



Configuration Guide

Adaptive Server Enterprise

12.5

UNIX

DOCUMENT ID: 35823-01-1250-02

LAST REVISED: September 2002

Copyright © 1989-2002 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC-GATEWAY, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, MainframeConnect, Maintenance Express, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC Net Library, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 07/02

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

Contents

About This Book	ix
CHAPTER 1	Introduction
	1
	About Adaptive Server
	2
	System-specific issues
	2
	Definition of terms
	3
	User roles
	3
	Environment variables
	4
	Adaptive Server devices and system databases
	6
	The master device
	6
	The sybssystemdb device
	7
	The sysprocsdev device
	7
	Optional devices and databases
	7
	Client/server communication
	9
	Lightweight Directory Access Protocol
	10
	Changing Adaptive Server configuration
	11
	Support for high-availability products
	11
	Languages other than U.S. English
	12
CHAPTER 2	Starting and Stopping Servers
	13
	Overview
	13
	Requirements for starting servers
	13
	Starting servers
	14
	Server start-up parameters
	14
	Using a runserver file
	15
	Using the startserver command
	15
	Using the monserver command
	16
	Starting servers when the operating system restarts
	16
	For Compaq Tru64
	16
	For HP-UX
	18
	For IBM RS/6000
	19
	For SGI
	20
	For Sun Solaris
	20

For Macintosh OS X	22
Stopping servers	25
Stopping Adaptive Server.....	25
Stopping Backup Server.....	26
Stopping Monitor Server.....	27
Using the kill command	27
Shutdown and shared memory files.....	29

CHAPTER 3

Configuring the Operating System	31
Setting environment variables.....	31
DSLISTEN.....	32
DSQUERY.....	32
PATH.....	32
Using the stty setting.....	33
Restoring correct permissions.....	33
File descriptors and user connections.....	33
For Compaq Tru64	34
For HP-UX.....	34
For Sun Solaris and SGI	34
For Macintosh OS X.....	35
Displaying current soft and hard limits	35
Increasing the soft limit.....	35
Increasing the hard limit	36
Sample program.....	36
Adjusting the client connection timeout period.....	37
For Compaq Tru64	38
For HP-UX.....	38
For IBM RS/6000.....	38
For Sun Solaris.....	39
For Macintosh OS X.....	39
Checking for hardware errors.....	39
For Compaq Tru64	39
For HP-UX.....	39
For IBM RS/6000.....	40
For SGI.....	40
For Sun Solaris.....	40
For Macintosh OS X.....	40
Monitoring the use of operating system resources	40
For Compaq Tru64	40
For HP-UX.....	41
For IBM RS/6000.....	41
For Sun Solaris.....	42
For SGI.....	42
For Macintosh OS X.....	43

	A sample C shell maintenance script	43
CHAPTER 4	Adaptive Server Default Configuration	45
	Default settings	45
CHAPTER 5	Setting Up Communications Across the Network	47
	How clients connect to Adaptive Server	48
	Creating an interfaces file	49
	Contents of an interfaces file	49
	How Adaptive Server listens for client connections	50
	How a client uses an interfaces file	50
	Heterogeneous and homogeneous environments	51
	Understanding the format of the interfaces file	53
	Components of an interfaces file entry	54
	Format by platform	56
	Creating interfaces files for SPX	56
	Utilities for editing the interfaces file	58
	Creating a master interfaces file	58
	Using dsedit or dscp to create a master interfaces file	58
	Using a text editor to create a master interfaces file	59
	Configuring interfaces files for multiple networks	60
	Configuring the server for multiple network handlers	60
	Configuring the client connections	61
	Configuring for query port backup	64
	Troubleshooting	65
	Server fails to start	65
	Error when executing an ESP	66
CHAPTER 6	Lightweight Directory Access Protocol in Adaptive Server	67
	Overview	67
	LDAP directory services versus the Sybase interfaces file	69
	The libtcl.cfg and libtcl64.cfg files	72
	Enabling LDAP directory services	73
	Adding a server to the directory services	75
	Multiple directory services	77
	Encrypting the password	78
	Performance	79
	Migrating from the interfaces file to LDAP	79
CHAPTER 7	Customizing Localization for Adaptive Server	81
	Overview of localization support	81
	Language modules	82

Default character sets for servers	83
Supported character sets	84
Character set conversion	89
Conversions between server and client	89
Sort orders	90
Available sort orders.....	90
Language modules.....	93
Installing a new language module	93
Message languages	93
Localization	94
Localization directories.....	94
About the directory	95
About the charsets directory.....	95
About the locales.dat file	96
Changing the localization configuration	98
For Adaptive Server	99
For Backup Server	99
Sort orders.....	101
Character sets	102
charset utility	104

CHAPTER 8	Logging Error Messages and Events	107
	Adaptive Server error logging	107
	Enabling and disabling error logging	108
	Types of information logged	108
	Setting error log paths	108
	Setting the Adaptive Server error log path	109
	Managing messages	109
	Logging user-defined messages	109
	Logging auditing events	110

CHAPTER 9	Managing Adaptive Server Databases.....	111
	Managing database devices	111
	Device requirements	111
	Creating .dat files for database devices	112
	Backing up and restoring data	113
	Using a tape drive	113
	Dumping across a network.....	114
	Examples of backing up and restoring databases.....	115
	Using disk drives	120
	Monitoring disk usage	121

CHAPTER 10	Adding Optional Functionality to Adaptive Server	123
	Adding auditing.....	123
	Audit system devices and databases	124
	Overview of audit installation	124
	Pre-installation tasks for auditing devices.....	125
	Installing auditing	125
	Installing online help for Transact-SQL syntax	131
	Online syntax help: sp_syntax.....	131
	Default device for the sybsyntax database	132
	Installing sybsyntax.....	132
Index.....		135

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" presents 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, “Lightweight Directory Access Protocol in Adaptive Server” 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.

Related documents

The following documents comprise the Sybase Adaptive Server Enterprise documentation:

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

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

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

- *Configuraiton Guide* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5, the system changes added to support those features, and the changes that may affect your existing applications.
- *Transact-SQL User's Guide* – documents Transact-SQL®, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *Reference Manual* – contains detailed information about all Transact-SQL commands, functions, procedures, and datatypes. This manual also contains a list of the Transact-SQL reserved words and definitions of system tables.
- *Performance and Tuning Guide* – explains how to tune Adaptive Server for maximum performance. This manual includes information about database design issues that affect performance, query optimization, how to tune Adaptive Server for very large databases, disk and cache issues, and the effects of locking and cursors on performance.
- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book. Available only in print version.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.

-
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
 - *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.
 - *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
 - *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
 - *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
 - *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using Sybase's DTM XA Interface with X/Open XA transaction managers.
 - *Glossary* – defines technical terms used in the Adaptive Server documentation.
 - *Sybase jConnect for JDBC Programmer's Reference* – describes the jConnect™ for JDBC™ product and explains how to use it to access data stored in relational database management systems.
 - *Full-Text Search Specialty Data Store User's Guide* – describes how to use the full-text search feature with Verity to search Adaptive Server Enterprise data.
 - *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server and Adaptive Server.
 - *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
 - *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.

**Other sources of
information**

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

- 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.

❖ 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.

❖ 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 updates

❖ Finding the latest information on EBFs and software updates

- 1 Point your web browser to the Sybase Support Page at <http://www.sybase.com/support>.

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

Conventions

Table 1 shows the conventions used in this manual.

Table 1: Conventions for this manual

Item	Example
Command names, keywords, command options, utility names, utility flags, and system procedure names	create database isql -U sp_configure
Database names	master, model, and tempdb databases
Datatypes	int, datetime, and varchar datatypes
File names and path names	interfaces \$SYBASE/bin
Variables, or words that placeholders for values that you fill in	select column_name from table_name where search_conditions
Examples showing the use of Transact-SQL commands	select * from publishers
Examples of output from the computer	0736 New Age Books Boston MA 0877 Binnet & Hardley Washington DC 1389 Algodata Infosystems Berkeley CA
Parentheses – if shown in the syntax, type them as part of the command.	compute row_aggregate (column_name)
Comma – choose one or more options. Type commas as part of the command.	cash, check, credit
Vertical bar – choose only one option. Do not type the bars.	cash check credit
Curly braces around options separated by commas – you must choose at least one option. You may choose more than one. Do not type the braces.	{cash, check, credit}

Item	Example
Curly braces around options separated by vertical bars – you must choose only one option. Do not type the braces.	<code>{cash check credit}</code>
Brackets around options separated by commas – choosing an option is optional. You may choose more than one. Do not type the brackets.	<code>[cash, check, credit]</code>
Brackets around options separated by vertical bars – choosing an option is optional. You may choose only one. Do not type the brackets.	<code>[cash check credit]</code>
An ellipsis (...) means that you can repeat the preceding optional unit (between the square brackets) as many times as you like.	<code>buy toy = price {cash check credit} [, toy = price {cash check credit}]...</code>
In this example, you must specify at least one toy and enter its price. You must choose only one method of payment: cash, check, or credit. You may also buy additional toys. Separate each toy with a comma. You may put each item on a separate line for readability, but this is not required.	<code>buy doll = 25.50 check, train = 60.00 credit, bicycle = 150.00 credit, paint set = 6.00 cash</code>

Transact-SQL does not restrict the number of words you can put on a line, or where to break a line. However, for readability, examples and syntax statements in this manual are formatted so each clause of a statement begins on a new line. Clauses that have more than one part extend to additional lines, which are indented.

You can disregard case when typing Transact-SQL keywords. For example, Adaptive Server understands `SELECT`, `Select`, and `select` to be the same. In this manual, most examples are in lowercase. Case sensitivity of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. For more information, see Chapter 7, “Configuring Character Sets, Sort Orders, and Languages,” in the *System Administration Guide*.

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

Terms

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

Introduction

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

- Compaq Tru64 (formerly Digital UNIX)
- HP 9000 Series 800 HP-UX
- IBM RISC System/6000 AIX
- Sun Solaris 2.x (SPARC)
- Linux Kernel 2.2.14-5
- Macintosh OS X 10.2

Note The instructions in this book assume that Adaptive Server is installed and running. For information about installing and starting 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.

Topic	Page
About Adaptive Server	2
System-specific issues	2
Definition of terms	3
User roles	3
Environment variables	4
Adaptive Server devices and system databases	6
Client/server communication	9
Lightweight Directory Access Protocol	10
Changing Adaptive Server configuration	11
Support for high-availability products	11
Languages other than U.S. English	12

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.

Definition of terms

The following terms are used in this book:

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

Note PowerDynamo, PowerDesigner, and SQL Modeler are not available on Mac.

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.

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 the server name given at installation.
- **DSQUERY** – defines the Adaptive Server name that client programs try to connect to if no Adaptive Server is specified with a command-line option. If DSQUERY is not set, and you do not supply the Adaptive Server name with a command-line option, clients attempt to connect to the server name given at installation.
- **SYBASE** – defines the path of the Sybase installation directory. The installation program sets up the variable SYBASE to point to the release directory specified during installation.
- **SYBASE_ASE** – defines the subdirectory of the Adaptive Server component.
- **SYBASE_OCS** – defines the subdirectory to which the 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 which directory path to search for executables. The Sybase executables are located in the installed components */bin* directory. When using the source files *SYBASE.csh* or *SYBASE.sh*, the following paths are prepended to PATH:

`$SYBASE/$SYBASE_ASE/bin:$SYBASE/$SYBASE_OCS/bin`

- LD_LIBRARY_PATH – this variable specifies which 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 executables. The Sybase executables are located in the installed component's */bin* directory.

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

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

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 transaction in process.
- 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 Adaptive Server. 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

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

For instructions about creating the sybsystemdb device and database for data transfer management (two-phase commit), see Chapter 9, “Upgrading Sybase Servers” in the Adaptive Server installation documentation for your platform.

The sysprocsdev device

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

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

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

Optional devices and databases

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

The sample
databases

The sample databases are:

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

dbcc checkstorage records configuration information for the **target database**, operation activity, and the results of the operation in the dbccdb database. Stored in the database are dbcc stored procedures for creating and maintaining dbccdb and for generating reports on the results of dbcc checkstorage operations.

