SYBASE®

Installation Guide

Adaptive Server® Enterprise

12.5.4

[Linux]

Contents

About This Book

The *Installation Guide for Adaptive Server Enterprise* provides:

- An overview of the Sybase® Adaptive Server® Enterprise installation infrastructure

- Instructions for installing and upgrading Adaptive Server, and installing Backup Server™, Monitor Server, XP Server™, jConnect™ for JDBC™, Java utilities, and client products, including the Adaptive Server plug-in for Sybase Central™

- Instructions for installing optional Adaptive Server functionality, such as auditing, and sample databases, and localization information

Audience

This guide is written for Sybase System Administrators and other qualified personnel who are familiar with their system's environment, networks, disk resources, and media devices.

How to use this book

This book contains the following chapters:

- Chapter 1, "Overview" provides an over view of Adaptive Server, and describes the Enterprise Edition, Small Business Edition, and Developer's Edition of Adaptive Server. It also provicdes product descriptions and directory layout information.

- Chapter 2, "Installing Adaptive Server" describes pre-installation procedures, and how to install server components.

- Chapter 3, "Post-Installation Tasks" describes how to verify that servers are running, connect to servers, set passwords, and install sample databases.

- Chapter 4, "Installing Sybase PC-Client Products" describes how to install Adaptive Server client products like the Adaptive Server plug-in to Sybase Central.

- Chapter 5, "Upgrading Adaptive Server" describes how to upgrade an existing Adaptive Server to the current version.

- Chapter 6, "Downgrading Adaptive Server" describes how to downgrade an existing Adaptive Server to the prior version.

- Chapter 7, "Troubleshooting" provides installation error messages and possible solutions to installation problems.

- Chapter 8, "Removing Adaptive Server" describes how to uninstall Adaptive Server.

- Appendix A, "Alternative Installation Methods" describes how to install Adaptive Server in noninteractive mode.

- Appendix B, "Creating Raw Partitions" describes how to create partitions, bind partitions for raw disk I/O, and access raw devices from Adaptive Server.

- Appendix C, "Upgrading Servers with Replicated Databases" describes how to upgrade servers with replicated databases.

- Appendix D, "Post-Upgrade Tasks" describes how to restore funcitonality to Adaptive Server after upgrading, and how to restore audition, and reenable Replication Server.

- Appendix E, "Using sybsystemprocs" describes how to use the sysystemprocs database.

- Appendix F, "Backup Server tape device support" describes Backup Server tape device support.

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

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

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

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

- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5.4, the system changes added to support those features, and the changes that may affect your existing applications.

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

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

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

- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.

- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.

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

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

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

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

- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).

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

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

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

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

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

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

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

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

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

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

  - *Commands* – Transact-SQL commands.

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

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

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

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

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

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

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

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

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

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

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

Other sources of information

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

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

• The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

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

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

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **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 maintenance

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

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

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

3   Select a product.

4   Specify a time frame and click Go.

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

Conventions   The following style conventions are used in this manual:

•   In a sample window display, commands you should enter exactly as shown are given in:

        this font

•   In a sample window display, words that you should replace with the appropriate value for your installation are shown in:

        this font

- In the regular text of this document, the names of files and directories appear in this font:

  /usr/u/sybase

- The names of programs, utilities, procedures, and commands appear in this font:

  sqlupgrade

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

Table 1 shows the conventions for syntax statements in this manual.

Table 1: SQL syntax conventions

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in bold.
variable	Variables, or words that stand for values that you fill in, are in *italic*.
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
[ ]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
( )	Parentheses are to be typed as part of the command.
|	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

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

CHAPTER 1   **Overview**

This chapter introduces important concepts about the product and installation of Adaptive Server and Adaptive Server products.

Topic	Page
Product components	1
Adaptive Server editions	4
Adaptive Server specifications	6
Product descriptions and directory layout	7
PC-Client product descriptions and directory layout	8

Product components

Adaptive Server Enterprise is based on the client-server model. Each Adaptive Server user is associated with a client process that communicates with a server process via a network interface called the Tabular Data Stream (TDS) protocol. The client process may execute on one machine and communicate with a database server on the same or a different machine.

Adaptive Server Enterprise

Adaptive Server runs as an application on top of an operating system and depends solely on the services exported by the operating system to function. Adaptive Server uses operating system services for process creation and manipulation, device and file processing, and interprocess communication. The hardware that the operating system runs on is completely transparent to Adaptive Server — Adaptive Server sees only the operating system's user interfaces.

Adaptive Server has a virtual server architecture and has one or more operating system processes (engines) that service client requests. You can configure multiple processes to enhance performance on multiprocessor systems. A process is divided into two components, a DBMS component and a kernel component. The kernel component manages multiple tasks, while the DBMS component performs the database functions on behalf of the current task. Each client is associated with a task, and there are several tasks that perform specific services; for example, writing buffers or audit data to disk, and communicating with the network. The DBMS component of Adaptive Server manages the processing of SQL statements, accesses data in a database, and manages different types of server resources.

All server processes share two distinct areas of shared memory: a kernel area and a DBMS area. The kernel area contains data structures that relate to task management and operating system services, while the DBMS component contains caches of database buffers, object descriptors, and other resources used by the DBMS component of the server process.

The kernel layer of the server abstracts the operating system-specific functions so that the rest of the server has a consistent view of those functions regardless of the underlying operating system. Essentially, the kernel provides low-level functions, such as task and engine management, network and disk I/O, and low-level memory management to the rest of the server. The part of the server that processes a TDS request (the TDS engines part of the kernel layer.

The heart of Adaptive Server is the SQL/DBMS engine. The SQL/DBMS engine, also a separate task within Adaptive Server, processes data manipulation statements (DML), utility commands, which are generally data definition statements (DDL) and administrative functions, and performs stored procedure executions. The DBMS engine uses the services provided by the resource manager routines and the access methods routines.

A resource manager is a set of subroutines that manage a particular system resource, such as pages, buffers, or a particular data structure. The DBMS engine uses these subroutines to manipulate various system resources. There are also a number of tasks within the server that perform specialized tasks. The resource managers, as well as these specialized tasks, rely on the kernel layer for low-level functions.

Access methods has two primary functions: the management of data on disk and manipulation of database constructs, such as tables and rows. Actual disk I/O routines are provided to the access methods routines by the kernel disk I/O routines.

Backup Server

Like Adaptive Server, Backup server is also an operating system process. Backup Server is dedicated to backing up and restoring Adaptive Server files. Clients do not communicate directly with Backup Server. Only Adaptive Server communicates with Backup Server.

Stored procedures

Stored procedures encapsulate a series of SQL commands that the server can execute as a single command. System stored procedures (SSPs) are created when the server is installed.

The client

A client uses functions provided by the CT-Library (CTLIB) library and the Common Library (CSLIB) to establish a connection to the server, submit requests to the server, and receive results back from the server.

Figure 1-1: Adaptive Server system model
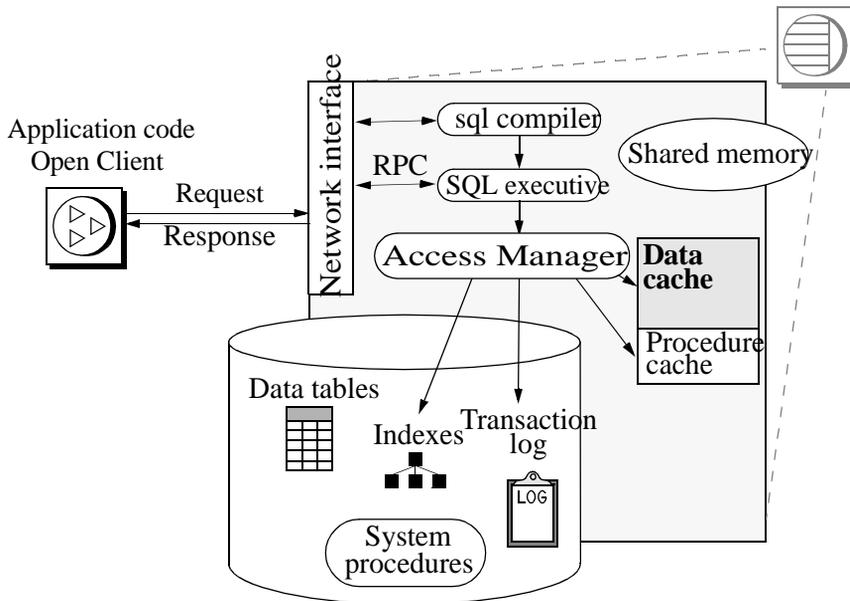
Adaptive Server editions

Adaptive Server is available in three editions:

- Enterprise Edition – a full-featured server that can run all optional features.

- Small Business Edition – includes the features required by most small businesses, but excludes some of the more advanced features.

- Developer's Edition – is created for you to design and build applications for Adaptive Server in a development environment.

Sybase also offers a series of options. Some of these options are intended for an enterprise-level site, and are not available in the Small Business Edition. Table 1-1 describes the availability of each option. For a full description of the option contents, see your sales representative.

