML, see the *XML User's Guide*.

Types of SySAM systems

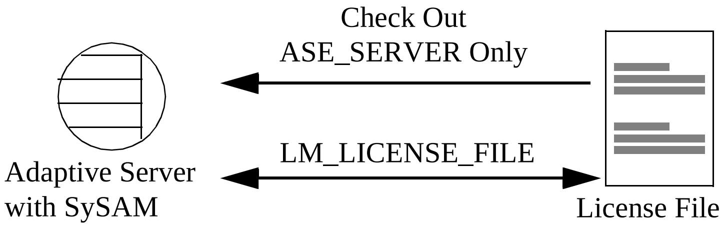
The basic components of SySAM can be arranged for a standalone server, for a collection of servers on a network accessing a primary license host, and with multiple, redundant servers across a LAN or a WAN, for backup and load-balancing purposes.

Depending on which instance of Adaptive Server (primary or secondary servers) you are installing, the procedures for registering licenses vary.

Standalone system

In a standalone system, *all* of the SySAM components shown in Figure 11-2 reside on a single machine. The license file points directly to the asset management software, which runs on the same machine. This machine is called the **license host**.

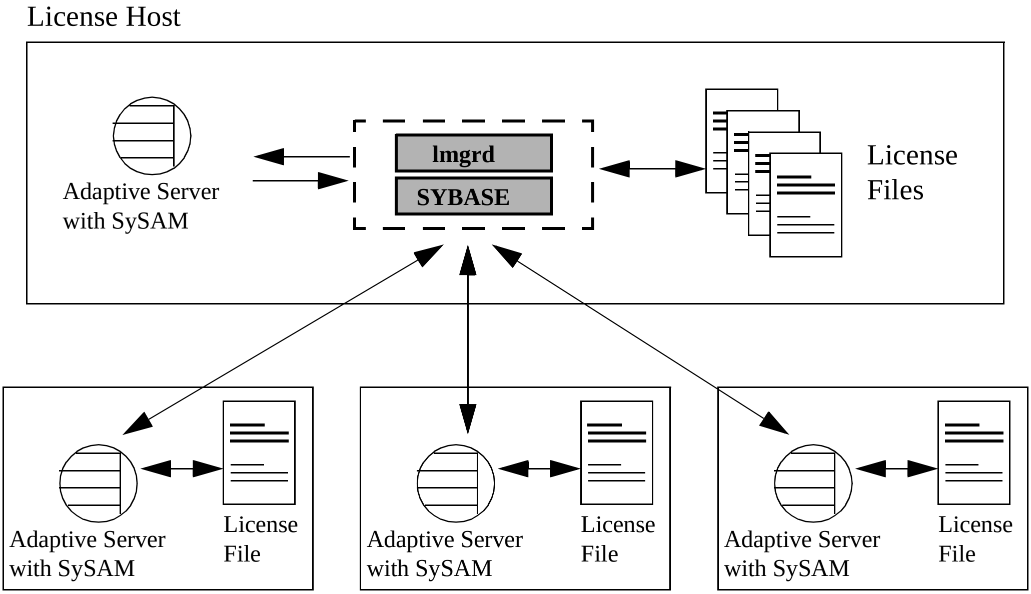
Figure 11-2: Standalone system without optional features



Network system

In a network system, the license host runs SySAM, which handles check-in and check-out requests from all **secondary servers**, as shown in Figure 11-3.

Figure 11-3: Network system



The license files for secondary servers contain only pointers to the license host. Secondary servers check out licenses from the license host through the SySAM software.

Note If you are using networked software asset management system, you must install the software on the license host before installing it on secondary servers.

Redundant servers

As part of a failover plan or a load-balancing system, you can have multiple servers running the same Adaptive Server configuration, using a redundant license across all servers. The **redundant server system** can be connected by a local-area network (LAN) or by a wide-area network (WAN).

A redundant server system is an excellent load-balancing mechanism for physically distant servers. The redundant license allows you to specify local servers as the first in the queue, and make remote servers available as backup. The SySAM application attempts to check out a license from a license-file list, starting with the first server. If that server fails for any reason, the second server in the list is contacted, and so on.

The redundant server system also provides failover protection. If one server in the list goes down for any reason, the second server responds to the license request. If that server fails too, the next server in the list is contacted, and so on.

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. In this system, the license file contains all license information for ASE_SERVER. Adaptive Server obtains the base 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.

SySAM in the network environment

Before you begin installing additional Adaptive Servers in a network system, the license manager must be installed and running on the license host. To verify that the license manager is running, see “SySAM administration” on page 133.

To install Adaptive Server on secondary servers, you need to have your Sybase License Certificate available. Also, you need to know:

- The host name of the license host.
- The port number on the license host where the asset manager listens for connections.

When installing Adaptive Server on secondary servers, run *$SYBASE/$SYBASE_SYSAM/bin/lmgr*.

1. lmgr prompts: “Do you have Sybase Software Asset Management Certificates to register?”