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

Client/server communication

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

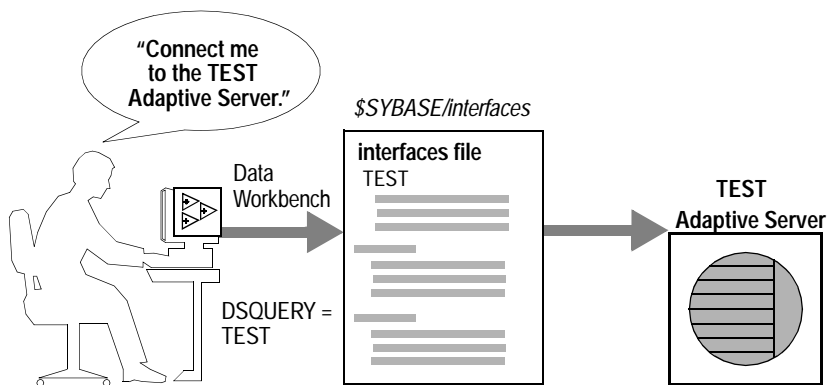
After your Adaptive Server or client software is installed, it can connect with any server on the network that is listed in the directory services.

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 to which they want to connect. It also tells Adaptive Server where to listen for incoming connection attempts from clients. Adaptive Server uses a single port for these two services (referred to as *query service* and *listener service*).

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

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



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

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

Lightweight Directory Access Protocol

In Adaptive Server 12.5 there is an alternative to the traditional Sybase interfaces file. For information about using Lightweight Directory Access Protocol (LDAP) directory services to establish connections, see Chapter 6, "Lightweight Directory Access Protocol in Adaptive Server."

Changing Adaptive Server configuration

On UNIX platforms, you use `sp_configure` to change the configuration of Adaptive Server. To change the configuration of several servers, enter the `sp_configure` information in a script. For details on using `sp_configure`, see the *System Administration Guide* and the *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 *Distributed Transaction Management User's Guide*.

Support for high-availability products

Adaptive Server is compatible with high-availability packages for several platforms and operating systems.

Note High-availability is not available on Mac OS X.

You can access the configuration procedures for these high-availability packages from the Sybase World Wide Web site.

❖ Viewing the high-availability setup procedures

- 1 Use a Web browser to access the Sybase World Wide Web site at the following address:
at <http://www.sybase.com/support/manuals>
- 2 In the Search dropdown menu, choose “All Technical Documents.”
- 3 Type “High Availability” in the Search utility, and click Search.

The Search utility generates a list of titles for Technical Notes that contain high-availability configuration procedures. See the titles that include “Configuring Sybase Adaptive Server Enterprise for High Availability.”

Languages other than U.S. English

Many of the configuration tasks described in this manual require the use of the Server Config utility.

Note The Server Config utility is not available on Mac OS X.

If you are running Server Config in a language other than U.S. English, make sure that any input you provide uses a character set that is supported by the `us_english` character set.

Note The `us_english` character set does not support accent marks, such as tildes (~) and umlauts (ü). This prevents Server Config 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.

Starting and Stopping Servers

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

Topic	Page
Overview	13
Starting servers	14
Starting servers when the operating system restarts	16
Stopping servers	25

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 only when an XP command is issued through isql.

You can use Sybase Central to start and stop servers manually or automatically.

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

Requirements for starting servers

To start a server your user account must have:

- Administrator privileges, or you can use the default “sa” login
- 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 UNIX command line by using *runserver* files and *startserver* commands. You can move the Adaptive Server shared memory files by adding a flag to the *runserver* file.

Server start-up parameters

The default start-up parameters are stored under *\$SYBASE/ASE_12_5/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 “_BS”, “_MS”, and “HS”, respectively.

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

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

Parameter	Switch	Description
Arg0	-d <i>\$SYBASE/data/master.dat</i>	Location of the master device file
Arg1	-s <i>server_name</i>	Name of the Adaptive Server
Arg2	-e <i>\$SYBASE_ASE/install/errorlog</i>	Location and name of the error log file
Arg4	-M <i>\$SYBASE/\$SYBASE_ASE/</i>	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/ASE_12_5/install/RUN_server_name* file. You can also specify additional start-up parameters within the *\$SYBASE/ASE_12_5/install/RUN_server_name* file.

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

Using a runserver file

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

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

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

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

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

Using the startserver command

To start a server from the command line, enter:

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

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

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

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

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

Using the `monserver` command

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

Starting servers when the operating system restarts

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

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

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

For Compaq Tru64

Before configuring your operating system for automatic restart:

- 1 Before starting the server, define the environment variable `LD_LIBRARY_PATH`.
- 2 Add `$SYBASE/product_name/lib` first to `LD_LIBRARY_PATH`. If `/usr/shlib` comes before `$SYBASE/product_name/lib` in `LD_LIBRARY_PATH`, you may encounter unresolvable symbol errors.

- 3 Verify that all network resources are available before you start Adaptive Server or Backup Server. The server cannot start if the network is down.
- 4 Verify that the entry in the *rc* (run control) directory for restarting the server *follows* any commands for starting network operations.
- 5 Designate a period of time for your server to wait before starting, by inserting a sleep command at the beginning of the linked *runserver* file you created. Slow start-up of network operations can prevent servers from starting, even when the commands are in the correct order in the *rc* directory.

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

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

```
#!/sbin/sh
# start a SYBASE Adaptive Server on system reboot
# set SYBASE to the path of the sybase directory
# set RUNSERVER to the name of the runserver file
#   for the Adaptive Server
# set BSERVER to the name of the runserver file
#   for the backup server
#
SYBASE=/sybase/rel11151
RUNSERVER=RUN_REL1151_XYZ
BSERVER=RUN_REL1151_XYZ_BACKUP
PATH=/sbin:/usr/sbin:/usr/bin
#
export PATH SYBASE RUNSERVER BSERVER
#
case "$1" in
'start')
    echo "Starting Sybase Adaptive Server"
    nohup su sybase -c \
        "$SYBASE/$SYBASE_ASE/install/startserver -f
        $SYBASE/$SYBASE_ASE/install/$RUNSERVER >/dev/null &"
    echo "Starting Sybase Backup Server"
    nohup su sybase -c \
        "$SYBASE/$SYBASE_ASE/install/startserver -f
        $SYBASE/$SYBASE_ASE/install/$BSERVER >/dev/null &"
    ;;
*)
    echo "Usage:  $0 {start}"
```

```
;;
esac
```

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

```
cp script_name /sbin/init.d/script_name
```
- 3 Review the “initdefault” entry in the `/etc/inittab` file to determine the default run level for your operating system. The default run level is usually 2 or 3.
- 4 Use the `ln` command to create a hard link from the runserver file to the appropriate run control (*rc*) directory, *rc#*, where # is the default run level obtained in step 3.

For example:

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

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

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

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

returns the following results:

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

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

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

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

For HP-UX

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

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

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

For IBM RS/6000

On production systems, 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 runserver file for the server you are starting.

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

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

```
sleep seconds_of_rest
```

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

For 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 will not start if the network is not up. 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

Before you set Adaptive Server or Backup Server to start automatically, make sure that all network resources are available. Your server will not start if the network is not up. Be sure that the entry for restarting the server *follows* any commands in the *rc* directory for starting network operations. Slow start-up of network operations 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 *runserver* file you created.

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

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

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

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

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

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

- 3 Review the contents of the `/etc/inittab` file to determine the default run level for your operating system. The `initdefault` entry specifies the default run level, which is usually 2 or 3.
- 4 Use the `ln` command to create a hard link from the `runserver` file to the appropriate *run control* (`rc`) directory, `rc#`, where `#` is the default run level obtained in step 2.

Use syntax like the following:

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

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

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

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

returns:

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

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

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

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

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 firmware is activated. 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-ROM. 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; for details, see “The rc.boot and rc scripts” on page 24. */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 24.
- 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*. This directory and its content should be copied 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*.

Stopping servers

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

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

Stopping Adaptive Server

To shut down Adaptive Server:

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

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

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

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

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

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

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

This is normal behavior.

If the message indicates that Adaptive Server is waiting for processes to complete, and you need to stop Adaptive Server immediately, you can use the shutdown with nowait command. shutdown with nowait does not wait for currently executing statements to finish, nor does it perform checkpoints in every database. Use the shutdown with nowait command 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:

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

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

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

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

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

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

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

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

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

Stopping Monitor Server

Note Monitor Server is not available on Mac OS X.

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. Monitor Server must be shut down to release resources. Otherwise, Adaptive Server may not be able to allocate enough resources to restart.

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

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

Using the kill command

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

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

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

```
kill pid
```

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

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

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

```
showserver
```

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

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

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

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

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

Shutdown and shared memory files

On Compaq Tru64, HP-UX, IBM RS/6000, and Mac OS X, when Adaptive Server starts, it creates the following files in *\$SYBASE* to store information about the shared memory segments that it uses:

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

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

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.

Topic	Page
Setting environment variables	31
Using the stty setting	33
Restoring correct permissions	33
File descriptors and user connections	33
Adjusting the client connection timeout period	37
Checking for hardware errors	39
Monitoring the use of operating system resources	40
A sample C shell maintenance script	43

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:

```
setenv
```

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

Table 3-1: System environment variables

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

DSLISEN

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

DSQUERY

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

PATH

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

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

Using the *stty* setting

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

```
stty -tostop
```

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

Restoring correct permissions

Sybase software files and directories are installed with the correct access permissions. If you notice that the permissions are no longer correct, you can restore the correct permissions with the script *setperm_all*, located in the *\$SYBASE/\$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 Compaq 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 Compaq Tru64 operating system documentation.

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

```
ulimit -n
```

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

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.

For Sun Solaris and SGI

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

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

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

For Macintosh OS X

Adaptive Server automatically sets the maximum file descriptors per process to the operating system maximum of 10240. 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 1024 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 *runserver* file to increase the hard and soft limits. Because the *runserver* file is a Bourne shell script, be sure to use the Bourne shell versions of these commands in the *runserver* file.

Increasing the hard limit

To increase the hard limit, use a program like the sample program shown in “Sample program” on page 36.

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

- 1 Create the ASCII file *file_name.c* (where *file_name* is the name you give the file), by using an ASCII text editor. Type the text shown in the sample in “Sample program” on page 36.

- 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 *runserver* file.

Sample program

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

```
#include <sys/time.h>  
#include <sys/resource.h>  
#include <sys/types.h>  
/*  
** define MAX_CONNECTIONS to a number less than
```

```
** 10000. The number defined will then become the maximum
** number of connections allowed by an Adaptive Server.
*/
#define MAX_CONNECTIONS 9999
extern int errno;

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

    rlp.rlim_cur = MAX_CONNECTIONS;
    rlp.rlim_max = MAX_CONNECTIONS;
    /* set the number of open file descriptors 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*.

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 Compaq 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 lower case “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_keeppidle
```

The tcp_keeppidle 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

There is currently no way to adjust the timeout value on Mac OS X.

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 Compaq 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, `sunddiag`, 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`.

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, you need to monitor system and CPU usage.

For Compaq 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 mbufs 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

Sun Solaris supplies the following tools to help monitor performance:

- The `iostat` command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The `vmstat` command monitors virtual memory usage.
- The `netstat` command monitors network status.
- The `ps` command gives you an accurate snapshot of accumulated CPU time and usage for individual processes. This can be very helpful in determining the 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
```

```
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.

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	<i>AdaptiveServername</i>
Network support	Named Pipes, Windows Sockets (TCP/IP)
Pipe name	<i>/pipe/sybase/query</i>
Socket number	5000 For Mac OS X –11222
Command-line options	None
Error log path	<i>\$SYBASE/\$SYBASE_ASE/install/servername.log</i>
Event logging	Not configured

Item	Default value
International support (Localization)	
- Language	us_english
- Character set	DUNIX – ISO 8859-1 HP – Roman8 IBM – ISO 8859-1 Sun – ISO 8859-1 Mac OS X – 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 “About Adaptive Server” on page 2.

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

Server	Item	Default value
Backup Server	Name	<i>AdaptiveServername_bak</i>
	Network support	Named Pipes, Windows Sockets (TCP/IP)
	Pipe name	<i>/pipe/sybase/backup</i>
	Socket number	5001 For Mac OS X – 11223
	Error log path	
Monitor Server	Name	<i>AdaptiveServername_MS</i>
	Network support	Named Pipes, Windows Sockets (TCP/IP)
	Pipe name	<i>\pipe\sybase\monitor</i>
	Socket number	5002
	Error log path	
XP Server	Name	<i>\$SYBASE/\$SYBASE_ASE/_XP</i>
	Network support	Named Pipes, Windows Sockets (TCP/IP)
	Pipe name	<i>/pipe/sybase/xp</i>
	Socket number	5003 For Mac OS X – 11224
	Error log path	N/A

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.

Adaptive Server uses interface files and directory services to establish client and RPC connections. On UNIX platforms, the interfaces file is named *interfaces*. This chapter describes the interfaces file and provides specific configuration information for each supported UNIX platform.