Table 1-1: Optional packages available for Adaptive Server

Available in the following editions:

Option	Enterprise Edition	Small Business Edition	Developer's Edition	Description
Security and Directory Services option	Optional	Optional	Included	Lightweight directory services and network-based authentication and encryption using DCE and CyberSafe.
XML Management option	Optional	Optional	Included	Semi-structured and unstructured-data handling.
Content Management option	Optional	Optional	Included	Semi-structured and unstructured-data management.
e-business option	Optional	Not available	Included	Data-storage, access, replication, and security via an e-business platform.
High Availability	Optional	Not available	Included	Systems continuously available.
Distributed Transaction Management (DTM)	Optional	Not available	Included	
Enhanced Full Text Search (EFTS)	Optional	Optional	Not available	
SQL Expert	Optional	Optional	Not available	

	Available in the following editions:			
Option	**Enterprise Edition**	**Small Business Edition**	**Developer's Edition**	**Description**
BMC DBXray	Optional	Optional	Not available	
Disaster Recovery option	Optional	Not available	Not available	Provides a disaster recovery package for Adaptive Server users.
Web services	Optional	Optional	Included	Provides access to Adaptive Server through Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL).

For more information, see "SySAM Configuration" in the *Configuration Guide*.

Use license_enabled to determine which edition of Adaptive Server you are running. Enter:

```
select license_enabled("server_edition")
```

Where *server_edition* is:

• ase_server for the Enterprise Edition

• ase_sbe for the Small Business Edition

• ase_dev for the Developer's Edition

Adaptive Server returns a value of "1" for the version that is enabled. See the *Reference Manual* for more information about the license_enabled function.

Enterprise Edition

The Enterprise Edition includes a full-featured Adaptive Server. In addition to the base server, you can purchase such options as High Availability, DTM, Java, and so on. Your licensing agreement describes the features licensed for your site.

The options available for Adaptive Server Enterprise are listed in Table 1-1.

Small Business Edition

The Small Business Edition includes features that meet the needs of small businesses. This edition of Adaptive Server includes most of the functionality of the Enterprise Edition, and allows for the following Adaptive Server configuration:

- Number of engines – maximum of 4.

- Number of connections – maximum of 256.

The options available for Adaptive Server Enterprise are listed in Table 1-1.

Developer's Edition

The Developer's Edition is a competitively-priced version of Adaptive Server that developers can use to create applications in their development environment. The Developer's Edition includes a full set of Adaptive Server Enterprise features and options (that is, DTM, High Availability, Java, and so on), and allows for the following configuration:

- Number of engines – 1

- Number of connections – 25

The options available for Adaptive Server Enterprise are listed in Table 1-1.

Adaptive Server specifications

Table 1-2 and Table 1-3 provides Adaptive Server specifications for Linux.

Table 1-2: Adaptive Server specifications on Linux

Hardware		
Processor	32-bit Intel Pentium II	
Minimum RAM required for Adaptive Server	128MB	With default stack size, packet size, and user log cache size
Minimum RAM per additional user	Approx. 140KB	
Default user stack size	80KB	

Table 1-3: Adaptive Servr specification on Linux 64-bit

Hardware		
Processor	64-bit Intel Itanium2	With default stack size, packet size, and user log cache size
Minimum RAM required for Adaptive Server	128 MB	
Minimum RAM per additional user	Approximately 240 K	
Default user stack size	92K	

Note Adaptive Server 12.5.4 supports Pentium and higher chip sets.

Note For more information about Adaptive Server specifications see Chapter 1 "Introduction" in the *Configuration Guide* for your platform.

Product descriptions and directory layout

The Adaptive Server installation installs a wide array of software products into the *SYBASE* directory. Table 1-4 provides a brief description of the server products installed, and the top-level directory structure.

Table 1-4: Product descriptions

Product	Description
Adaptive Server	Installed into the *ASE-12_5* directory.
	• Adaptive Server – the database server.
	• Backup Server – an Open Server-based application that manages all database backup (dump) and restore (load) operations.
	• Monitor Server – an Open Server-based application that obtains performance information on Adaptive Server and makes that information available to Monitor Server client applications.
	• XP Server– An Open Server application that manages and executes extended stored procedures (ESPs) from within Adaptive Server.
Job Scheduler	Installed into the *JS-12_5* directory. Provides a job scheduler for Adaptive Server.

Product	Description
Open Client	Installed into the *OCS-12_5* directory. • Open Client libraries needed for development and execution of Open Client applications. • Monitor Client Library. • XA Libraries.
Sybase Central 4.3	Install into the *Sybase Central v43* directory. This is a Java-based common framework for managing Adaptive Server.
Adaptive Server Plug-in	Installed into *ASEP* directory. This plug-in contains the management software for managing Adaptive Server and its associated servers.
Language Modules 12.5.4	Installed into the *locales* directory. Provides system messages and date/time formats.
Character Sets	Installed into the *charsets* directory. Provides character sets available for use with Adaptive Server.
Collation Sequences	Installed into the *collate* directory. Provides the collation sequences available for use with Adaptive Server.
jConnect	Installed into the *jConnect-5_5* or *jConnect6_0* directory. Provides a Java Database Connectivity (jdbc) driver for Adaptive Server.
ASE Replicator	Installed into the *RPL-12_5* directory. Provides lightweight replication for those companies who do not need a full-featured replication product such as Sybase Replication Server.
SNMP Subagent	Installed into the *SNMP-1_0* directory. A Java-based SNMP subagent that supports the AgentX protocol. See the README file in the SNMP-1_0 directory for more information.
SQL Remote	Installed into the *SQLRemote* directory. Enables two-way replication between Adaptive Server and multiple computer databases through e-mail or dial-up connections.
Sybase Software Asset Management (SySAM)	Installed into the *SYSAM-1_0* directory. Provides asset management of Adaptive Servers and optional features.
Enhanced Full-Text Search	Installed into the *EFTS-12_5_2* directory. An optional product that provides for faster searching of text data in the Adaptive Server. (Not available on SGI.)
Web Services	Installed into the *WS-12_5* directory. An optional product that provides access to Adaptive Server using HTTP/SOAP and WSDL.

PC-Client product descriptions and directory layout

The following products are installed as Adaptive Server PC-Client products. In addition to these products, there are several separately installed products listed in the following sections.

Table 1-5: Products and platforms

Product	Description
Open Client	Installed into the *OCS-12_5* directory. • Open Client libraries needed for development and execution of Open Client applications. • ODBC – driver used to connect to Adaptive Server from ODBC based applications. • OLEDB • Monitor Client libraries • XA libraries
Sybase Central	Install Sybase Central v43 directory. A Java-based common framework for managing Adaptive Server.
Adaptive Server Plug-in	Installed into *ASEP* directory. This plug-in contains the management software for managing Adaptive Server and its associated servers.
Language Modules	Installed into the *locales* directory. Provides system messages and date/time formats.
Character Sets	Installed into the *charsets* directory. Provides charsets available for use with Adaptive Server.
jConnect	Installed into the *jConnect-5_5* directory. Provides a Java Database Connectivity (jdbc) driver for Adaptive Server.

Separately installable PC-Client products

The PC-Client CD also contains the following products, which are separately installable:

• InfoMaker – a tool for personal data access, management, and reporting, used by both developers and end users. InfoMaker® allows you to create presentation-quality reports and powerful queries without the complexities of programming. It complements client/server business applications and development tools as well as desktop productivity suites.

• PowerDesigner Physical Architect – a tool for data modeling, including database design, generation, maintenance, reverse engineering, and documentation for database architects.

• PowerTransfer – a plug-in for PowerDesigner that transfers table data from any supported database to Adaptive Server.

CHAPTER 2   **Installing Adaptive Server**

This chapter provides instructions for installing a new Adaptive Server, Backup Server, and Monitor Server.

Overview

The installation process copies all of the files from the installation media to the hard drive and then configures a base Adaptive Server. At the end of the installation the Adaptive Server and associated software is ready to be customized for the user's environment (for example, creating devices, databases and tables, and loading data).

The basic process for installing Adaptive Server is:

1   Perform the pre-installation tasks as described in "Pre-installation tasks" on page 14.

2   Perform the installation using any of the methods described in "Installation methods" on page 15.

3   Perform the post-installation tasks as described in Chapter 3, "Post-Installation Tasks."

•   You can specify only one engine in the max online engines configuration parameter.

•   You can configure up to 2GB of max memory.

•   The total disk space is limited to 5GB.

System requirements

The following tables show the required version-level, RAM, and network-protocol requirements for Adaptive Server. See the release bulletin for the latest information on required system patches.

Note For the latest certification information, see the Technote at http://www.sybase.com/detail/1,6904,1024194,00.htm.

Table 2-1: System requirements for Linux 32-bit

Hardware	Operating system	Network protocols
Linux with Intel architecture, Pentium II, 230 MHz	RedHat Enterprise Linux 2.1: • kernel-2.4.9-e.3 or higher • glibc-2.2.4-26or higher RedHat Enterprise Linux 3.0: • kernel 2.4.21-4.EL or higher • glibc-2.3.2-95.3 or higher • compat-libstdc++-7.3-2.123 or higher SuSE Linux Enterprise Server SLES 8.0 / United Linux (UL) 1.0: • kernel 2.4.18-e.37 or higher • glibc 2.2.5-165 or higher • compat-2002.8.15-29 or higher	TCP/IP sockets

Table 2-2: System requirements for Linux 64-bit

Hardware	Operating system	Network protocols
Linux with Intel architecture, Itanium 2, 900 MHz	RedHat Enterprise Linux 2.1: • kernel 2.4.18-e.37 or higher • glibc-2.2.4-31.7 or higher FedHat Enterprise Linux 3.0: • kernel 2.4.21-4.EL or higher • glibc-2.3.2-95.3 or higher • compat-libstdc++-7.3-2.96.122 or higher SGI Altix Adanced Linux Environment 2.1.1 with ProPack 2.3: • kernel-2.4.21-sgi230r7 or higher • glibc-2.2.4-32.3sgi230r1 or higher	TCP/IP sockets

Table 2-3: Disk space requirements for Linux 32 -bit platforms

Product	Disk space requirements
Adaptive Server typical install	500MB
EJB Server (option)	n/a
Enhanced Full-Text Search (optional)	165MB
Default databases created during install	150MB
Total	815MB

Table 2-4: Disk space requirements for Linux 64-bit platforms

Product	Disk space requirements
Adaptive Server Enterprise typical install	550MB
EJB Server (option)	n/a
Enhanced Full-text Search (optional)	n/a
Web Services (optional)	n/a
Default databases created during install	150 MB
Total	700 MB

Note Set character or block devices for each disk partition, one or the other should be used, not both.