   Click No.

2. The installer prompts: “Have you registered Sybase Software Asset Management Certificates at a central license host?”

   Click Yes.

3. Enter the following information about the license host from which this machine will be checking out Adaptive Server licenses:

   - License Manager Host is the host name of the license host on which you installed the asset management software.
   - License Manager Port is an unused port number on the License host. The local Adaptive Server uses this port number to contact the primary network node when checking out licenses.

4. Click OK.

Redundant servers

As part of a failover plan or a load-balancing system, you may have multiple servers running the same Adaptive Server configuration, using redundant licenses across all servers. The redundant servers can be connected by LAN or by WAN.

To configure redundant servers:

1. Follow the instructions in this guide to install Adaptive Server on each of the servers in the redundant system.

2. As you install the software, make a note of the machine names and the dedicated port number that SySAM uses on each machine.

   Note The InstallShield Installer automatically returns the machine name and a valid port number during the installation process.

3. After the installation is complete on each machine, run *$SYBASE/$SYBASE_SYSAM/bin/lmgr*. See instructions in “Adding feature licenses”.

4. After the license entry is completed, and for each server in the redundant system, set the LM_LICENSE_FILE environment variable to:

   ```
   "port@machine:port@machine:port@machine"
   ```

For example, if you use port 29722 on each machine and the machine names are Huey, Dewey, and Louie, the environment variable that points to the license file would look like this:

```
LM_LICENSE_FILE="29722@huey:29722@dewey:
29722@louie"
```

The first server in the license list is the first server queued by the licensing software.

SySAM administration

If you are installing servers in a network environment, SySAM must be installed and running on the license host before installing Adaptive Server on additional servers. This section describes how to start SySAM manually or as an automatic service, and other SySAM administration information.

Verify the license daemons are running

To verify that the license daemons are 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. If you have not already done so, source the *SYBASE.csh* or *SYBASE.sh* file in the top level *SYBASE* directory. Use:

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

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

If the SySAM software is not running, see “Starting the software manually” on page 133.

Starting the software manually

If the license manager is not running, you must start the license manager manually.

Run the *startd.sh* script in the */bin* directory:

```
$SYBASE/$SYBASE_SYSAM/bin/startd.sh $SYBASE/$SYBASE_SYSAM
```

Starting SySAM as an automatic service

To configure SySAM to start automatically when the server is restarted, here is what you would do on Linux, for example:

1. Make sure the *license.dat* file has the correct path for the SYBASE daemon.
2. Using an account with “root” privileges, log on to the machine where you installed the asset management software.
3. Use a text editor to create a new file, */etc/rc3.d/S17lmgrd*.
4. Add the following lines to */etc/rc3.d/S17lmgrd*:

```
SYBASE=<Sybase reldir>
SYBASE_ASE=ASE-12_0
/bin/su user_name -c ‘echo starting lmgrd> \
$SYBASE_SYSAM/bin/boot.log’

/bin/nohup /bin/su user_name -c ‘umask 022; \
$SYBASE_SYSAM/bin/lmgrd -c \
$SYBASE_SYSAM/licenses/license.dat>> \
$SYBASE_SYSAM/bin/boot.log’\

/bin/su user_name -c ‘echo sleep 5>> \
$SYBASE_SYSAM/bin/boot.log’

/bin/sleep 5

/bin/su user_name -c ‘echo lmdiag >> \
$SYBASE_SYSAM/bin/boot.log’

/bin/su user_name -c
‘$SYBASE_SYSAM/bin/lmutil lmdiag -n -c \
$SYBASE_SYSAM/licenses/license.dat>> \
SYBASE_SYSAM/bin/boot.log’

/bin/su user_name -c ‘echo exiting >> \
$SYBASE_SYSAM/bin/boot.log’
```

5. Save */etc/rc3.d/S17lmgrd* and exit the text editor.

6 Change the permissions, ownership, and group for */etc/rc3.d/S17lmgrd* enter:

```
chmod 744 /etc/rc3.d/S17lmgrd
chown root /etc/rc3.d/S17lmgrd
chgrp sys /etc/rc3.d/S17lmgrd
```

Adding feature licenses

This section describes how to update the license file to enable Adaptive Server features.

To add features for which you have licenses:

1 Verify that the license manager software is running. See “SySAM administration” on page 133.

2 Log on to the machine where the license manager is installed (license host).

3 Launch the license manager:

```
$SYBASE/$SYBASE_SYSAM/bin/lmgr
```

4 Click Yes when prompted: “Do you have Sybase Software Asset Management Certificates to register?” The SySAM License Manager screen prompts you for:

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

5 Click More until you have entered all licenses.

6 Click Done.

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

If you encounter problems with new licenses, check the *lmgrd.log* file in the *$SYBASE/$SYBASE_SYSAM/log* directory to see that there were properly appended to the license file.

8 Enable the new feature. See “Starting Adaptive Server with optional features” on page 128.

9 Restart Adaptive Server.

Index