Topic	Page
How clients connect to Adaptive Server	48
How Adaptive Server listens for client connections	50
How a client uses an interfaces file	50
Heterogeneous and homogeneous environments	51
Understanding the format of the interfaces file	53
Utilities for editing the interfaces file	58
Creating a master interfaces file	58
Configuring interfaces files for multiple networks	60
Troubleshooting	65

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

Table 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 58

Type of interfaces file	Task or topic	See
	Configuring for multiple networks	“Configuring interfaces files for multiple networks” in this document
	Reference information	“Understand the format of the interfaces file” in this document
PC-client	Configuring a client	<i>Installing Adaptive Server and OmniConnect on UNIX Platforms</i>
	Reference information and instructions for advanced tasks	<i>Open Client Supplement</i> for your PC client platform, or the appropriate Open Client documentation
Client platforms not listed	Configuring, reference information, and instructions for advanced tasks	<i>Open Client Supplement</i> for your PC client platform, or the appropriate Open Client documentation

How clients connect to Adaptive Server

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

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

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

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

Creating an interfaces file

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

Contents of an interfaces file

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

The network information in the file includes the server name, network name or address of the host machine, and the port, object, or socket number (depending on the network protocol) on which the server listens for queries. See “Understanding the format of the interfaces file” on page 53 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 58.

If the host machine supports multiple networks, see “Configuring interfaces files for multiple networks” on page 60.

How Adaptive Server listens for client connections

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

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

How a client uses an interfaces file

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

- Determines the name of the server either programmatically or by referring to the DSQUERY environment variable. If the application user has not set DSQUERY, the runtime value for the server name defaults to the SYBASE environment variable.
- Looks in the interfaces file for an entry whose name matches the name of the server, stops at the first instance of that name, and reads the query line.
- Uses the network information provided by the query line to connect to the server. If the client cannot connect the first time, it makes additional attempts according to the delay and retry numbers indicated in the interfaces file. If no matching entry is found, an error message is written to the client's standard error file. If multiple networks are supported, the client attempts to connect using the information in the second query entry for the server. See "Configuring for query port backup" on page 64.

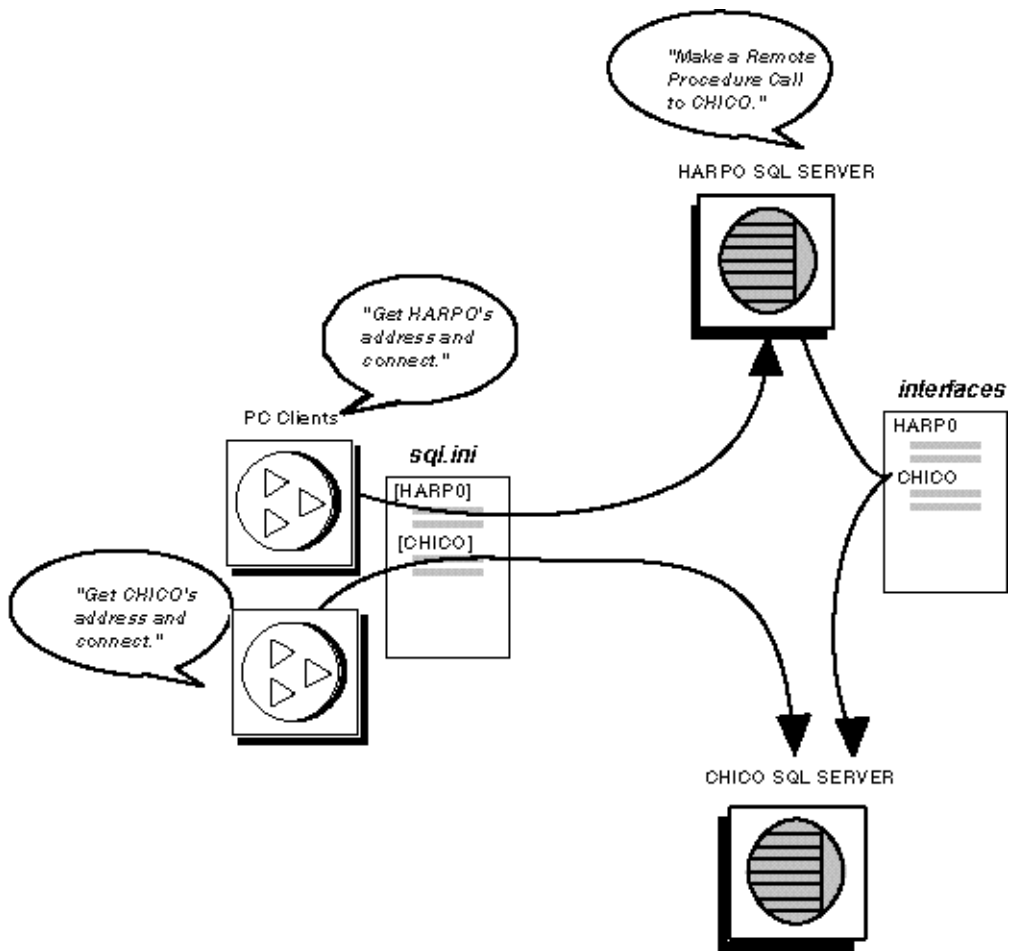
The Open Client documentation discusses client connections in much greater detail. See the *Open/Client Programmer's Supplement* for your client platform or the appropriate Open/Client documentation.

Heterogeneous and homogeneous environments

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

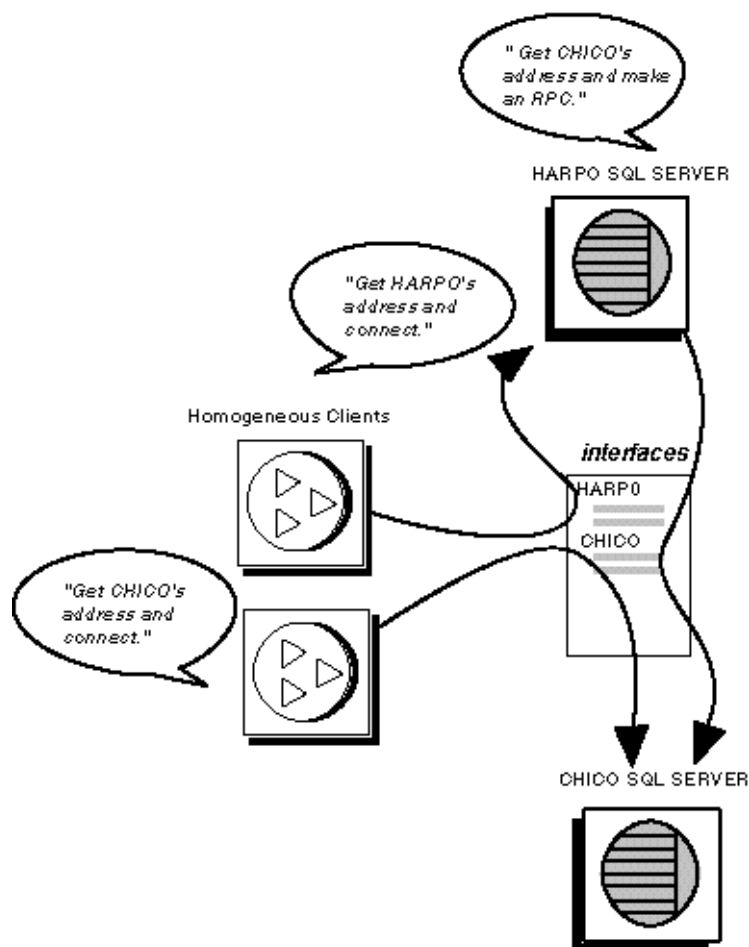
If the platforms are different, each platform may require a different format and configuration for its *interfaces* file. Figure 5-1 illustrates how a client PC uses network information in its *interfaces* file (*sql.ini*) to connect to Adaptive Server running under UNIX, and how Adaptive Server uses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

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



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



Understanding the format of the interfaces file

When you install an Adaptive Server, `srvbuild` creates an entry for it in the `interfaces` file.

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
<i>servername</i>	Name of Adaptive Server or Backup Server. Requirements for a server name are: <ul style="list-style-type: none"> • 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 (_).
<i>retry_attempts</i> (optional)	Number of times you want the client to try to connect to a server after initial failure. Default is 0.
<i>delay_interval</i> (optional)	Number of seconds between connection attempts. Default is 0.
<i>service_type</i>	Type of service defined by entry. Must be one of the following: <ul style="list-style-type: none"> • master • query • debug (connection for use by Sybase Technical Support only)
<i>api</i>	Application programming interface available to the network.
<i>protocol</i>	Name of the network protocol. Protocol types available are: <ul style="list-style-type: none"> • TCP/IP, represented by the letters “tcp” • SPX, represented by the letters “spx”
<i>network</i>	Name of the network; not currently used by Adaptive Server. <i>srvbuild</i> enters “ether” as a placeholder.
<i>host</i>	Network name or address of server’s host machine. <ul style="list-style-type: none"> • 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 • For DECnet, use the node name. Maximum size of entry is 6 bytes.
<i>machine</i>	Network name or address of server’s host machine. <ul style="list-style-type: none"> • 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

Component	Value
<i>device</i>	<p>The network device endpoint.</p> <ul style="list-style-type: none"> For TCP networks, it varies according to the vendor supplying the networking software. Check the vendor-specific documentation for the name of the device. Your network may provide multiple streams devices corresponding to different protocols in the TCP protocol suite. Choose the TCP streams device. Common TCP streams devices include <i>/dev/tcp</i>, <i>/dev/ito</i>, and <i>/dev/inet/tcp</i>. For the IPX protocol, the value is always <i>/dev/nspx</i>.
<i>address</i> for a TCP protocol entry	<p>Address consists of:</p> <ul style="list-style-type: none"> Address prefix, “\x” for tli. Network type, always 0002. Port number converted to four digits, hexadecimal. Must be between 1025 and 65535. Must be unique. Check the <i>/etc/services</i> 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 <i>/etc/services</i> labeled “Sybase specific services.” You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using the port number. IP network node address of the host machine converted to eight digits, hexadecimal. Trailing zeros, optional, 16 digits.
<i>address</i> for an SPX protocol entry	<p>Address consists of:</p> <ul style="list-style-type: none"> Address prefix, “\x” for tli Network address, 8 digits Node address, 12 digits Socket number, 4 digits
<i>port/object number</i>	<ul style="list-style-type: none"> For TCP/IP networks, port is a unique port number between 1025 and 65535. Check the <i>/etc/services</i> 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 <i>/etc/services</i> labeled “Sybase specific services.” You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using that number. For DECnet networks, <i>object</i> is an object number between 128 and 253. The number must be unique to the node on which the server is running.
<i>port</i>	<p>A unique port number between 1025 and 65535. Check the <i>/etc/services</i> 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 <i>/etc/services</i> 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.</p>

Format by platform

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

- Each Adaptive Server has only one entry, although there may be multiple lines in the entry.
- Each line that follows the *servername* line must begin with a space or a character tab.
- Each element on the line must be separated by a single space.
- Each entry is separated by a blank line.
- You can add comments to an interfaces file by adding a pound sign (#) at the beginning of the line and a line break at the end.

Compaq Tru64

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

HP-UX

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

IBM RS/6000

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

Sun Solaris

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

Macintosh OS X

```
server_name <retry> <delay> <newline>  
<tab>query tcp ether <hostname> <port_number><newline>  
<tab>master tcp ether <hostname> <port_number><newline>
```

Creating interfaces files for SPX

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

```
servername retry_attempts delay_interval<newline>
```



```
<tab>service_type api protocol device address<newline>
```

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

The *servername*, *retry_attempts*, *delay_interval*, and *service_type* components have the same meaning in an SPX entry as in a TCP entry. Table 5-3 describes the other components.

Table 5-3: remaining components in the interfaces file

Component	Value
<i>api</i>	<i>api</i> is the application programming interface to the network. Since Adaptive Server uses the Transport Layer Interface (TLI) for SPX protocol connections, use the letters “tli”.
<i>protocol</i>	<i>protocol</i> is the name of the network protocol. Use the letters “spx” to designate the SPX protocol.
<i>device</i>	<i>device</i> is the network device endpoint supplied for SPX networks. The device that is supplied with the Novell Protocol Suite network software is named <i>/dev/nspx</i> .
<i>address</i>	<p><i>address</i> is a composite entry generated from the network number, node, and socket number, defined as follows:</p> <ul style="list-style-type: none"> • The first four bytes of the address represent the network number converted to hexadecimal. • The next six bytes represent the network node of the machine running Adaptive Server. • Both the network number and node are assigned by the network administrator of your system. You can check the <i>/usr/netware/npsconfig</i> file to determine the network and node numbers for your machine. • The socket number is a hexadecimal number between 0x8000 and 0xFFFF that identifies the process on the machine. <p>If you choose a socket number that is already in use on the machine where your Adaptive Server is located, you receive a message similar to the following when you attempt to restart your Adaptive Server:</p> <pre>ninit: t_bind, Incorrect permissions ninitconn_free: t_close, fd=5, Illegal transport fd ninit: All master network listeners have failed. Shutting down. ueshutdown: exiting</pre> <p>More than one Adaptive Server can have the same socket number, if each server is on a different machine. The <i>combination</i> of network number, node, and socket uniquely identifies the Adaptive Server.</p> <p>For example, if your machine has a network number of 4D160D14 and a node number of 000000000001, and you use socket number 86bb, the following address entry is added:</p> <pre>/x4D160D140000000000000186bb</pre>

Utilities for editing the *interfaces* file

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

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

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

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, a utility called Server Discovery can be used to add entries to the *interfaces* file. For information, see the Installation Notes for Adaptive Server version 12.5.0.2 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, each connected to a different network. The server host machine's network address is 131.214.1.246 on Network 1 and 131.214.1.247 on Network 2.