Pre-installation tasks

Before installing Adaptive Server:

1   Read the release bulletins for the latest information on the products (Adaptive Server, Monitor Server, and so on) that you are installing. See "Special Installation Instructions" in the release bulletin.

2   Install operating system patches, if required.

    Recommended operating system patches are:

    For information about operating system requirements see "System requirements" on page 12.

3   If you are installing multiple server, you should review the SySAM procedures and plan your client/server configuration using the *Configuration Guide* for your platform.

4   Create a "sybase" account on your system to perform all installation tasks.

    The "sybase" user must have permission privileges from the top (or root) of the disk partition or operating system directory down to the specific physical device or operating system file.

5   Log in to the machine as the "sybase" user.

    Maintain consistent ownership and privileges for all files and directories. A single user—the Sybase System Administrator with read, write, and execute permissions—should perform all installation, upgrade, and setup tasks.

6   Decide where the Adaptive Server software will be installed. Make sure that there is sufficient available disk space. There cannot be any spaces in the path name of the directory.

7   Verify that the operating system meets the version-level, RAM, and network protocol requirements for your platform.

8   Verify that your network software is configured.

    Sybase software uses network software even if Adaptive Server and Sybase client applications are installed on a machine that is not connected to a network.

    If you are having connection problems, or to verify your network configuration, ping the host.

9   The operating system shared memory default, for most Linux releases is 32MB. The minimum required by Adaptive Server is 64MB for default Server with 2K pages. Higher value is required if you plan to increase Adaptive Server's total memory.

Use the sysctl(8) method to check and adjust the operating system shared memory parameter.

To check the current shared memory size, enter:

```
# /sbin/sysctl kernel.shmmax
```

To adjust the shared memory size:

```
# /sbin/sysctl -w kernel.shmmax=nnn
```

where *nnn* is the new size in bytes (at least 64MB which is 67108864 bytes).

To guarantee that this value is applied every time the system is started, add the above line to your */etc/rc.d/rc.local* file. On SuSE systems, the file that needs to be edited to apply this change after each start is */etc/init.d/boot.local*.

Installing server components

Follow the instructions in this section to install Adaptive Server, Backup Server, and Monitor Server.

Be sure that you are logged in as "sybase", and that you have performed the "Pre-installation tasks" on page 14.

Installation methods

The following instructions are for an installation using the InstallShield in GUI mode. You can also perform the installation in a non-GUI (text-based) mode or a completely silent installation with no interaction. Or, you can perform the installation, then configure Adaptive Server separately using either the GUI-based configuration tools or silently with the use of a resource file. For information on alternate methods of installation see Appendix A, "Alternative Installation Methods."

Installing components with InstallShield in GUI mode

> **Note** You should be logged in as "sybase" when you are performing installation tasks. Do not run InstallShield as "root".

InstallShield creates the target directory (if necessary) and installs all the selected components into that directory.

At the end of the installation, you can verify the product installation. You may need to perform additional configuration procedures before you can use some products.

As part of the installation, InstallShield sets most of the environment variables needed for Adaptive Server products. However, you must source other environment variables by running the *SYBASE.csh* or *SYBASE.sh* script file after exiting InstallShield.

To install server components:

1   Insert the Adaptive Server CD into the CD.

    Change to the CD directory and start the installer by entering:

        ./setup

    The welcome window displays

2   When the license selection window displays, select the most appropriate location from the drop-down list, read the license terms, and then click "I agree..." to proceed. Click Next.

    Not every country is listed in the drop-down list. If the country you are located in is not listed, select the most appropriate area ('Americas & Asia Pacific', 'Europe, Middle-East, and Africa', or 'Any Other Locations').

3   The Install directory window allows you to select a directory for the installation by clicking Browse. You can accept the default of:

        /opt/sybase

    by clicking Next.

You can also enter a directory for installation.

> **Note** When you browse "..." in the Destination panel, a directory chooser dialog displays. In this dialog, you must navigate to one directory above the "to-be-chosen" directory, and enter or choose a directory name in the field, before clicking OK.

4   You can choose from three types of installations in the Install Type window:

 •   Typical

 •   Full

 •   Custom

> **Note** Enhanced Full-Text Search, EJB Server, Language Modules other than English, and Web Services are not installed as part of a typical install. To install them, select Full or Custom install. Or, you can install them later using the custom option.

5   If you select Custom, the Products and Features Selection window displays. Indicate the products you want to install by checking the box.

> **Note** Some features are dependent on other features; therefore, you cannot deselect some items without first deselecting others.

After you have made your selection, click Next. Before proceeding to the next window, InstallShield verifies the selections, and checks for dependencies and available disk space.

6   The Product Selection Summary window displays the selections that you have made.

Verify that you have selected the correct type of installation, and that you have enough disk space to complete the process. Click Next to proceed.

7   The Install Progress window shows the progress of the installation.

8   The Install Status window displays the result of the installation process.

Click Next to configure the software.

9   The Configure New Servers window asks if you want to configure new servers, depending upon the products you installed. If you choose to configure the new servers, the Custom Configure New Server Options window displays next. Select the servers you want to configure. If you do not configure the new servers at this point, deselect the item and click Next.

> **Note** As part of the installation, you should have a local server name for the server. If you do not, sp_sysmon "cachewizard" does not work.

10 The Custom Configure New Server Options window allows you to select servers to configure. If you do not select any servers in this window, InstallShield configures all servers using default value.

If you choose the default values, InstallShield chooses everything including the server name, port number, and master device location. InstallShield does this for the System Procedure device. It also installs Backup Server, Monitor Server, and the XP Server, choosing the name, port number, and error log.

If you accept the defaults, the server names are:

- Adaptive Server – *<host name>*
- Backup Server – *<host name>*_BS
- Monitor Server – *<host name>*_MS
- XP Server – *<host name>*_XP
- Job Scheduler – *<hostname>*_JS
- Web Services – *<hostname>* _WS

If you configure the servers with default values, the Configure Server Attributes Summary window displays next.

If you select to configure the server with custom values, the Custom Configure Input window displays next.

11 The Custom Configure Input window prompts you for the custom Adaptive Server configuration information.

- Server name
- Port numbers
- Page size
- Error log

- Master device

  > **Note** When configuring the master device, you should allow an extra 8K for the config block.

- Master device size (MB)

- Master database size (MB)

- System procedure device

- System procedure device size (MB)

- System procedure database size (MB)

Once you have customized the Adaptive Server configuration, select Next to record the input fields.

12 The Custom Configure Backup Server Input window prompts you for the Backup Server information.

Once you have entered the appropriate information, click Next to record the information.

13 The Custom Configure Monitor Server Input window prompts you to enter the Monitor Server configuration information.

Once you have entered the appropriate information, click Next to record the information.

14 The Custom Configure XP Server Input window prompts you to enter the XP Server configuration information.

Once you have entered the appropriate information, click Next to record the information.

15 The Configure Server Attributes Summary window displays a summary of the values that are used to configure the servers. These values are either the default or custom values. After verifying the information, click Next, and InstallShield proceeds with the server configuration.

16 The Configure Server Progress window displays the progress of the server configuration.

17 The Logout or Reboot window displays only when required. It reminds you to either log out or restart your machine to complete the installation process.

18 The Installation Complete window is the last window. Sybase strongly recommends that you check at http://www.sybase.com/downloads for software updates.

If you are installing the Developer's Edition, the installation is complete. If you are installing the Small Business Edition or Enterprise Edition, proceed to the next section to enter the SySAM certificates to enable the server and any optional features purchased.

If you encounter any errors, see the *Troubleshooting Guide*.

Sybase Software Asset Manager (SySAM)

If you install any components without the appropriate license information, only Adaptive Server, without licensed features, is enabled.

Warning! The following SySAM license manager instructions are for the installation of Adaptive Server on the primary license host.

1 Source *SYBASE.csh*.

2 Go to *$SYBASE/SYSAM-1_0/bin* and run lmgr. The lmgr program prompts, "Do you have any Sybase Software Asset Management Certificates to register?"

Click Yes.

3 Enter information from the Sybase License Certificate for each Adaptive Server feature you have purchased. Entries are case sensitive.

- Order Number – enter your Sybase order number.

- Feature Name – enter the name of the Adaptive Server feature. Valid Adaptive Server feature names include ASE_SERVER and ASE_EFTS.

- Feature Count – enter your license count number.

- Software Version – enter the Adaptive Server software version.

> **Warning!** You must enter the license information exactly as it appears on the certificate.
>
> The license certificates for the base server and some of the options indicate version 12.0, although the software is at version 12.5. The server and all features are considered to belong to the same "Version 12 Product Family."

- Authorization Code – enter the license key for the purchased feature.

The lmgr program records the information for the current feature in the license file and prompts you to enter information for an additional feature.

4   Click More.. . if you have purchased additional licensed features. The installer records the information for the current feature in the license file and prompts you to enter information for an additional feature.

5   After entering the license key information the lmgr program automatically stops and then restarts the license daemon.

SySAM LicenseManager scripts

SySAM LicenseManager runs in four modes:

- GUI mode – this is default, corresponding to the parameter -G.

- Interactive Console mode – where user is asked information, interactively. -I.

- Console mode – this is same as Interactive mode, but takes only one entry and quits -C.

- Silent mode – this best suits silent installation, when updating multiple machines at a single time which use networked SYSAM. This corresponds to the parameter -S.

Use the LicenseManager with the following options and arguments:

Options:

- -H – displays help files

- -V – prints version and copyright

- -G – run application in GUI mode

- -I – run application in Interactive Console Mode

- -S – run application in Silent Mode

- -C – run application in Console Mode

Arguments are only applicable with the -C or -S options:

- sybase=sybase directory

- host=hostname

- port=port number

- feature=feature name

- count=feature count

- order=order number

- version=software version

- code=authorization code

Adaptive Server and related products have been successfully installed, and minimally configured for use. See "Installing sample databases" on page 29, to begin experimenting with your servers, or see the *System Administration Guide* at http://www.sybase.com/support/manuals for more advanced topics.

Adaptive Server

You have a running server, as well as various system databases and system tables. Some references that may help you get started using your new Adaptive Server include:

- "Initializing Database Devices" and "Creating User Databases" in the *System Administration Guide* – information about creating an Adaptive Server user database and its devices

- *Transact-SQL User's Guide* – learn how to write queries

Backup Server

For information on developing a backup and recovery plan and backing up and restoring user databases, see the *System Administration Guide*.

Monitor Server

Monitor Server requires some additional configuration after installation. See the *Monitor Server User's Guide*.

XP Server

For information on using extended stored procedures, see the *Transact-SQL User's Guide*.

Installing version 12.5.4 over an existing version

Determining Adaptive Server version

Verify that your current Adaptive Server installation is at version 12.5.x by running the following query from isql:

```
1> select @@version
2> go
```

If your server is not running, you can get a version string by running:

```
$SYBASE/$SYBASE_ASE/bin/dataserver -v
```

If this shows that the Adaptive Server is not at version 12.5.x, you must upgrade Adaptive Server. See Chapter 5, "Upgrading Adaptive Server."

If your server is at a 12.5.x version level, you can begin installing Adaptive Server 12.5.4.

Backing up Adaptive Server

Installing Adaptive Server version 12.5.4 overwrites the current Adaptive Server software. Before installing, verify that your databases are error free and that your *$SYBASE* directory is backed up.

To ensure that your database is error free, run dbcc checkdb, dbcc checkcatalog and dbcc checkstorage prior to loading any new Adaptive Server binaries, including the master database. If the dbcc commands reveal problems, check the *Error Messages and Troubleshooting Guide* for actions necessary to fix the problem. If the error is not listed in the manual, call Sybase Technical Support.

After you have verified that your database is error free, back up your *$SYBASE* directory in case you need to roll back to the original version of the software.

Installing the Adaptive Server version 12.5.4 binary overlay

1   Read "Special Installation Instructions" in the release bulletin for the latest information that may affect specific 12.5.4 configurations.

This document may also contain last minute installation information for the 12.5.4 installation.

2   After backing up your databases, shut down Adaptive Server and back up your *SYBASE* directory.

3   Installing Adaptive Server 12.5.4 overwrites current Adaptive Server software. Prior to installing 12.5.4, make sure that your databases are error-free as described in "Backing up Adaptive Server" on page 23 and that your *SYBASE* directory is backed up.

4   Use InstallShield to load the new software from the CD.

      For Windows installations, InstallShield starts automatically.

      For UNIX installations, change to the CD drive and enter `./setup`.

5   Install Adaptive Server 12.5.4 over the *$SYBASE* installation path.

      Note Once files are loaded into your *$SYBASE* directory, InstallShield asks if you want to configure the new installed server. Deselect configuring the new server, and click continue to finish the installation.

6   Restart Adaptive Server.

7   Run select @@version. The server should now be at version 12.5.4.

8   In Adaptive Server version 12.5.4, several changes have been made to the system stored procedures and many new error messages have been added. You must perform the post-installation tasks to make these changes available.

      installmaster and *instmsgs.ebf* should be run, in addition to following the instructions in TechNote 1013610, Post-installation Scripts and Tasks for Adaptive Server Enterprise 12.5 EBFs at http://www.sybase.com/detail?id=1013610 for instructions on post-installation tasks you may need to perform.

Backward compatibility with previous Adaptive Server 12.5.x versions

Before using the new Adaptive Server 12.5.4 features, read "Adaptive Server 12.5.4 Backward Compatibility," the Sybase Web site at http://www.sybase.com/support/techdocs.

You can downgrade to earlier 12.5.x versions of Adaptive Server, but to do so you must take the precautions described in the technical note before using any of the new 12.5.4 features.

XML Services in Adaptive Server version 12.5.4

If you are planning on using XML in Adaptive Server version 12.5.4, there are installation issues you must take into consideration. See "Appendix E: Migrating Between the Java-based XQL Processor and the Native XML Processor" in *XML Services in Adaptive Server Enterprise.*

Changing the default language

Once the installation for Adaptive Server Enterprise version 12.5.4 is complete, the System Administrator must reinstall the langinstall utility, in *$SYBASE/ASE-12_5/bin/langinstall*, to update the localized messages.

CHAPTER 3   **Post-Installation Tasks**

After the installation process, you have a minimally configured Adaptive Server. This chapter describes post-installation tasks to administer Adaptive Server and install stored procedures and sample databases. Sybase recommends that all post-installation tasks be carried out as "sybase" for the user. For more information, see *Configuration Guide* for your platform.

Topic	Page
Verifying that servers are running	27
Verifying that you can connect to servers	27
Connecting to Adaptive Server via Sybase Central	28
Setting the System Administrator password	29
Installing sample databases	29

Verifying that servers are running

To determine whether the servers you installed are running, enter:

```
$SYBASE/$SYBASE_ASE/install/showserver
```

showserver shows all Adaptive Server-related processes that are on the system.

Verifying that you can connect to servers

Use isql to perform a quick test. Source *SYBASE.csh*. Then, use isql to connect to servers:

1   At the command prompt, enter:

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

where *server_name* is the Adaptive Server name.

The command prompt is displayed if the login is successful.

2   To display the Adaptive Server version number, enter:

```
1> select @@version
2> go
```

Adaptive Server's version number is displayed. The output should show Adaptive Server at version 12.5.4.

If you encounter errors, see the *Troubleshooting Guide*.

Connecting to Adaptive Server via Sybase Central

Sybase Central provides a graphical user interface where you can perform Adaptive Server administrative tasks. The Sybase Central interface provides a hierarchical list of servers in the left pane and a details list of the selected server in the right pane. To select a server, click on it in the left pane.

1   From a UNIX command line, enter:

```
source $SYBASE/SYBASE.csh
```

2   Then enter:

```
$SYBASE/ASEP/bin/aseplugin
```

3   Select Tools | Connect, then select the server to which you want to connect from the drop-down list.

4   In the Login window, enter, the System Administrator's user name and password.

You must log in as the System Administrator to perform administrative tasks.

5   Click OK.

Warning! The first time you log in to Adaptive Server, use the default "sa" user name and leave the password blank. After you log in for the first time, change the System Administrator's password. See "Setting the System Administrator password" on page 29.

The Adaptive Server plug-in to Sybase Central does not display all servers listed in the *interfaces* file. Instead, Sybase Central lists only those servers that you connected to earlier, or those servers that are started as Windows services.

To access a new server for the first time, select Tools | Connect to select a server listed in the *interfaces* file.

Setting the System Administrator password

A user account called "sa" is created for the Sybase System Administrator when you install the Sybase software. A user logged in as "sa" can use any database on Adaptive Server, including master, with full privileges.

Immediately after a new installation, there is no password on the "sa" account. The initial default value for the password is NULL. In a production environment, the Sybase System Administrator should always use a non-default password.

The Sybase System Administrator should log in to the new Adaptive Server as "sa" and set a password using sp_password:

```
$SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Sserver_name
1> sp_password null, new_password
2> go
```

where null is the default password and *new_password* is the password that you are assigning to the "sa" account.

For greatest security, Sybase recommends setting the password to at have at least six characters, with a combination of letters and numbers.

Installing sample databases

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

- "Running the database scripts" on page 31
- "Installing the interpubs database" on page 32

• "Installing the jpubs database" on page 32

The sample databases contain information about a fictitious business. You can use this information to learn about the Sybase products, without affecting essential data while learning.

Table 3-1 lists the scripts that you can use to install the sample databases.

Table 3-1: Sample database scripts

Script	Description
installpubs2	Installs the pubs2 sample database.
	This database contains data that represents a publishing operation. Use this database to test your server connections and to learn Transact-SQL. Most of the examples in the Adaptive Server documentation query the pubs2 database.
installpubs3	Installs the pubs3 sample database.
	This updated version of pubs2 uses referential integrity. In addition, its tables are slightly different than the tables used in pubs2. Where noted, the Adaptive Server documentation uses the pubs3 database in its examples.
installpix2	Installs the image data that is used with the pubs2 database.
	Note The master device size should be at least 30MB to install the full pubs2 database, including the image data. Run the *installpix2* script after you run *installpubs2*.