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

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

```
# PRODUCTION server with two network listeners
```

```
PRODUCTION<tab>3<tab>3<newline>
<tab>master tcp ether SERV_CORPNET 6464
<tab>master decnet ether SERV_ENGNET 150
<tab>query tcp ether zeppo 6464
```

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

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

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

Configuring the client connections

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

- Use 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 would look like the following example on the “engineering” network:

```
#PRODUCTION server on network2
PRODUCTION<tab>3<tab>3<newline>
<tab>query tli tcp /dev/tcp \x0002130a83d601f70000000000000000
<tab>master tli tcp /dev/tcp \x0002130a83d601f60000000000000000
<tab>master tli tcp /dev/tcp \x0002130a83d601f70000000000000000#PRODUCTION
server network2<newline>
PRODUCTION<tab>3<tab>3<newline>
<tab>query decnet ether CHICO 150
<tab>master tcp ether ZEPPO 6464
<tab>master decnet ether CHICO 150# PRODUCTION server network2<newline>
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_ENGNET 5470
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

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

UNIXWare

Using this method with UNIXWare, if the interfaces file examples shown in “Sample interfaces files for multiple network handlers” on page 60 were adapted for this single DSQUERY method, they would look like the following example. The following example shows an interfaces file on the “corporate” network in a homogeneous UNIXWare environment:

```
#PRODUCTION server network1
PRODUCTION<tab>3<tab>3<newline>
<tab>query tli tcp /dev/tcp \x0002130a83d601f60000000000000000
<tab>master tli tcp /dev/tcp \x0002130a83d601f60000000000000000
<tab>master tli tcp /dev/tcp \x0002130a83d601f70000000000000000#PRODUCTION
server network1
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether ZEPPO 6464
<tab>master tcp ether ZEPPO 6464
<tab>master decnet ether CHICO 150# PRODUCTION server network1
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 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_network1 and one for PRODUCTION_network2. For more information, see the Open Client documentation for your client platform.
- For UNIX clients, you can edit the *interfaces* files with an ASCII text editor. From the server’s *interfaces* files, copy the server name line and the “master” line for each network into the client interfaces file. Add the appropriate server name for each entry, and change “master” to “query.”

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

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

```
PRODUCTION_network1<tab>3<tab>3<newline>
<tab>query tcp ether serv_corpnet 4559
# Client entry for PRODUCTION on network2
PRODUCTION_network2<tab>3<tab>3<newline>
<tab>query tcp ether serv_engnet 5479
```

Configuring for query port backup

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

To configure Adaptive Server for query port backup:

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

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

```
# PRODUCTION server with two network listeners
PRODUCTION<tab>3<tab>3<newline>
<tab>master tli tcp /dev/tcp \x0002130a83d601f60000000000000000
<tab>master tli tcp /dev/tcp \x0002130a83d601f70000000000000000
<tab>query tli tcp /dev/tcp \x0002130a83d601f60000000000000000
<tab>query tli tcp /dev/tcp
\x0002130a83d601f70000000000000000#PRODUCTION server with two network
listeners
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether ZEPP0 6464
<tab>query decnet ether CHICO 150
<tab>master tcp ether ZEPP0 6464
<tab>master decnet ether CHICO 150# PRODUCTION server with two network
listeners
PRODUCTION<tab>3<tab>3<newline>
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
<tab>query tcp ether SERV_CORPNET 4559
<tab>query tcp ether SERV_ENGNET 5479
```


- 3 Configure PC-client interfaces files with the appropriate multiple “query” entries, as described in the Open Client documentation. For client interfaces files in a homogeneous environment, you can copy the entire interfaces file entry for the Adaptive Server into the client interfaces file.
- 4 A connection on the secondary port occurs only if the corporate network is disabled, or if the corporate network interface on the host machine fails or is shut down due to a network-related failure.

Troubleshooting

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

Server fails to start

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

❖ To investigate the port assignment

- 1 Look in the interfaces file to identify the port number assigned to the server.
- 2 Determine whether another process is using the same port number by entering:

```
netstat -a
```

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

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

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

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

❖ 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 *Utility Programs*.
 - 3 If the server that failed to start was an XP Server, kill the server that you started manually, then start Adaptive Server.
XP Server should start automatically.

Error when executing an ESP

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

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

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

If you find no processes using the same port number:

- 1 Restart Adaptive Server.
- 2 Execute the ESP that you attempted earlier.
XP Server should start automatically.

Lightweight Directory Access Protocol in Adaptive Server

Adaptive Server uses directory services to establish client and RPC connections over the Internet. This chapter provides information about using LDAP directory services to establish connections.

Note LDAP is not supported for Macintosh OS X.

Topic	Page
Overview	67
LDAP directory services versus the Sybase interfaces file	69
The libtcl.cfg and libtcl64.cfg files	72
Enabling LDAP directory services	73
Adding a server to the directory services	75
Multiple directory services	77
Encrypting the password	78
Performance	79
Migrating from the interfaces file to LDAP	79

Overview

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

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

The LDAP server stores and retrieves information about:

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

The LDAP server can be configured with these access restrictions:

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

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

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

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

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

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

If multiple directory services are supported in a server, then the order in which they are searched is specified in *libtcl*.cfg*. You cannot specify the search order with the `dataserver` command-line option. See “Multiple directory services” on page 77.

LDAP directory services versus the Sybase interfaces file

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

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

Table 6-1 highlights the differences between the Sybase interfaces file and an LDAP server.

Table 6-1: interfaces file versus LDAP directory services

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

LDAP directory services support more attributes than the Sybase interfaces file. These attributes can include server version, server status, and so on. See Table 6-2 for a list of attributes.

Note LDAP is only supported with reentrant libraries. You must use `isql_r`, instead of `isql`, when connecting to a server using LDAP directory services.

Table 6-2 lists the Sybase LDAP directory entries.

Table 6-2: Sybase LDAP directory definitions

Attribute name	Value type	Description
ditbase	interfaces file or <i>libtcl*.cfg</i>	DIT base for object tree. If the <i>libtcl*.cfg</i> file is specified, the interfaces file is ignored. The <i>libtcl*.cfg</i> file can be overridden with <code>ct_con_prop()</code> for a specified connection.
dn	Character string	Distinguished name. Must be unique name that identifies the object.
sybaseVersion	Integer	Server version number.
sybaseServername	Character string	Server name.

Attribute name	Value type	Description
sybaseService	Character string	Service type: Sybase Adaptive Server, or Sybase SQL Server.
sybaseStatus	Integer	Status: 1 – Active, 2 – Stopped, 3 – Failed, 4 – Unknown.
sybaseAddress	String	Each server address includes: <ul style="list-style-type: none"> • Protocol: TCP, NAMEPIPE, SPX, DECNET (entry is case-sensitive). • Address: any valid address for the protocol type. <hr/> <p>Note dscp splits this attribute into Transport type and Transport address.</p> <hr/>
sybaseSecurity (optional)	String	Security OID (object ID).
sybaseRetryCount	Integer	This attribute is mapped to CS_RETRY_COUNT, which specifies the number of times that ct_connect retries the sequence of network addresses associated with a server name.
sybaseRetryDelay	Integer	This attribute is mapped to CS_LOOP_DELAY, which specifies the delay, in seconds, that ct_connect waits before retrying the entire sequence of addresses.
sybaseHAServername (optional)	String	A secondary server for failover protection.

The traditional interfaces file on a UNIX machine with TCP connection and a failover machine looks like:

```
master tcp ether huey 5000
query tcp ether huey 5000
hafailover secondary
```

An example of an LDAP entry with TCP and a failover machine looks like:

```
dn: sybaseServername=foobar, dc=sybase, dc=com
objectClass: sybaseServer
sybaseVersion: 12500
sybaseServername: foobar
sybaseService: ASE
sybaseStatus: 4
sybaseAddress: TCP#1#foobar 5000
```

```
sybaseRetryCount: 12
sybaseRetryDelay: 30
sybaseHAServernam: secondary
```

All entries in the LDAP directory service are called entities. Each entity has a distinguished name (DN) and is stored in a hierarchical tree structure based on its DN. This tree is called the **directory information tree** (DIT). Client applications use a DIT base to specify where entities are stored. See “The libtcl.cfg and libtcl64.cfg files” on page 72.

In the example above, the entry describes an Adaptive Server named “foobar” listening on a TCP connection with a port number of 5000. This entity also specifies a retry count of 12 (times) and a retry delay of 30 (seconds). Once a client has found an address where a server responds, the login dialog between the client and the server begins.

You can find a complete list of Sybase’s LDAP directory schema in `$SYBASE/$SYBASE_OCS/config`

In the same directory, there is also a file called *sybase-schema.conf*, which contains the same schema, but uses a Netscape-specific syntax.

Since LDAP supports multiple entries for each attribute, each address attribute must contain the address of a single server, including protocol, access type, and address. See `sybaseAddress` in Table 6-2.

For example, this is an LDAP entry for an NT server listening on two addresses, with different connection protocols:

```
sybaseAddress = TCP#1#TOEJAM 4444
sybaseAddress = NAMEPIPE#1#\pipe\sybase\query
```

Note Each entry in the address field is separated by the # character.

You can edit these entries with the `dscp` or `dsedit` utility. See “Adding a server to the directory services” on page 75.

To ensure cross-platform compatibility for all Sybase products, the protocol and address attribute fields should be in a platform- and product-independent format.

The *libtcl.cfg* and *libtcl64.cfg* files

You use the *libtcl.cfg* and the *libtcl64.cfg* files (collectively *libtcl*.cfg* files) to specify the LDAP server name, port number, DIT base, user name, and password to authenticate the connection to an LDAP server.

The purpose of the *libtcl*.cfg* files is to provide configuration information such as driver, directory, and security services for Open Client/Open Server and Open Client/Open Server-based applications. Both *libtcl.cfg* and *libtcl64.cfg* are provided on 64-bit platforms. 32-bit utilities (on 64-bit platforms) such as *dsedit* and *srvbuild*, look up the *libtcl.cfg* file while 64-bit applications look up the *libtcl64.cfg* file for configuration information.

You should edit both the *libtcl.cfg* and the *libtcl64.cfg* files to ensure compatibility between 32- and 64-bit applications.

The default *libtcl*.cfg* file is located in
`$SYBASE/$SYBASE_OCS/config`

If LDAP is specified in the *libtcl*.cfg* file, the interfaces file is not used.

Note Open Client/Open Server applications that use the `-l` option at start-up override the *libtcl*.cfg* file and use the interfaces file.

In its simplest form, the *libtcl*.cfg* file is in this format:

```
[DIRECTORY]
ldap=libdldap.so ldapurl
```

where the *ldapurl* is defined as:

```
ldap://host:port/ditbase
```

The following LDAP entry, using these same attributes, is an anonymous connection and only works only if the LDAP server allows read-only access.

```
ldap=libdldap.so ldap://test:389/dc=sybase,dc=com
```

You can specify a user name and password in the *libtcl*.cfg* file as extensions to the LDAP URL to enable password authentication at connection time.

Enabling LDAP directory services

To use a directory service, you must:

- 1 Configure the LDAP server according to the vendor-supplied documentation.
- 2 Add the location of the LDAP libraries to the path environment variable for your platform. See Table 6-4 on page 74.
- 3 Configure the *libtcl*.cfg* file to use directory services.

Use any standard ASCII text editor to:

- Remove the semicolon (;) comment markers from the beginning of the LDAP URL lines in the *libtcl*.cfg* file under the *[DIRECTORY]* entry.
- Add the LDAP URL under the *[DIRECTORY]* entry. See Table 6-3 for supported LDAP URL values.

Warning! The LDAP URL must be on a single line.

```
ldap=libldap.so ldap://host:port/ditbase??scope??
bindname=username password
```

For example:

```
[DIRECTORY]
ldap=libldap.so ldap://huey:11389/dc=sybase,dc=com??
one??bindname=cn=Manager,dc=sybase,dc=com secret
```

“*one*” indicates the scope of a search that retrieves entries one level below the DIT base.