Default devices for sample databases

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

If you have not used sp_diskdefault to change the status of the master device or to specify another default device, the scripts install the sample databases on the master device. Sybase does not recommend this configuration because it uses valuable space that is best used for system tables. Each sample database requires 3MB on a 2K server, and multiples of 3MB on a 4K, 6K, 8K, and 12K server on your database device.

To avoid installing sample databases on the master device, either:

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

• Modify each sample database installation script to specify a different device.

❖   **Running the database scripts**

1   Start Adaptive Server.

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

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

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

From the Adaptive Server scripts directory (*$SYBASE/$SYBASE_ASE/scripts*), use isql to log in to Adaptive Server and run the script:

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

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

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

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

5   To install the image data associated with pubs2 (pubs3 does not use image data), run:

```
isql -Usa -Ppassword -Sservername \
-i $SYBASE/$SYBASE_ASE/scripts/installpix2
```

Note The image data requires a fair amount of space—there are six pictures, two each in the PICT, TIFF, and Sun raster file formats. Run *installpix2* script only to use or test the image datatype. Sybase does not supply any tools for displaying image data. You must use appropriate window graphics tools to display the images after you have extracted them from the database.

For more information about running these scripts, see *Configuring Adaptive Server Enterprise*.

interpubs database

interpubs is a database similar to pubs2 that contains French and German data. This data contains 8-bit characters and is available for use at Adaptive Server installations using the ISO 8859-1 (iso_1), ISO 8859-15 (iso15), Roman8, or Roman9 (for HP-UX) character set. To display the French and German data correctly, you must set up your terminal to display 8-bit characters.

❖ **Installing the *interpubs* database**

1 Be sure iso_1, iso_15, Roman8, Roman 9, or UTF-8 is installed as the default character set or as an additional character set.

2 Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device where you will be storing the interpubs database. You will need to provide this information later.

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

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

5 Execute the script, using the -J flag to ensure that the database is installed with the correct character set:

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

For more information on the -J option in isql, see the *Utility Guide*.

jpubs database

If you installed the Japanese Language Module with your Adaptive Server, the scripts file contains the *installjpubs* script for installing the jpubs database. jpubs is a database similar to pubs2 that contains Japanese data. *installjpubs* uses either the EUC-JIS (eucjis), UTF-8 (utf8), or the Shift-JIS (sjis) character sets.

❖ **Installing the *jpubs* database**

1 Set your terminal to display 8-bit characters.

2 Verify that either the EUC-JIS, Shift-JIS, or the UTF-8 character set is installed as Adaptive Server's default character set or as an additional character set.

3 Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device where you will be storing the jpubs database. You will need to provide this information later.

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

5 Use a text editor to edit the script, if necessary, to specify a default device other than the master device, or use sp_diskdefault. See "Default devices for sample databases" on page 30.

6 Execute the *installjpubs* script, using the -J flag to ensure that the database is installed with the correct character set:

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

or:

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

For more information on the -J option in isql, see the *Utility Guide*.

Maintaining the sample databases

> **Note** Sybase recommends that you remove the guest user from user databases in production systems.

The sample databases contain a guest user that allows access to the database by any authorized Adaptive Server user. The guest user has a wide range of privileges, including permissions to select, insert, update, and delete user tables. For more information about the guest user and a list of guest permissions, see the *System Administration Guide*.

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

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

Installing Sybase PC-Client Products

Your UNIX-based Adaptive Server also includes PC-client products that you install on a PC and connect, via the *interfaces* file, with your Adaptive Server.

This chapter describes how to install client products on Windows Operating Systems.

Overview

Clients, such as Sybase Central and Open Client, are used to access Adaptive Server. See the *Open Client/Server Configuration Guide for Desktop Platforms* for more information.

Sybase Central and ASE plug-in

Sybase Central allows you to connect to Adaptive Server via client connections. You can use Sybase Central to connect to remote servers, local servers (installed on the same machine), and multiple servers. See "Configuring network connections for client products" on page 39. Before you use the plug-in, restart your computer to implement changes made to the PATH and CLASSPATH environment variables during installation.

Open Client runtime

The Sybase PC-Client CD includes Open Client software for Windows. Several products, such as the Adaptive Server plug-in and SQL Advantage, require Open Client runtime. Monitor Client Library requires Open Client/C with the SDK option.

ODBC/OLEDB   Some applications do not connect to Adaptive Server directly through the Open Client software but through the Open Database Connectivity (ODBC) driver, or the jConnect driver instead. For example, PowerDesigner connects through the ODBC driver.

See *Configuration Guide* for more information.

ODBC and OLEDB require the Microsoft Data Access Component (MDAC). You can download it from the Microsoft Web site at http://www.microsoft.com/data.

Before you install the PC-client products

Before starting the installation process for PC-client products, be sure that you have at least an extra 5MB of disk space, a *\temp* directory and TEMP environment variable.

The installation program uses the extra space and directory to write files temporarily during the installation. The installation program frees this space after installation is complete.

Installing PC-client products

The PC-Client CD contains several products that are each packaged with their own installer. When you insert the PC-Client CD, a menu program launches automatically. The menu program presents the list of products that can be installed from the CD and allows you to install them one at a time. The major products on the CD are described in Chapter 1, "Overview."

There are six selections on the menu:

*   View *Readme.txt*
*   Install PC-client components 12.5.4
*   Install PowerDesigner 12.0
*   Install PowerTransfer 10.0
*   View our Web site
*   Exit

Before installing any products, review the *readme.txt* file, which contains a brief description of each of the products, lists dependencies, and contains any last-minute instructions or changes.

In general, some portion of the PC-client components are queried by the other products so it is best to install the PC-client components first.

Table 4-1: PC-client system requirements

Hardware	Operating system
Windows 2000 Professional, Server, Advanced Server, and Data Center Editions	Service Pack 3
Windows 2003	Service Pack 1
Windows XP Professional (32-bit)	None; Sybase recommends using latest service pack from Microsoft for best security protection

❖ **Installing client products on Windows**

1   Verify that your computer has sufficient RAM for each product.

2   If you are unloading components on Windows, log in using an account with Windows administrator privileges.

3   Close any open applications or utilities to free memory and system resources.

4   Insert the PC-Client CD in to the drive. The menu program should start automatically.

   Alternatively, you can start the menu program from the Windows Start menu, select Start | Run, and enter the following where *X* is your CD drive:

   ```
   X:\autorun.exe
   ```

5   Select Install PC Client Components 12.5.4 from the menu.

6   When InstallShield starts, the welcome window displays, click next.

7   The Sybase license agreement is displayed, select agreement, click next.

8   The window displays the destination directory, enter the directory path and click Next.

9   Select the type of installation to be performed.

   •   Typical Install – installs the default components a user needs.

   •   Full Install – installs every component on the CD.

- • Customized Install – allows you to select the components to install. Certain components are automatically installed if they are required to run other selected components.

10 If you select Customized Install, the next window is the Component Selection window, which allows you to specify which components to install.

Components that would be installed in a typical installation appear with a check in the check box to the left of the product name. You may select or deselect components from this list.

11 Click Next.

The Summary window displays every component to be installed by the InstallShield Installer, the required disk space, and the available disk space.

If the target directory does not have enough free space, the available space appears in red. You must click Previous to return to the previous window and select other products or directory. You can also select Cancel and quit the installer.

12 Click Next.

Note Optionally, you can select Save on the Summary window to save all the installation information into a *cmdfile* to proceed with the installation in a noninteractive, silent install. See Appendix A, "Alternative Installation Methods."

13 If the target directory does not exist, InstallShield prompts: "OK to create directory?" Click Yes.

InstallShield installs the components and displays a progress indicator.

To install InfoMaker 10.5, insert CD 2 use Start | Run and browse the CD for the *setup.exe* file.

To configure client network connections to Adaptive Server, see the *Configuration Guide* for your platform.

Installing PowerDesigner, PowerTransfer, and InfoMaker

Install PowerDesigner by selecting Install PowerDesigner 12.0 from the menu program. Install PowerTransfer by selecting Install PowerTransfer 10.0. Insert CD 2 to install InfoMaker.

You must install PowerDesigner and the PC-Client components before installing PowerTransfer. You must also have the Microsoft .NET framework installed.

Configuring network connections for client products

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

For Sybase products to interact with one another, each product needs to know where the others reside on the network. This information is stored in the *interfaces* file (*sql.ini* on Windows) or in an LDAP server.

Configuring *libtcl.cfg* for LDAP

Use the *libtcl.cfg* files to specify an LDAP (Lightweight Directory Access Protocol) server name, port number, DIT base, user name, and password to connection to an LDAP server.

The default *libtcl.cfg* file is located in:

%SYBASE%\%SYBASE_OCS%\ini

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

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

where *ldapurl* is defined as:

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

When an LDAP server is specified in the *libtcl.cfg* file, the server information is accessible only from the LDAP server. Adaptive Server ignores the *interfaces* file. Open Client/Open Server applications that use the -I option at start-up override the *libtcl.cfg* file and use the *interfaces* file.

To use a directory service, you must:

1   Add the location of the LDAP libraries to the PATH environment variable for your platform.

2   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 the *Configuration Guide* for supported LDAP URL values.