Table 6-3 defines the keywords for the *ldapurl* variables.

Table 6-3: Idapurl variables

Keyword	Description	Default
<i>host</i> (required)	The host name or IP address of the machine running the LDAP server	None
<i>port</i>	The port number that the LDAP server is listening on	389
<i>ditbase</i> (required)	The default DIT base	None
<i>username</i>	Distinguished name (DN) of the user to authenticate	NULL (anonymous authentication)
<i>password</i>	Password of the user to be authenticated	NULL (anonymous authentication)

- 4 Verify that the appropriate environment variable points to the required third-party libraries. Table 6-4 lists the location of the Netscape LDAP SDK libraries.

Table 6-4: Environment variables

Platform	Environment variable	Library location
Compaq Tru64	LD_LIBRARY_PATH	\$SYBASE/\$SYBASE_OCS/lib3p
HP-UX 32-bit	SHLIB_PATH	\$SYBASE/\$SYBASE-OCS/lib3p
HP-UX 64-bit	LD_LIBRARY_PATH	\$SYBASE/\$SYBASE_OCS/lib3p64
Linux	LD_LIBRARY_PATH	\$SYBASE/\$SYBASE_OCS/lib3p
IBM RS6000 32-bit	LIBPATH	\$SYBASE/\$SYBASE_OCS/lib3p
Sun Solaris 32-bit	LD_LIBRARY_PATH	\$SYBASE/\$SYBASE_OCS/lib3p
Sun Solaris 64-bit	LD_LIBRARY_PATH_64	\$SYBASE/\$SYBASE_OCS/lib3p64

- 5 Add your server entry to the LDAP server using dscp or dsedit. See “Adding a server to the directory services” on page 75.

Adding a server to the directory services

Warning! Most LDAP servers have an `ldapadd` utility for adding directory entries. Sybase recommends that you use `dsedit` or `dscp` instead since they have built-in semantic checks that generic tools do not provide.

Each server entry is made up of a set of attributes. When you add or modify a server entry, you are prompted for information about server attributes. Some attributes are provided by default, others require user input. When a default value is provided, it appears in brackets “[]”. See Table 6-2 for accepted values.

You can use `srvbuild` to add entries, but not modify or delete them.

❖ Adding a server entry to the directory service using `dsedit`

Use `dsedit` to add a server to the directory service:

- 1 From the `$SYBASE/$SYBASE_OCS/bin` directory, enter:

```
dsedit
```

- 2 Select LDAP from the list of servers, and click OK.
- 3 Click Add New Server Entry.
- 4 Enter:
 - The server name – this is required.
 - Security mechanism – optional. A list of security mechanism OIDs are located in `$SYBASE/$SYBASE_OCS/config/objectid.dat`.
 - HA server name – optional. This is the name of the high-availability failover server, if you have one.
- 5 Click Add New Network Transport.
 - Select the transport type from the drop-down list.
 - Enter the host name.
 - Enter the port number.
- 6 Click OK two times to exit the `dsedit` utility.

❖ **Using *dscp* to add a server entry to the directory service**

dscp is a command-line utility that adds server entries to LDAP servers. After opening a session, you can check your configuration, view and modify existing entries, and create new entries.

Use *dscp* to add a server to the directory services.

- 1 To start *dscp*, enter:

```
$SYBASE/$SYBASE_OCS/bin/dscp
```

- 2 Before you can view, add, or modify server entries, you must open a session.

Opening a *dscp* session allows you to interact with any directory service that has a driver listed in *libtcl*.cfg*. To open a session, enter:

```
open DSNAME
```

where *DSNAME* is the name of the directory service.

If you do not specify *DSNAME*, *dscp* uses the default directory service provider specified in the *libtcl*.cfg* file. If there are no entries in the *libtcl*.cfg* file, *dscp* uses the default interfaces file located in *\$SYBASE*.

- 3 A connection to an LDAP server is indicated by this prompt:

```
Session 1 ldap>>
```

If the LDAP server requires user authentication, you must use the *-Username* command-line parameter flag when starting *dscp* and *dsedit*.

- 4 To add a server to the directory services, enter:

```
add server_name
```

where *server_name* is the name of the server to be added.

- 5 The next prompt specifies the service type. Adaptive Server is the default value:

```
Service [ASE Server]
```

Click Enter to accept the default.

- 6 Enter the transport type. Press Enter to accept the default value TCP, or enter a value from Table 6-2.

- 7 Enter the transport address. Valid entries are any values that enable the transport type specified. For example, for a TCP connection, *host_name port_number*.
- 8 Since any LDAP server entity can have multiple address entries, the next prompt returns to the prompt for "Transport type." You can enter another transport type, or click Enter (leaving the field blank) to skip this prompt and proceed.
- 9 You are then prompted for another transport address. You can enter another valid address that corresponds to the additional transport type, or press Enter (leaving the field blank) to proceed.
- 10 Optionally, enter a security mechanism OID.
- 11 Optionally, enter a secondary server for failover.
- 12 Press Enter. When completed, the message: "Added *server_name* done" displays.

To view the server entries, enter the following URL in Netscape:

```
ldap://host:port/ditbase??one
```

For example:

```
ldap://huey:11389/dc=sybase,dc=com??one
```

Note Microsoft Internet Explorer does not recognize LDAP URLs.

For more information about dscp, see the *Open Client/Server Configuration Guide*.

Multiple directory services

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

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

For example:

[DIRECTORY]

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

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

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

Encrypting the password

Entries in the *libtcl*.cfg* file are in human-readable format. Sybase provides a `pwdcrypt` utility for basic password encryption. `pwdcrypt` is a simple algorithm that, when applied to keyboard input, generates an encrypted value that can be substituted for the password. `pwdcrypt` is located in `$$SYBASE/$SYBASE_OCS/bin`

From the `$$SYBASE/$SYBASE_OCS` directory, enter:

```
bin/pwdcrypt
```

Enter your password twice when prompted.

`pwdcrypt` generates an encrypted password. For example:

```
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706
```

Copy and paste the encrypted password into the *libtcl*.cfg* file using any standard ASCII-text editor. Before encryption, the file entry appears as:

```
ldap=libldap.so
ldap://dolly/dc=sybase,dc=com????bindname=cn=Manager,dc=sybase,dc=com?secret
```

Replace the password with the encrypted string:

```
ldap=libldap.so
ldap://dolly/dc=sybase,dc=com????bindname=cn=Manager,dc=sybase,dc=com?
```

0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706

Warning! Even if your password is encrypted, you should still protect it using file-system security.

Performance

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

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 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_servename* file.

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

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

- 1 Shutdown server. See Chapter 2, “Starting and Stopping Servers.”

- 2 Using a standard ASCII text editor, edit the `$$SYBASE/$SYBASE_OCS/config/libtcl.cfg` file to add or remove the directory service. See “Enabling LDAP directory services” on page 73.
- 3 Use `dsedit` or `dscp` and add the server entry to directory service. See “Adding a server to the directory services” on page 75.
- 4 Using a standard ASCII text editor, edit `$$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.

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*.

Topic	Page
Overview of localization support	81
Character set conversion	89
Sort orders	90
Language modules	93
Localization	94
Changing the localization configuration	98

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 12.5 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
 - Polish
 - Spanish

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
- `roman8` character set
- 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.

For information on using `sqlloc`, see Chapter 7, “Customizing Localization for Adaptive Server.”

By default, when Adaptive Server and Backup Server are installed on Compaq Tru64, IBM, 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 will work 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 version 12.5:

- Arabic – see Table 7-1.
- Baltic – see Table 7-2.

- Chinese, Simplified – see Table 7-3.
- Chinese, Traditional – see Table 7-4
- Cyrillic – see Table 7-5.
- Eastern European – see Table 7-6.
- Greek – see Table 7-7.
- Hebrew – see Table 7-8.
- Japanese – see Table 7-9.
- Korean – see Table 7-10.
- Thai – see Table 7-11.
- Turkish – see Table 7-12.
- Unicode – see Table 7-13.
- Vietnamese – see Table 7-14.
- Western European – see Table 7-15.

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

- Checkmark (x) – the character set requires Unilib conversion.
- No checkmark – 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: 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: Baltic character sets

Character set	Unilib required	Description
cp1257	X	Microsoft Windows Baltic

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

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

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: 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: 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: Hebrew character sets

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

Table 7-9: Japanese character sets

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

Table 7-10: Korean character sets

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

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: 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: Unicode character set

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

Table 7-14: Vietnamese character set

Character set	Unilib required	Description
cp1258	X	Microsoft Windows Vietnamese

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
cp863	X	IBM PC Canadian French code page
cp1252	X	Microsoft Windows US (ANSI)
iso_1		ISO 8859-1 Latin-1
mac		Standard Macintosh coding
roman8		HP ROMAN8
iso 885915	X	ISO 8859-15 Latin-1 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 need to install the character set definition files on the server for all the character sets being used by your clients. Built-in conversion support is available for many character sets.
- Unilib conversion support is available for all character sets supported by Sybase. To enable Unilib conversion, you must use `sp_configure` and turn enable unicode conversions on. 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 will occur.

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

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

Sort orders

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

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

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

Available sort orders

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

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

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

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

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

`%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 Digital, 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.

Sort order name	Description
Dictionary order case insensitive with preference	<p>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.</p> <p>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.</p> <hr/> <p>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.</p> <hr/>
Alternate dictionary order case sensitive	<p>Case-sensitive alternate dictionary sort order with lowercase variants sorted before uppercase.</p> <p>Use with several of the Western European languages.</p>
Alternate dictionary order case insensitive accent insensitive	<p>Case-insensitive and accent-insensitive alternate dictionary sort order.</p> <p>Use with several of the Western European languages.</p>
Alternate dictionary order case insensitive uppercase preference	<p>Case-insensitive alternate dictionary sort order with uppercase preference.</p> <p>Use with several of the Western European languages.</p>
Spanish dictionary order case sensitive	<p>Case-sensitive Spanish dictionary sort order.</p> <p>Use with Spanish and for most Latin American locales.</p>
Spanish dictionary order case insensitive	<p>Spanish case-insensitive dictionary sort order.</p> <p>Use with Spanish and for most Latin American locales.</p>
Spanish dictionary order case insensitive accent insensitive	<p>Spanish case-insensitive and accent-insensitive dictionary sort order.</p> <p>Use with Spanish and for most Latin American locales.</p>
Scandinavian dictionary order case sensitive	<p>Case-sensitive dictionary sort order.</p> <p>Use with Scandinavian languages.</p>
Scandinavian dictionary order case insensitive uppercase preference	<p>Case-insensitive and accent-insensitive dictionary sorting, with uppercase preference.</p> <p>Use with Scandinavian languages.</p>

To see the sort orders that are available, use `sqlloc` to display the sort orders for the character sets you plan to use.

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

To install a new language module:

- 1 Load the language module software from the distribution media. You must load this software into the same directory in which you loaded Adaptive Server.
- 2 Reconfigure the language and, if necessary, the character set and sort order for Adaptive Server. For instructions, see “Changing the localization configuration” on page 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/	<i>charsets</i>	<i>charset_name</i>	*.srt files
		<i>charset_name...</i>	<i>charset.loc</i>
		<i>unicode</i>	*.uct files
	<i>locales</i>	<i>language_name</i>	<i>charset_name</i>
		<i>language_name...</i>	<i>charset_name...</i>
		<i>locales.dat</i>	
		<i>message</i>	<i>language_name</i>
			<i>language_name...</i>

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 comes with 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 which is enclosed in brackets:
 - For Sun Solaris, [*sun_svr4*]
 - For HP, [*hp ux*]
 - For IBM, [*aix*]
 - For Compaq Tru64 [*axposf*]
- 3 Make sure the section contains an entry for the language (*syb_language*) and character set (*syb_charset*) combination that you want to use.
 - If an entry does not exist, continue with step 4.

- If an entry does exist, continue with step 5.

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.

For Adaptive Server

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 following:
 - Select the default language.
 - Select the default character set.
 - Select the default sort order.
- 6 Select any other languages you want to install. You may only select languages that are supported by the default character set.

The Add and Remove Languages window lists all Sybase supported languages.
- 7 The Localization Summary window summarizes the configuration options you selected. Click OK to confirm your selections.

The Status Output window notifies you upon completing the installation.

For Backup Server

When you 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 the `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 the `charset` utility to load any additional character sets. See “`charset` utility” on page 104 for more about this utility.

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

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

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

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

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

Replace *sort_order_id* with the ID for your sort order. See Table 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 boots, rebuilds all the system indexes, then shuts down. Reboot a second time to bring the server up in a stable state.

Sort orders

Table 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	<i>binary.srt</i>	50
Cyrillic	Dictionary order, case sensitive, accent sensitive	<i>cyrdict.srt</i>	63
	Dictionary order, case sensitive, accent sensitive	<i>cyrnocs.srt</i>	64
English French German (These sort orders work with all Western European character sets.)	Dictionary order, case sensitive, accent sensitive	<i>dictiona.srt</i>	51
	Dictionary order, case insensitive, accent sensitive	<i>nocase.srt</i>	52
	Dictionary order, case sensitive, accent sensitive, with preference	<i>nocasepr.srt</i>	53
	Dictionary order, case insensitive, accent insensitive	<i>noaccent.srt</i>	54
English French German (These sort orders work only with CP 850.)	Alternate dictionary order, case sensitive	<i>altdict.srt</i>	45
	Alternate dictionary order, case sensitive, accent insensitive	<i>altmoacc.srt</i>	39
	Alternate dictionary order, case sensitive, with preference	<i>altnocsp.srt</i>	46
Greek (This sort order works only with ISO 8859-7.)	Dictionary order, case sensitive, accent sensitive	<i>elldict.srt</i>	65

Language or script	Sort orders	File name	ID
Hungarian (These sort orders work only with ISO 8859-2.)	Dictionary order, case sensitive, accent sensitive	<i>hundict.srt</i>	69
	Dictionary order, case insensitive, accent sensitive	<i>hunnoac.srt</i>	70
	Dictionary order, case insensitive, accent insensitive	<i>hunnocs.srt</i>	71
Russian (This sort order works with all Cyrillic character sets except for CP 855.)	Dictionary order, case sensitive, accent sensitive	<i>rusdict.srt</i>	58
	Dictionary order, case insensitive, accent sensitive	<i>rusnocs.srt</i>	59
Scandinavian (These sort orders work only with CP 850.)	Dictionary order, case sensitive, accent sensitive	<i>scandict.srt</i>	47
	Dictionary order, case insensitive, with preference	<i>scannocp.srt</i>	48
Spanish	Dictionary order, case sensitive, accent sensitive	<i>espdict.srt</i>	55
	Dictionary order, case insensitive, accent sensitive	<i>espnocs.srt</i>	56
	Dictionary order, case insensitive, accent insensitive	<i>espnoac.srt</i>	57
Thai	Dictionary order	<i>dictionary.srt</i>	51
Turkish (These sort orders work only with ISO 8859-9.)	Dictionary order, case sensitive, accent sensitive	<i>turdict.srt</i>	72
	Dictionary order, case insensitive, accent insensitive	<i>turnoac.srt</i>	73
	Dictionary order, case insensitive, accent sensitive	<i>turnocs.srt</i>	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

Character sets	Sybase name
CP 437	cp437
CP 850	cp850
CP 852	cp852
CP 855	cp855
CP 857	cp857
CP 860	cp860
CP 863	cp863
CP 864	cp864
CP 866	cp866
CP 869	cp869
CP 874	cp874
CP 932	cp932
CP 936	cp936
CP 950	cp950
CP 1250	cp1250
CP 1251	cp1251
CP 1252	cp1252
CP 1253	cp1253
CP 1254	cp1254
CP 1255	cp1255
CP 1256	cp1256
CP 1257	cp1257
CP 1258	cp1258
DEC Kanji	deckanji
EUC-CNS	euccns
EUC-GB	eucgb
EUC-JIS	eucjis
EUC-KSC	eucksc
GREEK8	greek8
ISO 8859-1	iso_1
ISO 8859-2	iso88592
ISO 8859-5	iso88595
ISO 8859-6	iso88596
ISO 8859-7	iso88597
ISO 8859-8	iso88598
ISO 8859-9	iso88599
ISO 8859-15	iso885915

Character sets	Sybase name
Koi8	koi8
Macintosh Cyrillic	mac_cyr
Macintosh Eastern European	mac_ee
Macintosh Greek	macgrk2
Macintosh Western European	mac
Macintosh Turkish	macturk
ROMAN8	roman8
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.

Keywords and Options	Description
-S	Specifies the name of the server. If not specified, charset uses the DSQUERY environment variable to identify the server name. If 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.
<i>sort_order</i>	When charset is used to load the default character set and sort order, <i>sort_order</i> is a mandatory parameter specifying the name of the sort order file to be used by Adaptive Server. When loading additional character sets, use <i>charset.loc</i> to indicate the name of the character-set files.
<i>charset</i>	Specifies the directory of the character set to be used by Adaptive Server.

Logging Error Messages and Events

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

Topic	Page
Adaptive Server error logging	107
Setting error log paths	108
Managing messages	109

Adaptive Server error logging

Each time Adaptive Server starts it begins to write 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.
- Retains its contents until you rename, move, or empty the file.

Note When you want to make more memory 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 109.

Types of information logged

The Adaptive Server Error Log stores the following types of messages:

- Start-up messages from Adaptive Server
- Backtraces and stack traces from Adaptive Server
- Fatal error messages (severity level 19 and higher)
- Kernel error messages
- Informational messages

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: `$SYBASE/ASE-12_5/install/error.log`
- Backup Server: `$SYBASE/install/backup.log`
- Monitor Server: `$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 `isql` command 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

The error log path can be changed by editing the `$SYBASE/ASE_12_5/install/RUN_server_name` file.

For example, to change the error log path from `$SYBASE/ASE-12_5/bin/dataserver -d/Devices/ASE_2K.dat -sASE_2K -i/ASE_125 -e/$SYBASE/ASE-12_5/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 *runserver* file, see “Using a runserver file” on page 15.

Managing messages

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

- Use the `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 *Adaptive Server 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*.

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.

Topic	Page
Managing database devices	111
Backing up and restoring data	113
Using disk drives	120

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 up to 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.

To create a database device using the file */devices/user1.dat*:

- 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/bin> isql -Usa -Ppassword -Sserver_name
```

On Mac OS X, enter:

```
$SYBASE/$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*.

Backing up and restoring data

Sybase supports tape drives and hard disks for backing up and restoring databases.

- The dump command backs up databases and transaction logs.

To back up your databases, follow the instruction for “Using a tape drive” on page 113.

- The load command restores databases and transaction logs.

To copy Sybase-supplied databases, see *Installation Guide* for your platform.

For more information about backing up and restoring databases, see the *System Administration Guide*.

Using a tape drive

Note Tape backup is not available on Mac OS X.

Sybase software can back up and restore databases to tape drives that are compatible with Windows NT, including:

- 1/4-inch cartridge
- 4-mm and 8-mm digital audio tape (DAT) formats

To back up a database to a tape drive:

- 1 Install the tape drive according to the manufacturer’s instructions.

This task includes installing an NT-compatible driver for the tape drive by using the Add/Remove buttons in the Tape Devices dialog box from the Control Panel. For instructions, see your tape drive and NT operating system documentation.

- 2 Start isql, and connect to Adaptive Server:

```
/sybase/bin> isql -Usa -Ppassword -Sserver_name
```

On Mac OS X, enter:

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

- 3 Use the NT tape device name with isql statements to name the tape drive.

For more information about using the dump and load commands, see “Examples of backing up and restoring databases” on page 115.

Setting the maximum capacity for a tape drive

To run properly, the dump command needs to know the maximum capacity of the destination tape drive. It determines this capacity in one of two ways, depending on the tape device name that you use:

- The physical device name – you must include the capacity parameter in the dump command. This parameter specifies the maximum number of bytes to write to a tape device.

Check your tape’s capacity, and keep the following in mind:

- The minimum value that the capacity parameter can accept is 5 databases pages, 2K each.
- The maximum value that the capacity parameter can accept is 4,294,967,295K.
- The actual capacity value should be 70 to 80 percent of the true capacity of the tape.
- If you omit the capacity parameter for NT, Backup Server writes the maximum number of bytes for the specified tape device.
- The logical device name – the command uses the size parameter stored in the sysdevices system table.

You can override that value by using the capacity parameter as described in the preceding list item.

Dumping across a network

Backup Server may issue an “Access denied” message when you try to dump to a device mounted from across a network, particularly if you started Backup Server from Sybase Central.

To work around this restriction, configure Backup Server to start with a regular user account, rather than the NT default account. The user should have the permission to access remote drives.

To start Backup Server with a regular user account:

- 1 Double-click the Services icon from the Control Panel.

- 2 Select the Backup Server to configure, and click the Startup button.
- 3 In the Log On As area, name the user in the This Account box to activate that option, type the user's password, and confirm that password.
- 4 Click OK to exit the Services dialog box.
- 5 Click Close to exit Services.

Examples of backing up and restoring databases

Following are examples of using the dump and load commands for backup and recovery of Adaptive Server database on NT. For more information, see the *System Administration Guide* and the *Performance and Tuning Guide*.

User databases

The following sections provide examples for backing up and restoring user database.

Specifying a database and device

This section provides examples on using a tape drive and a *.dat* file as the backup and recovery resources.

Using a tape drive

In the commands in this section, the physical device name TAPE0 replaces the *stripe_device* variable.

To use the first tape device to back up and load a database:

```
1> dump database pubs2 to "\\.\TAPE0"
2> go
1> load database pubs2 from "\\.\TAPE0"
2> go
```

For Mac OS X, enter:

```
1> dump database pubs2 to /Volumes/volname
2> go
1> load database pubs2 from /Volumes/volname
2> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

Using a *.dat* file

To back up and load the pubs2 database using a *.dat* file:

```
1> dump database pubs2 to "/work/backups/backup1.dat"
2> go
1> load database pubs2 from "/work/backups/backup1.dat"
2> go
```

Specifying a remote Backup Server

To back up to and restore from the first tape drive on a remote NT Backup Server named REMOTE_BKP_SERVER:

```
1> dump database pubs2 to "\\.\TAPE0" at REMOTE_BKP_SERVER
2> go
1> load database pubs2 from "\\.\TAPE0" at REMOTE_BKP_SERVER
2> go
```

For Mac OS X, enter:

```
1> dump database pubs2 to /Volumes/volname
2> go
1> load database pubs2 from /Volumes/volname
2> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

Naming a backup file

To back up a transaction log, the syslogins system table and create a default backup file name:

```
1> dump tran publications to "\\.\TAPE0"
2> go
```

For Mac OS X, enter:

```
1> dump tran publications to /Volumes/volname
2> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

To restore the log using the default file name in the file clause:

```
1> load tran publications from "\\.\TAPE0"
with file = "cations930590E100"
2> go
```

For Mac OS X, enter:

```
1> load tran publications from /Volumes/volname with
file = "cations930590E100"
```

```
2> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

Note The dump command uses the last 7 characters in the database name publications to create the transaction log backup file *930590E100*. See the *System Administration Guide*.

In the following example, as directed by the user, the 15-character file name, *personnel97sep111800* records the following backup information:

- The database name (personnel)
- The date (*97sep11*) – September 11, 1997
- The time (*1800*) – 18:00 or 6:00 p.m

To back up the personnel database using the file clause to create the file name:

```
1> dump database personnel to "\\.\TAPE0"
2> with file = "personnel97sep111800"
3> go
```

For Mac OS X, enter:

```
1> dump database personnel to "/Volumes/volname"
2> with file - "personel97sept111800"
3> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

To restore the personnel database by advancing the tape automatically to *personnel97sep111800* before restoring:

```
1> load database personnel from "\\.\TAPE0"
2> with file = "personnel97sep111800"
3> go
```

For Mac OS X, enter:

```
1> load database personnel from "/Volumes/volname"
2> with file = "personnel97sep111800"
3> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

Note The file names in the preceding examples are valid only for systems that use the NTFS file system. If you are using a FAT-based file system, file names are limited to 8 characters with a 3-character extension.