Warning! You must enter the LDAP URL on a single line.

```
ldap=libdldap.dll ldap://host:port/ditbase??scope??
bindname=username password
```

For example:

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

3   Verify that the appropriate environment variable points to the required third-party libraries. The Netscape LDAP SDK libraries are located in *%SYBASE%\%SYBASE_OCS\lib3p*. The Windows PATH environment variable must include this directory.

4   Once you have edited the *libtcl.cfg* file, use dsedit to add a server to the directory service:

a   From the Windows task bar, select Start | Programs | Sybase | dsedit.

b   Select LDAP from the list of servers, and click OK.

c   Click Add New Server Entry.

d   Enter:

*   The server name – this is required.

- Security mechanism – optional. A list of security mechanism OIDs are located in *%SYBASE%\%SYBASE_OCS%\ini\objectid.dat*.

- HA server name – optional. This is the name of the high-availability failover server, if you have one.

e   Click Add New Network Transport.

- Select the transport type from the drop-down list.

- Enter the host name.

- Enter the port number.

f   Click OK twice to exit the dsedit utility.

For more information, see *Configuration Guide* for your platform.

Adding a server to the *sql.ini* file

During installation, the srvbuild process adds entries to the *interfaces* file for your new Adaptive Server, Backup Server, Monitor Server, and XP Server.

To access a Sybase server through the *interfaces* file, each server must be included in the *interfaces* file on the client computer.

To add a server entry to the *sql.ini* file on your PC-client computer:

1   From the Microsoft Windows task bar, select Start | Programs | Sybase | dsedit.

2   Click OK on the first window to open the InterfacesDriver window.

3   Select ServerObject from the menu, and click Add.

4   In the Input Server Name box, enter the name of the server for which you are creating an entry.

Click OK.

5   In the Attributes column, double-click the server address row you just added.

6   In the ProtocolNetwork Address, click Add.

7   From the drop-down list in the Protocol box, select TCP, NAMEPIPE, or SPX.

By default, connections from client products to Adaptive Server are enabled through the Named Pipes and Winsock network protocols.

8   In the Network Address box, you can enter either the server name or the IP address, along with the server port number. For example:

```
machine_name, 4100
```

where *machine_name* is the name of the computer, and *4100* is the port number the server is using to "listen" for clients.

Note Windows now accepts TCP-style connection information:

```
[SERVER]
MASTER=NLWNSCK,huey,2222
QUERY=NLWNSCK,huey,2222
```

or

```
[SERVER]
MASTER=TCP,heuy,2222
QUERY=TCP,heuy,2222
```

The preferred format is to use TCP and a space between the *host_name* and *port_number*, because it is supported across all platforms. You can edit the *sql.ini* with any standard ASCII text editor.

9   To find the IP address for a machine, use ypmatch on a UNIX box.

Note To connect to a UNIX server, the entries you add to *sql.ini* must match the entries in the *interfaces* file on the UNIX system.

For instructions on using dsedit or dscp to modify *interfaces* file entries or to create new *interfaces* file entries for existing servers, see the *Utility Guide*.

For additional information on using dsedit on the client computer, see the *Open Client/Server Configuration Guide for Desktop Platforms* and the *Open Client/Server Supplement*.

For specific information about entries for connections to Monitor Server and Historical Server, see:

•   *Monitor Server User's Guide*

•   *Historical Server User's Guide*

Testing the Sybase Central installation

After you install Adaptive Server, Sybase Central, and the Java Runtime Environment, it is a good idea to test the installation and network connections:

1   Select Start | Programs | Sybase | Sybase Central v4.3 from the Windows task bar to start Sybase Central.

2   Select Tools | Connect from the Sybase Central menu bar to activate a Sybase Adaptive Server login window.

3   Log in using the default user ID, "sa", without a password. If you changed the password for Adaptive Server according to the post-installation instructions, you need to use the new password.

4   From the drop-down list, select the Adaptive Server to which you want to connect.

If Sybase Central is installed on the same computer as the server, click the icon for the server.

If prompted to start Adaptive Server, click OK. When the traffic light icon representing the server is green, the server is running.

> **Warning!** Do not start XP Server from Sybase Central. Adaptive Server starts XP Server automatically when Adaptive Server receives the first request for an extended stored procedure.

5   To disconnect from a server, select Tools | Disconnect.

6   Exit Sybase Central.

If Adaptive Server fails any of these tests, see Chapter 7, "Troubleshooting." Before retrying the installation, follow the instructions in Chapter 3, "Post-Installation Tasks."

Upgrading Adaptive Server

This chapter describes how to upgrade approved versions of Adaptive Server, Backup Server, and Monitor Server to version 12.5.x, and describes how to migrate an Adaptive Server installation built on 32-bit Adaptive Server to run on a 64-bit Adaptive Server after an upgrade.

Overview of the upgrade process

You can upgrade to Adaptive Server 12.5.x from any of these versions:

- 11.5.x

- 11.9.x

For a server installation older than version 11.5.x, Sybase recommends that you upgrade to version 12.0, then upgrade to version 12.5.x.

Only upgrades to and from the same page size are supported. Changing the server schema from a 2K page to *n*K page size is a database migration. See sybmigrate in the *Utility Guide* for information about migration.

Upgrading Adaptive Server consists of four processes:

- Install the new server either into its own installation directory or on top of the old *$SYBASE* director.

> **Note** You must have both the old installation and the new installation to perform an upgrade.

- From the new server installation, perform the pre-upgrade checks on the old server using the preupgrade utility.

- If necessary, fix any problems that pre-upgrade process reports.

- Run the upgrade utility against the old servers to update the underlying schema so that their structures are correct for the new server.

The preupgrade and upgrade utilities are internally called by the sqlupgrade utility.

Each new version of Adaptive Server contains different features that introduce new parameters, commands, reserved words, and so on. For this reason, preupgrade is for preparing the old server for upgrade.

sqlupgrade runs various checks, such as reserved word checks, to determine how much space you must add to the old server to successfully upgrade the old server to the new.

As part of the pre-upgrade tasks, sqlupgrade scans all databases and catalogs and determines how much free space is required to upgrade successfully. Essentially, sqlupgrade searches for the largest catalog, then calculates the required free space by doubling the size of the largest catalog, and adding approximately 10 percent for logging the upgrade changes for each catalog.

During the pre-upgrade process, sqlupgrade returns informational messages as it checks the old server. You must fix all reported problems, and run sqlupgrade cleanly before beginning the upgrade process. Once the old server is eligible for upgrade, sqlupgrade shuts down the old server, starts the new server against the existing databases, and begins the upgrade process.

System catalog changes during upgrade

Adaptive Server version 12.5.x introduces support for wider columns, more columns, a larger number of user logins, and multiple logical page sizes. To support these extended limits, there have been several changes to existing system catalogs.

The catalog upgrade in 12.5.x may affect your existing application. If you are upgrading from pre-12.5, see *What's New in Sybase Adaptive Server 12.5.4?* for a complete list of catalogs that are affected by the relaxed server limits.

Pre-upgrade tasks

Note Before you begin the upgrade process, install Adaptive Server 12.5.x onto your system. See Chapter 2, "Installing Adaptive Server." Also, see "Special Upgrade Instructions" in the release bulletin.

To ensure a successful upgrade, review the following pre-upgrade tasks and perform them as necessary. Depending on the old server configuration, you may not need to perform all pre-upgrade tasks.

1 Check system and upgrade requirements.

2 Check *RUN_server* file location.

3 If you are upgrading Adaptive Server, the previously installed version of the server *must* be running. If you are upgrading Backup Server, Historical Server, Monitor Server, or XP Server, those servers must *not* be running.

4 Procedure text is required for upgrade.

5 Reserved words.

6 Using quoted identifiers.

7 Verify that users are logged off.

8 Check database integrity.

9 Back up databases.

10 Dump transaction log.

11 Make sure that master is the default database for the "sa" user.

12 Prepare the database and devices for upgrade.

13 Create a sybsystemdb database.

Note If you are upgrading from Adaptive Server Enterprise from 11.9.x or earlier, you must create the sybsystempdb.

---

14 Disable auditing.

15 Disable disk mirroring.

16 Verify that your SYBASE environment variable points to the location of the new Adaptive Server software files you just unloaded.

You also can set the OLDSYBASE environment variable to the location of the server you are upgrading, to avoid having to type this path when you run sqlupgrade.

17 Copy the following files from earlier versions of Adaptive Server to their corresponding Adaptive Server 12.5.x installation:

For UNIX platforms:

- *$SYBASE/interfaces*

- *$SYBASE/<servername>.cfg*

- *$SYBASE/$SYBASE_OCS/config/libtcl.cfg*

- *$SYBASE/SYSAM-1_0/licenses/license.dat*

For Windows platforms:

- *$SYBASE\sql.ini*

- *$SYBASE\<servername>.cfg*

- *$SYBASE\$SYBASE_OCS\config\libtcl.cfg*

Checking system and upgrade requirements

To verify that your system environment is configured correctly:

1 Verify that the computer on which you plan to upgrade the Sybase products meets the requirements.

2 Determine whether your server supports an upgrade to Adaptive Server 12.5.x.

3 Set the environment variables by sourcing the *SYBASE.xxx* file.

4 Verify that you unloaded Adaptive Server 12.5.x in a *different* directory than your previous Adaptive Server installation.

   If you overwrote your previous server installation:

   a Restore it from your most recent backup.

   b Reinstall the product files for Adaptive Server in a different directory.

   c Continue with the upgrade.