Specifying additional dump devices

To back up the database to three devices using the *stripe* on parameter and *three* devices:

```
1> dump database personnel to "\\.\TAPE0"
2> stripe on "\\.\TAPE1"
3> stripe on "\\.\TAPE2"
4> go
```

For Mac OS X, enter:

```
1> dumpt database personnel to "Volumes/volname"
2> stripe on /Volumes/volname
3> stripe on /Volumes/volname
4> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

To restore the database using the *stripe* on parameter and *two* devices:

```
1> load database personnel from "\\.\TAPE0"
2> stripe on "\\.\TAPE1"
3> go
```

For Mac OS X, enter:

```
1> load database personnel from /Volumes/volname
2> stripe on /Volumes/volname
3> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

To back up a database using three devices, each attached to the remote Backup Server, REMOTE_BKP_SERVER:

```
1> dump database personnel
2> to "\\.\TAPE0" at REMOTE_BKP_SERVER
3> stripe on "\\.\TAPE1" at REMOTE_BKP_SERVER
```

```
4> stripe on "\\.\TAPE2" at REMOTE_BKP_SERVER
5> go
```

For Mac OS X, enter:

```
1> dump database personnel
2> to /Volumes/volname at REMOTE_BKP_SERVER
3> stripe on /Volumes/volname at REMOTE_BKP_SERVER
4> stripe on /Volumes/volname at REMOTE_BKP_SERVER
5> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

Tape handling options

To initialize two devices to overwrite the existing contents with the new transaction log backups:

```
1> dump transaction personnel to "\\.\TAPE0"
2> stripe on "\\.\TAPE1" with init
3> go
```

For Mac OS X, enter:

```
1> dump transaction personnel to "/Volume/volname"
2> stripe on /Volumes/volname with init
3> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

Getting information about files

To return header information for the first file on the tape:

```
1> load database personnel from "\\.\TAPE0"
2> with headeronly
3> go
```

For Mac OS X, enter:

```
1> load database personnel from /Volumes/volname
2> with headeronly
3> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

To return header information for the file *personnel92229510945*:

```
1> load database personnel from "\\.\TAPE0"  
2> with headeronly, file = "personnel9229510945"  
3> go
```

For Mac OS X, enter:

```
1> load database personnel from /Volumes/volname  
2> with headeronly, file = "personnel92295190945"  
3> go
```

where *volname* is the name of a FireWire device, USB device, or a locally mounted disk drive.

System databases

You can restore the following system databases:

- master
- model
- sybsystemprocs

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

Using disk drives

The overall performance in an I/O-bound application is determined by the number of disk drives on a system, not by the amount of space available. A single disk drive might not be able to deliver the number of I/Os per second that are needed for your Adaptive Server application.

To achieve your performance objectives for an application, you must have enough disk drives to give the necessary number of I/Os per second.

Note Your disk drive requirements may not be directly related to the size of your database. Depending on the amount of I/O you need, you may have free space on your disk drives.

Monitoring disk usage

Sybase recommends that you distribute data in heavily used databases across multiple disks. To do this effectively, you need to monitor disk usage.

If one or more disks are consistently very busy, distribute the database objects on those disks to other devices. This strategy spreads out the work among disks and allows for greater data throughput.

You can use stored system procedures on Adaptive Server to monitor the disk space:

- To determine which devices a specific database is using, run `sp_helpdevice` or `sp_helpdb`.

For more information, see `sp_helpdevice` and `sp_helpdb` in the *Reference Manual*; also see the *System Administration Guide*.

- To check for disk space usage rates and I/O contention, run `sp_sysmon`.

For more information, see `sp_sysmon` in the *Reference Manual*; see also the *Performance and Tuning Guide*.

Adding Optional Functionality to Adaptive Server

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

- Auditing – you can detect access of the system and misuse of resources by using auditing to track security-related system activity in an audit trail.

Topic	Page
Adding auditing	123
Installing online help for Transact-SQL syntax	131

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

The sybsecurity device and database

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

Tables and devices for the audit trail

Adaptive Server stores the audit trail in system tables, named sysaudits_01 through sysaudits_08. For example, if you have two audit tables, they are named sysaudits_01 and sysaudits_02. At any given time, only *one* of the audit tables is *current*. Adaptive Server writes all audit data to the current audit table. A System Security Officer can use sp_configure to set or change which audit table is current.

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

Device for syslogs systems table

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

Overview of audit installation

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

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

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

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

Installing auditing

To configure Adaptive Server for auditing:

- 1 Log in to your machine using your Sybase System Administrator (“sa”) user account.
- 2 Start auditinit at the UNIX prompt:

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

 auditinit displays the following menu:

```
AUDITINIT
1. Release directory: /usr/u/sybase
2. Configure a Server product
```
- 3 Select Configure a Server Product.
- 4 Select Adaptive Server.
- 5 Select Configure an Existing Sybase Server.
- 6 Select the server to configure.
- 7 Provide the SA Password for the server you selected.
- 8 From the Sybase Server Configuration screen, select Configure Auditing.

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

CONFIGURE AUDITING

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

List of devices for the audit tables:

Logical name	Physical name	Segment name	Table name	Size
--------------	---------------	--------------	------------	------

Device for the audit database transaction log:

Logical name	Physical name	Segment name	Table name	Size
--------------	---------------	--------------	------------	------

- 9 From the Configure Auditing screen, select Configure Auditing.
 auditinit redisplay the Configure Auditing menu with the value “yes” displayed for Configure Auditing.

Creating a device for an audit table

To create a device for an audit table:

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

auditinit displays the following menu:

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

- 2 Select Sybsecurity Physical Device Name.
- 3 Enter the *full path* of the physical device (raw partition) that you located in “Pre-installation tasks for auditing devices” on page 125.

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

```
/dev/path_to_partition
```

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

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

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

- 4 Press Return to acknowledge the warning.

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

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

- 5 Proceed through the remaining items on this menu.

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

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

CONFIGURE AUDITING

1. Configure auditing: yes

2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:

Logical name	Physical name	Segment name	Table name	Size
--------------	---------------	--------------	------------	------

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

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

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

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

CONFIGURE AUDITING

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

List of devices for the audit tables:

Logical name	Physical name	Segment name	Table	name	Size
--------------	---------------	--------------	-------	------	------

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

Creating a device for the audit database transaction log

To create a device for the audit database transaction log:

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

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

ADD/CHANGE A NEW DEVICE FOR AUDITING

1. sybsecurity physical device name:
 2. Logical name of the device:
 3. Size of the new device (Meg):
 4. Device size for auditing:
- 2 Select Sybsecurity Physical Device Name.

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

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

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

- 3 Enter the full path name of a physical device.

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

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

- 4 Press Return to acknowledge this warning.

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

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

- 5 Proceed through the remaining items on this menu. As you do so, be aware of the following:
 - Sybase recommends a minimum size of 2MB for the size of the transaction log.
 - auditinit displays the size in both Size of the Device and in Device Size for Auditing in the Add/Change a New Device for Auditing menu.
 - The Device Size for Auditing default value is equal to the size of the device, based on the assumption that you may want to devote the entire device to log for the auditing task. If you want to use only a subset of the device, you can edit the Size of the Device value.
- 6 Press Ctrl+A to accept the settings displayed in the Add/Change a New Device for Auditing menu.

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