5 Verify that your operating system is at the proper version level and has all operating system patches needed for Adaptive Server. See the *Release Bulletin* for your platform.

Checking RUN_*server* file location

Verify the name and location of the *runserver* file. Be sure the *runserver* file for your current server is located in:

• *$SYBASE/install* directory for versions 11.5.x and 11.9.x

• *$SYBASE/$SYBASE_ASE/install* for version 12.0.x

Verify that the file is still named *RUN_servername*, where *servername* is the name of the old server. The *servername* must appear exactly as it appears in the *interfaces* file.The *RUN_servername* file for a server SYBASE is called *RUN_SYBASE*. If the *RUN_servername* file for your current Adaptive Server is named, you must change the name during the upgrade process.

Procedure text is required for upgrade

If you created any compiled objects or stored procedures in a pre-11.9.3 server, then removed the syscomments text, the upgrade succeeds but you will encounter runtime problems. To upgrade stored procedures, the stored procedure text must be available in syscomments.

Reserved words

Reserved words are pieces of SQL syntax that have special meaning when used as part of a command.

Transact-SQL does not allow words that are part of command syntax to be used as identifiers, unless they are enclosed in quotation marks, as explained in "Using quoted identifiers" on page 51. If you are upgrading Adaptive Server, and the identifiers in your user databases match new reserved words, errors can result when you run queries, stored procedures, or applications that use these identifiers.

Note You must use sp_renamedb to change the name of any user database that is a reserved word before performing the upgrade.

Conflicts between object names do not prevent the upgrade process from completing. However, applications that refer to conflicting object names may not work after the upgrade. Rename all objects that use reserved words.

As part of the pre-upgrade process, sqlupgrade can perform the reserved word check for you. See the *Reference Manual: Building Blocks* for a complete list of reserved words.

Note If you change an object name, change applications and stored procedures that refer to that object.

Running a reserved word check

If you plan to use sqlupgrade to perform the upgrade process, you do not have to check for reserved words as a separate step. In sqlupgrade, you can check for reserved words at the start of the upgrade process, and then continue with the upgrade if no problems are found.

If you plan to use sqlupgraderes to perform the upgrade process using a resource file, you can run sqlupgrade first to identify reserved word conflicts and other potential upgrade eligibility problems. You can specify that you do not want the upgrade to proceed even if no problems are found.

If you use either sqlupgrade or sqlupgraderes, they automatically install new reserved words and sp_checkreswords, a stored procedure used to detect and display any identifiers in your existing databases that conflict with reserved words in the new database. You can then run the new version of sp_checkreswords at anytime while preforming pre-upgrade tasks.

Note You must use the sp_checkreswords stored procedure from the new version of Adaptive Server to verify that the old installation of Adaptive Server does not use any reserved words that have been introduced with the new server.

Checking for reserved words generates a list of identifiers that conflict with reserved words, and the owners of those identifiers, in the file *$SYBASE/$SYBASE_ASE/init/logs/sqlupgradeMMDD.nnn*. Review this file to determine which identifiers must be changed.

Addressing reserved words conflicts

If any database names are reserved words, you must use sp_renamedb to change the database names before you can upgrade. Use sp_dboption to set the database to single-user mode, and then run sp_renamedb, specifying the new name. See the *Reference Manual: Procedures* for more information on these procedures.

If other identifiers are reserved words, you can:

- Use sp_rename to change the name of the object, before or after the upgrade.

- Use quoted identifiers. See "Using quoted identifiers" on page 51 for more information.

Run sp_checkreswords in master and in each user database to display the names and locations of conflicting identifiers.

For more information about sp_rename and sp_checkreswords and methods for avoiding reserved word conflicts, see the *Reference Manual: Procedures*.

Using quoted identifiers

You can enclose the identifiers that are reserved words in double quotation marks and invoke the quoted_identifier option of the set command in procedures and queries that include the reserved words. The set quoted_identifier option tells Adaptive Server to treat any character string enclosed in double quotation marks as an identifier.

To avoid reserved word conflicts, all users on the server must invoke the quoted_identifier option in all stored procedures and queries that include the reserved words.

Preparing the database and devices for the upgrade

If you ran the pre-upgrade option in sqlupgrade, the utility scanned the system catalogs and calculated how much additional space you need to perform the upgrade.

If you did not run the pre-upgrade check, you must manually calculate how much free space you must add to your system catalogs and databases.As a general rule, you must double the size of the largest catalog that you are going to upgrade, and add approximately 10 percent more to calculate the free space required for the upgrade to succeed. For example, if you have a large number of stored procedures or compiled object in any database, the syscomments and sysprocedures catalogs will require additional space.

Note Sybase recommends that you use preupgrade to determine how much free space is required for the upgrade.

❖ **Increasing default database sizes**

sp_helpdb or Sybase Central can be used to determine the current size of the master, model, sybsystemprocs, and tempdb databases.

To enlarge the master, tempdb, and model databases:

1 In isql, use alter database to increase the size of the master database. For example:

```
1> use master
2> go
1> alter database master on master=x
2> go
```

This example increases the size of the master database; x is the number of megabytes of space added to the existing database size.

2 Repeat this step to increase the size of each of the tempdb and the model databases. For example:

```
1> alter database tempdb on master=x
2> go

1> alter database model on master=x
2> go
```

Note You must increase the size of tempdb before increasing the size of model.

3 Verify the size of each database. Enter the following where database_name is the name of the system database you are checking:

```
1> sp_helpdb database_name
2> go
```

Create a *sybsystemdb* database

In versions 11.5.x and 11.9.x, the sybsystemdb database was required only for servers using two-phase commit transactions. Beginning with version 12.0, all servers must have a sybsystemdb database. Adaptive Server uses this database for tracking transactions and during recovery. In addition, it is used for applications using two-phase commit and distributed transaction management (DTM).

If you have a *sybsystemdb* database

>   If you have a sybsystemdb and the database is at least 4MB, you do not need to make any changes. If the database is smaller than 4MB, increase the size with alter database.

If you do not have a *sybsystemdb* database

>   If you do not have a sybsystemdb, and will not be using two-phase commit or DTM applications, create a sybsystemdb with a minimum of 4MB.

>   If you will be using two-phase commit or DTM, you can either:

>   •   Create a minimum-sized sybsystemdb for upgrade and expand it later, or

>   •   Create a sybsystemdb of 5 to 20MB, depending on your expected usage. Approximately 25 percent of the database should be data storage, and 75 percent should be log storage.

sybsystemprocs

>   sybprocsdev is the default name for this device. However, it is frequently referred to as the sybsystemprocs device, as in the Adaptive Server attribute window, since it stores the sybsystemprocs database.

>   Verify that the sysystemprocs database is large enough. For an upgrade, the recommended minimum size for sybsystemprocs is the larger of 132MB, or enough free space to accommodate the existing sybsystemprocs database, and the largest catalog that is to be upgraded, plus an additional 10 percent for logging upgrade changes. You may need more space if you are adding user-defined stored procedures. 124MB accommodates additional internal data structures, but does not account for the possibility of a larger number of user-defined system procedures. For instructions on increasing the size of the sybsystemprocs database see Appendix E, "Using sybsystemprocs."

Upgrading to Adaptive Server 12.5.x

>   To assist with the upgrade process, Sybase recommends that you set the OLDSYBASE environment variable to point to the old installation of Adaptive Server. If you set the OLDSYBASE environment variable to the old installation of the Adaptive Server, the upgrade utility automatically populates input fields with the correct information as you perform the upgrade.

Use either of the following methods to upgrade Adaptive Server:

- sqlupgrade – provides an X-Windows or Motif GUI for the upgrade process. See "Using sqlupgrade" on page 54.

- sqlupgraderes – provides a noninteractive, file-based interface for the upgrade process. sqlupgraderes allows you to upgrade Adaptive Server using a resource file. This is useful for customer sites that want to upgrade many similar servers. See "Using sqlupgraderes" on page 56.

Using *sqlupgrade*

sqlupgrade verifies that the server is eligible for the upgrade. Perform all pre-upgrade tasks before running sqlupgrade.

To upgrade using sqlupgrade:

1   Make sure your current server is running.

2   Verify that you have performed the tasks mentioned on the window, before beginning the upgrade.

3   Execute sqlupgrade in either of the following ways:

- At the UNIX prompt, enter:

      ```
      $SYBASE/$SYBASE_ASE/bin/sqlupgrade
      ```

  A sqlupgrade window is displayed.

- Alternatively, you may access sqlupgrade through the asecfg utility.

      ```
      $SYBASE/$SYBASE_ASE/bin/asecfg
      ```

  Then click Upgrade an Existing Server.

4   Click OK.

5   The Specify Sybase Directories window displays. Provide the installation directory location for both the new server version and your current server.

    If the SYBASE environment variable is set, sqlupgrade displays the value of the SYBASE environment variable as the directory location for the new server version.

    If the OLDSYBASE environment variable is set, sqlupgrade displays that value as the directory location for your current server.

6   Enter the name of the component directory that contains the new server version. For example, *ASE-12_5*.

7   Enter the full path of the old Sybase release directory. For example, */opt/sybase-12.0*.

If the OLDSYBASE environment variable is set, sqlupgrade displays that value as the directory location for the old server.

8   In the field, "Enter the name of the component directory which contains the old version," enter the Adaptive Server component directory. For example, *ASE-12_0.*

If you are upgrading from a version earlier than 12.0, leave this field blank. Beginning with Adaptive Server 12.0, the directory structure changed.