```
CONFIGURE AUDITING
```

```
1. Configure auditing: yes
```

- 2. Add a device for audit table(s)
- 3. Add a device for the audit database transaction log
- 4. Delete a device entry
- 5. Change a device entry

List of devices for the audit tables:

Logical name	Physical name	Segment name	Table	
name	Size			
6. Audit_01'	/secret1/sybase_	dr/install/aud1.dat'	sysaudits_01	5
7. Audit_02'	/secret1/sybase_	dr/install/aud2.dat'	sysaudits_02	5
8. auditlog	/secret1/.../auditlog.dat	logsegment	syslogs	2

- 7 When you are ready to execute the audit configuration, press Ctrl+A. auditinit returns you to the Sybase Server Configuration screen.
- 8 Press Ctrl+A again. auditinit prompts with:

Execute the Sybase Server Configuration now?

- 9 Enter “y” (yes).

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

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

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

Deleting a device entry To delete a device entry:

- 1 Select Delete a Device Entry from the Configure Auditing menu.
- 2 Enter the number of the device to delete.
- 3 Press RETURN.

Changing a device entry To change a device entry:

- 1 Select Change a Device Entry from the Configure Auditing menu.
- 2 Enter the number of the device to change.

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

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

- 3 Select each remaining entry you want to change.
- 4 Press Ctrl+A to save the new entries.

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 the `sp_syntax` system procedure. For more information on `sp_syntax`, see the *Adaptive Server 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 receive 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. This configuration is not recommended 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:

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.

Make a copy of the original script. Be sure you can access this copy, in case you experience problems with the edited script.

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 132.

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
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.

5 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. For example:

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

- 6 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."

Index

A

- accented letters 12, 91
- Adaptive Server
 - character set, changing 83
 - character sets 89
 - client communications with 47
 - conversions between, and clients 89
 - customizing features 45
 - default character set 83
 - default configuration 45, 46
 - default sort order 83
 - error log path 109
 - language, changing 83
 - localizing 83
 - multiple disk drives and 120
 - naming in interfaces file 54
 - shutting down 25
 - sort order 83
 - starting from UNIX command line 14
 - starting with operating system 16
 - start-up script 21
- adding a server, LDAP 75
- address component in interfaces files
 - described 57
 - SPX protocol entry 55
 - TCP protocol entry 55
- address, server 48
- administrator
 - operating system 3
 - Sybase system 3
- API component in interfaces file
 - described 54
 - SPX protocol entry 57
- Arabic character sets 85
- audit system 123
- audit trail
 - overview 123
 - system audit tables 124
- auditing
 - database for 124
 - device for 124
 - global options 124
 - installing using the auditinit utility 124
 - installing using the installsecurity script 124
 - process 124
 - tables for tracking 124
- auditinit utility 8
- automatically restarting Adaptive Server 20

B

- backup operations 113
 - across a network 114
- Backup Server
 - character sets 89, 99
 - configuring 94, 98
 - default configuration 46
 - remote 116
 - starting from UNIX command line 14
 - starting with operating system 16
- Backup Server, default, for Adaptive Server 46
- binary sort order 83, 91
- Bourne shell xiv, 31

C

- C shell xiv, 31
- case sensitivity, in SQL xv
- changing
 - character sets 83, 98
 - languages 98
 - server language 83
 - sort order 98
- character sets 89
 - accented letters in 12
 - changing 83, 84, 98
 - client selection of 84

Index

- code conversions and 89
 - configuring 99
 - converting between 89
 - databases and 90
 - default 83
 - in a heterogeneous environment 89
 - sort orders and 90
 - U.S. English 12
 - charsets directory 91, 94
 - about the 95
 - Chinese character sets 86
 - client interfaces files
 - difference between client and server versions 49
 - heterogeneous 51
 - homogeneous 51
 - clients
 - Adaptive Server communications with 47
 - applications and locales.dat file 96
 - conversion between, and server 89
 - default character set 84
 - DSQUERY and 61
 - file servers 61
 - code conversion
 - between character sets 89
 - collating sequences. *See* sort orders 90
 - commands
 - gr_osview 42
 - iostat 41, 42
 - maxfiles 34
 - maxfiles_lim 34
 - netstat 41, 42
 - netstat -v 41
 - no -a 42
 - osview 42
 - ps 41, 42, 43
 - sar 41, 42
 - setenv 31
 - setperm_all 33
 - time 41, 42
 - timex 43
 - vmstat 41, 42
 - common.loc file 95
 - communications between client and Adaptive Server 47
 - configurations
 - default 45, 46
 - configurations, default 46
 - configuring
 - Backup Server 94, 98
 - character sets 99
 - conventions in this book xvi
 - conventions. *See* syntax
 - conversions, Unicode character 85
 - converting between character sets 89
 - create database command, system tables created by 6
 - creating
 - interfaces files 58, 59
 - interfaces files automatically 48
 - interfaces files for beginners 58
 - master interfaces files with dscp utility 58
 - master interfaces files with dsedit 58
 - master interfaces files with text editor 58
 - .cshrc file xiv, 31
 - customizing installations 45
 - Cyrillic character sets 86
- ## D
- data translation 81
 - database devices
 - master 6
 - sybsyntax 132
 - sybssystemdb 6
 - sysprocsdev 6, 7
 - databases 90
 - backing up and restoring 113, 120
 - dbccdb 8
 - devices 111
 - master 6, 7
 - master, restoring 120
 - media for backups and restores 113
 - model 6
 - sample 7, 8
 - sybsecurity 8
 - sybssystemprocs 7
 - sybssystemprocs, restoring 120
 - system databases, dump and load examples 120
 - tempdb 6
 - user, backup and restore examples 115
 - dataserver utility 28
 - dbcc checkstorage, database for 8
 - dbcc error messages 44

- dbccdb database 8
- debug service type 54
- Dec-Kanji character set 89
- default
 - character set for Adaptive Server 83
 - character set, changing 83
 - character sets installed by 83
 - language for Adaptive Server 83
 - language, changing 83
 - sort order 83
- delay_interval component in interfaces files 54
- device component in interfaces files 55, 57
- devices
 - files 111
 - using additional 118
- dictionary sort orders 91
 - Scandinavian 92
 - Spanish 92
- Digital UNIX
 - hardware error messages 39
 - iostat command 40, 41
 - monitoring systems 41
 - netstat command 41
 - network protocol 54
 - ps command 41
 - time command 41
 - timeout period 38
 - vmstat command 41
- directories xvi
 - charsets 91, 95
 - localization 94
 - run control 21
- directory schema, LDAP 71
- disk drives
 - multiple 120
- disk usage, monitoring 121
- displaying current file descriptors 33
- documentation
 - Adaptive Server translated 82
- dscp
 - adding an LDAP server 76
- dscp utility 10
 - creating master interfaces files with 58
- dsedit
 - adding an LDAP server 75
- dsedit utility 10

- creating master interfaces files with 58
- DSLISN environment variable 4
 - introduction to 32
 - multiple network handlers and 61
- DSQUERY environment variable 4
 - client connections and 61
 - described 50
 - introduction to 32
 - multiple networks, using different values 63
 - naming in 61
- dump command 113, 115, 119

E

- environment variables
 - DSLISN 4, 32, 61
 - DSQUERY 4, 32, 50, 61
 - LD_LIBRARY_PATH 5
 - LIBPATH 5
 - LM_LICENSE_FILE 5
 - PATH 5, 32
 - SYBASE 4
 - SYBASE_ASE 4
 - SYBASE_FTS 5
 - SYBASE_OCS 4
 - SYBASE_SYSAM 5
- error log paths 45, 108, 109
 - configuring 108
- error logging 108
 - configuring 108
- errors in dbcc messages 44
- ESPs. See extended stored procedures (ESPs)
- /etc/rc2.d directory 20
- /etc/services file 55
- ether placeholder in interfaces files 54
- EUC-JIS character set 89
- expressions, types of xv
- extended stored procedures (ESPs) 2

F

- file descriptors
 - displaying current 33
- file servers 61

Index

files

- .cshrc xiv, 31
- .login 31
- .profile 31
- common.loc 95
- device files 111
- locales.dat 95
- localization 82
- localized error messages (.loc) 95
- runserver 15
- shared memory 29
- sort order definition (.srt) files 90

formatting for local date, time, and currency 95

French sample database 8

G

German sample database 8

globalization support, Sybase 46, 81, 94, 98

gr_osview command (SGI) 42

Greek character sets 87

H

hard drives 113

hardware error messages 39

- Digital UNIX 39
- IBM RS/6000 39
- Sun Solaris 40

Hebrew character sets 87

heterogeneous environments 84, 89

- described 51
- interfaces files and 51

high-availability products 11

homogeneous environments

- described 51
- interfaces files and 51

host component in interfaces files 54

host name

- determining 54

HP-UX

- hardware error messages 39
- netstat command 41
- network protocol 54

- ps command 41
- sar command 41
- time command 41
- timeout period 38
- vmstat command 41

I

I/O monitoring 41

I/O-bound applications 120

IBM RS/6000

- hardware error messages 39
- iostat command 41
- monitoring systems 41
- netstat command 41
- netstat -v command 41
- network protocol 54
- no -a command 42
- ps command 42
- time command 42
- timeout period 38
- vmstat command 41

installation directory, Sybase xvi

interfaces file 9

interfaces files

- Adaptive Server, naming in 54
- Adaptive Server, used by 49
- address component for SPX 55
- API component in 54
- API component in SPX 57
- automatic creation of 48
- client and server versions, differences in 49
- clients, used by 48
- contents of 49
- creating automatically 48
- creating master files with dscp 58
- creating master files with dsedit 58
- creating master files with text editor 58
- creating, for beginners 58
- debug service type 54
- default location 48
- delay_interval component 54
- described 48
- device component 55
- device component in SPX 57

- ether placeholder 54
- format described 53
- heterogeneous environments and 50
- homogeneous environments and 50
- host component 54
- location 48
- loghost placeholder 59
- machine component 54
- master service type 54
- multiple network listeners 60
- multiple networks 50, 60
- network component 54
- port component 55
- port/object number component 55
- protocol component 54
- protocol component in SPX 57
- query port backup configuration 64
- query service type 54
- retry_attempt component 54
- servername component 54
- servers, used by 48
- service_type component 54
- spaces in 56
- tab characters in 56
- unique elements in entries 59
- used by clients 50
- international systems
 - support for 81
 - Sybase support for 81
- interpubs sample database 8
- iostat command
 - Digital UNIX 40, 41
 - IBM RS/6000 41
 - Sun Solaris 42
- iso-Latin1 character set 83

J

- Japanese sample database 8
- jpubs sample database 8

K

- KEEPALIVE option, TCP/IP 37

- kill command 27, 29
- Korean character sets 87
- Korn shell xiv

L

- language Modules 82
- language modules 93, 94
 - default 46
 - installing new 93
 - localization files 82
 - memory requirements for 99
- languages 12
 - changing 98
 - error reporting in specific 95
 - selecting message 93
 - translation support 81
- Latin character sets 86
- LD_LIBRARY_PATH environment variable 5
- LDAP
 - access restrictions 68
 - adding a server 75
 - defined 67
 - directory definitions 69
 - directory schema 71
 - enabling 73
 - multiple directory services 77
 - sample entry 70
 - specifying in libtcl.cfg 72
 - versus the interfaces file 69
- LDAP libraries
 - environment variables 74
 - location of 74
- LDAP server
 - using dscp to add and modify 76
 - using dsedit to add and modify 75
- ldapurl
 - defined 72
 - example 72
 - keywords 73
- letter case in sort orders 91
- LIBPATH environment variable 5
- libtcl*.cfg file 72
 - format of 72
 - location of 72

Index

- purpose of 72
- libtcl*cfg
 - password 78
- limits for file descriptors 34
- listener service 49
- listing backup files on a tape 119
- LM_LICENSE_FILE environment variable 5
- load command 113, 115, 119
- loc files 95
- local date, time, and currency formatting 95
- locales directory 94
- locales.dat file 95
- localization 81
 - changing languages 83
 - changing the configuration 98
 - common, information 95
- localization support 46
- log file contents 108
- loghost in interfaces files 59
- login
 - root 3
 - sa 3
 - superuser 3
 - sybase 4
- .login file 31

M

- machine component in interfaces files 54
- master
 - interfaces file 58, 59
 - service type 54
- master database 6
- master database, restoring 120
- master device 6
- maxfiles kernel parameters 34
- maxfiles_lim kernel parameters 34
- mbuf pools 42
- media supported for database backups 113
- messages
 - hardware errors 39
- messages, selecting language for 93
- model database 6
- Monitor Server
 - default configuration for 46

- monserver command 16
 - shutting down 27
 - starting from UNIX command line 14
 - starting with operating system 16
- monitoring
 - I/O 41
 - network status 41
 - operating system resources 40
 - virtual memory usage 41
- monitoring systems
 - Digital UNIX 41
 - IBM RS/6000 41
 - NCR 42
 - Sun Solaris 43
- monserver command 16
- multiple directory services
 - LDAP 77
- multiple installations
 - affecting interfaces files 49
 - creating one interfaces file for 58, 59
- multiple networks
 - interfaces files and 50
 - interfaces files for 60
 - used as a backup in case of network failure 64

N

- named pipes
 - default pipe 45, 46
- naming requirements for servers 54
- NCR
 - monitoring systems 42
- netstart -v command (IBM RS/6000) 41
- netstat command
 - Digital UNIX 41
 - HP-UX 41
 - IBM RS/6000 41
 - Sun Solaris 42
- network component in interfaces files 54
- network configuration
 - backing up files 114
- network protocols
 - DECnet 9
 - Digital UNIX 54
 - HP-UX 54

- IBM RS/6000 54
- SPX 9
- Sun Solaris 54
- TCP/IP 9
- UnixWare 54
- network support
 - default configuration 45, 46
- networks
 - backup connections and 64
 - DSQUERY and 61
 - failure of 65
 - interfaces files 48
 - monitoring status 41
 - multiple 50
- networks and restarting servers 20
- no -a command (IBM RS/6000) 42
- Novell Protocol Suite Version 3.0.1 56

O

- objectid.dat
 - location of 75
- online syntax help 131
- open_max parameter 34
- operating system
 - administrator 3
 - monitoring resource usage 40
 - resources 40
 - restarting 20
 - starting servers with 16
- osview command (SGI) 42

P

- parameters 118
- parent process identification (PPID) 28
- password encryption
 - for libtcl*.cfg 78
 - pwdcrypt 78
- PATH environment variable 5, 32
- paths, error log 108
- performance and tuning
 - I/O-bound applications 120
 - monitoring disk usage 121

- permissions 33
 - required in environment variables 31
 - restoring of 33
- PID. See process identification
- platforms, supported ix
- platform-specific locale names 95
- port component in interfaces files 55
- port numbers and interfaces files 60
- port/object number component in interfaces files 55
- PPID. See parent process identification
- procedures, Sybase extended stored
- process identification (PID) 28
- .profile file 31
- protocol
 - component in interfaces files 54
 - Novell Protocol Suite Version 3.0.1 56
 - SPX 54
 - SPX/IPX 56
 - TCP/IP 54
- protocol component in interfaces files
 - SPX protocol entry 57
- ps command
 - Digital UNIX 41
 - HP-UX 41
 - IBM RS/6000 42
 - SGI 43
 - Sun Solaris 42
- pubs2 sample database 7
- pubs3 sample database 7
- pwdcrypt
 - location of 78
 - password encryption 78

Q

- query port backup configuration 64
- query service type 49, 54

R

- rc (run control) directory 17
- restarting Adaptive Server 20
- restore operations 113
- restoring databases

Index

- master 120
- sybsystemprocs 120
- retry_attempts component in interfaces files 54
- roman8 character set 83
- run control directory 21
- runserver file 15

S

- sar command
 - HP-UX 41
 - SGI 42
- Scandinavian dictionary sort orders 92
- scripts
 - C shell 42
 - maintenance 42
 - sample maintenance 42
 - start-up 17, 18, 21
- security. See auditing
- Server Config utility 12
- servername component in interfaces files 54
- servers
 - addresses for 48
 - naming requirements 54
- service types
 - debug 54
 - listener 49
 - master 54
 - query 49, 54
- service_type component in interfaces file 54
- setenv command 31
- setperm_all command 33
- SGI
 - gr_osview command 42
 - osview command 42
 - ps command 43
 - sar command 42
 - timex command 43
- shared memory files 29
- shells
 - Bourne xiv, 31
 - C xiv, 31
 - Korn xiv
- Shift-JIS character set 89
- shutdown command 26, 27
- slloc utility 91
- socket number 57
- Sort order
 - changing 83
- sort order
 - Adaptive Server default 83
 - default for Adaptive Server 83
- Sort orders
 - changing 84
 - letter case in 91
- sort orders 90
 - binary 91
 - changing 98
 - character sets and 90
 - databases and 90
 - definition files 90
 - dictionary 91
- spaces in interfaces files 56
- Spanish dictionary sort orders 92
- SPX network protocol 9, 54
- srt files 90
- srvbuild utility 10, 83
- starting servers
 - from UNIX command line 14
 - with operating system 16
- startserver utility 15
- start-up script 17, 18
- stripe on parameter 118
- stty settings 33
- Sun Solaris
 - hardware error messages 40
 - iostat command 42
 - monitoring systems 43
 - netstat command 42
 - network protocol 54
 - ps command 42
 - time command 42
 - timeout period 39
 - vmstat command 42
- sundiag system diagnostic tool 40
- supported platforms ix
- SYBASE environment variable 4
- \$SYBASE environment variable as default server name 59
- Sybase globalization support 81, 94, 98
- Sybase installation directory xvi

SYBASE_ASE environment variable 4
 SYBASE_FTS environment variable 5
 SYBASE_OCS environment variable 4
 SYBASE_SYSAM environment variable 5
 sybsecurity
 database 8, 124
 device 8
 sybsyntax database 131
 sybssystemdb
 purpose of 6
 sybssystemprocs database 7
 sybssystemprocs database, restoring the 120
 sysprocsdev device
 purpose of 6, 7
 system audit tables 124
 system messages, translated 82
 system procedures, storage location of 7

T

tab characters in interfaces files 56
 tape drives 113
 dumping data to 113
 examples of dumping and loading 115
 loading data to 113
 TCP/IP 37, 54
 KEEPALIVE option 37
 TCP/IP network protocol 9
 tempdb database 6
 Thai character sets 87
 time command
 Digital UNIX 41
 HP-UX 41
 IBM RS/6000 42
 Sun Solaris 42
 timex command
 SGI 43
 TLI protocol 54
 transaction log, example 116
 translated messages
 error (.loc files) 95
 system 82
 Turkish character sets 88

U

Unicode
 character conversion 85
 UnixWare
 network protocol 54
 us_english language 83
 user connections 33
 user-defined message 110
 utilities
 Server Config 12
 slloc 91

V

vmstat command
 Digital UNIX 41
 HP-UX 41
 IBM RS/6000 41
 Sun Solaris 42

W

Windows Sockets
 default socket 45, 46

X

XP Server
 default configuration 46
 XP Server, starting 13
 xp_cmdshell command 13