9   Click OK. The Adaptive Server Selection window displays.

10 From the list of server names provided, select the server you want to upgrade.

If the server you are upgrading is not running, sqlupgrade prompts you to start the server. The server you are upgrading *must* be running.

11 Click OK. The SA Password window is displayed.

12 Enter the password for the System Administrator (sa) login.

13 Click OK. The Specifying Upgrade Options window is displayed.

14 Specify an upgrade option.

The upgrade eligibility tests verify that your current server meets the requirements for upgrading to the new version. (The online help provides a partial list of what the upgrade eligibility tests check.) If your current server fails the upgrade eligibility test, you must fix the problem.

15 Adaptive Server 11.9.x did not automatically create a sybsystemprocs database and device. Unless you created one, the eligibility test fails. The Back button launches the "Specify sybsystemprocs" window.

Enter the full path to the sybsystemprocs device and 124MB for the sybsystemprocs database. Click OK to continue with the eligibility test.

16 Click OK to upgrade your server.

sqlupgrade displays the Status Output window.View the Status Output window for the completion status and informational messages about the upgrade process.

 Warning! Do not interrupt the upgrade, and do not try to connect to
Adaptive Server or run any stored procedures while the upgrade is in
progress.

You can also check the progress of the upgrade by viewing the sqlupgrade
log in *$SYBASE/$SYBASE_ASE/init/logs/sqlupgradeMMDD.NNN*.

After all pre-upgrade checks are successful, the old server is shut down
and the new Adaptive Server dataserver binary is started on the old master
device. sqlupgrade internally runs the upgrade binary to perform the
upgrade.

If the upgrade is successful, a "Done" message is displayed in the Status
Output window.

Note On some UNIX platforms, the sqlupgrade utility may not return you
to the command prompt even though the upgrade has completed
successfully. To exit the utility, press Ctrl + C.

---

17 The upgrade process:

- Creates the *RUN_servername* file

- Runs installmaster and *installjconnect* scripts

18 Check that the upgrade has been successful by logging in to the server and
checking *@@version* to confirm that it is at 12.5.4. Also, sp_configure
'upgrade version' should return 12500.

19 For information about compiled objects, see the troubleshooting chapter.

When the upgrade completes successfully, you can:

- Click OK to specify another server to upgrade, or

- Exit sqlupgrade, and go to "Post-upgrade tasks" on page 58.

Using *sqlupgraderes*

You can upgrade Adaptive Server in noninteractive mode by using values from
a resource file that defines the attributes for the server to be upgraded.

To upgrade an Adaptive Server using a resource file, either:

1   Edit the sample resource file included in your Adaptive Server distribution in *$SYBASE/$SYBASE_ASE/init/sample_resource_files/sqlupgrade. adaptive.server.rs*. Use an ASCII text editor to edit the resource file as described in Appendix A, "Alternative Installation Methods."

2   Run the sqlupgrade utility. This utility creates a new file containing the values you specified for that upgrade session, and writes it to, where *server_name* is the name of the upgraded server:

$SYBASE/$SYBASE_ASE/init/logs/sqlupgradeMMDD.NNN-server_name.rs

---

Warning! Be sure to change device names in the resource file, if necessary, before executing the resource file using sqlupgraderes.

For details on the attributes used in a resource file, see "Resource file attributes for upgrading Adaptive Server," below.

Resource file attributes for upgrading Adaptive Server

Table 5-1 shows the resource file attribute names, their default values, and other options, which you use in the resource file for upgrading Adaptive Server. The sample resource file in the *$SYBASE_ASE/init/sample/upgrade* file included in your Adaptive Server distribution shows examples of these attributes.

All attributes are *required,* and all values are case-sensitive.

If the resource file you are modifying was created by sqlupgrade, the prefix of the attribute name may vary. sqlupgraderes ignores this prefix when processing the resource file.

Table 5-1: Resource file attributes for upgrading Adaptive Server

Attribute	Default value [other options]
sybinit.release_directory	$SYBASE [*path* = *_name_of_old_release*]
sybinit.product	sqlsrv
sqlsrv.server_name	*server_name*
sqlsrv.new_config	no
sqlsrv.sa_login	*current_login*
sqlsrv.sa_password	*current_password*
sqlsrv.do_upgrade	yes

You must also set the following environment variables:

- SYBASE<*new_release_area*>

- SYBASE_ASE ASE-12_5

- SYBASE_OCS OCS-12_5

Upgrading using *sqlupgraderes*

When you have finished editing your resource file, you can run the file using sqlupgraderes.

Note sqlupgraderes is a command line tool; therefore, unlike sqlupgrade, it does not require any X libraries.

To execute sqlupgraderes, at the UNIX prompt, enter the following, where *resource_file* specifies the resource file containing the attributes that describe the server to upgrade:

```
$SYBASE/$SYBASE_ASE/bin/sqlupgraderes -r resource_file
```

Post-upgrade tasks

After you have upgraded to the new version of Adaptive Server, perform the following tasks to make sure your new Adaptive Server is up and running.

Note After you have upgraded, you do not need to run update statistics on any tables. Upgrade process does not cause any changes in existing statistics.

1   Restore functionality in Adaptive Server

2   Reenable Replication Server

3   Reenable auditing

Upgrading Backup Server, Monitor Server, and XP Server

Before you can upgrade Backup or Monitor Server, you must upgrade Adaptive Server.

XP Server does not have a formal upgrade process.

Note Back up all databases, including master, after successfully upgrading Adaptive Server and Backup Server.

❖ **Upgrading Backup and Monitor Servers**

The procedures for upgrading Backup Server and Monitor Server are similar.

* If you are upgrading during the initial installation of Adaptive Server 12.5.x, select Upgrade Existing Servers from the drop-down menu when InstallShield prompts Build or Upgrade Adaptive Server, Backup, and Monitor Servers.

    This launches the sqlupgrade utility. Click OK.

* If you are upgrading after the initial installation, start the sqlupgrade utility from the command line. Enter:

        $SYBASE/$SYBASE_ASE/bin/sqlupgrade

1 A warning message appears. Click OK to continue.

2 From the Server selection window, select the server to upgrade. Click OK.

3 The Specify Sybase Directories window displays. Provide the installation directory location for both the new server version and your current server.

    If the SYBASE environment variable is set, sqlupgrade displays the value of the SYBASE environment variable as the directory location for the new server version.

    If the OLDSYBASE environment variable is set, sqlupgrade displays that value as the directory location for your current server.

4 Provide the release path:

    * Enter the path where the new Sybase server is installed.

    * Enter the name of component directory. Leave this field blank if the Backup Server is older than 12.0.

- • Enter the path to the Open Client/Server Library that the old Backup Server uses. Leave this field blank if it is older than version 12.0.

- • Click OK.

5 When you are prompted for the password, enter the password for the SA, and click OK.

6 The Status output window display the upgrade progress. When the upgrade is complete it displays, "Done." Click OK.

7 You return to the sqlupgrade window. You can either upgrade another server, or click Exit to leave the utility.

Repeat this procedure to upgrade Monitor Server.

Migrating from 32-bit to 64-bit versions

sqlupgrade upgrades Adaptive Server only from one version to another. This section describes how to migrate a 32-bit version of Adaptive Server to a 64-bit version of Adaptive Server.

To migrate Adaptive Server from the 32-bit to the 64-bit version, you must first install and configure the 64-bit operating system.

Method #1: Dump and load

1 Run dbcc checks on all databases in the 32-bit Adaptive Server and make sure they run cleanly.

2 Create a 64-bit Adaptive Server in a new directory.

3 Create devices and databases to match those in the 32-bit server. Make sure that the sysusages mapping is correct. See document ID #1324 at http://my.sybase.com/detail?id=1324.

Note Allow 10 percent additional space for the sybsystemprocs database.

4 Dump the databases on the 32-bit server.

5 Load the databases to the 64-bit server.

6 If you have partitioned tables, update partition statistics.

7 Run dbcc checks on the 64-bit server and make sure they run cleanly.

Method #2: BCP data out and in

If you have DDL scripts to create devices, databases, tables, rules, stored procedures, triggers, and views, you can use bcp to move data out of the old database and into the new.

If you do not have DDL scripts, use the ddlgen utility to re-create the schema for the Adaptive Server you are about to upgrade. See the *Utility Guide*.

1   Run dbcc checks on databases to verify data integrity.

2   Use bcp to extract all the data from all the tables in the databases.

3   Create a new 64-bit Adaptive Server in a new directory.

4   Create devices, databases, and tables.

5   Use bcp to bulk copy data into tables.

6   Re-create all views, triggers, stored procedures.

7   Run dbcc checks on the 64-bit server and make sure they run cleanly.

Method #3: Replacing the binary

The third method requires you to modify some key files, re-create stored procedures, then replace the binary.

1   Run dbcc checks (checkdb, checkalloc, checkcatalog and checkstorage) on all databases in the 32-bit Adaptive Server and make sure they run cleanly.

2   In a new directory, copy the files for the 64-bit Adaptive Server.

3   Shut down the 32-bit server.

4   Copy the *interfaces* file and the configuration file from the 32-bit *$SYBASE* directory to the 64-bit *$SYBASE* directory.

5   Copy the 32-bit *$SYBASE/install/RUN_server* file to the equivalent 64-bit *$SYBASE/install* directory.

6   Edit the *RUN_server* file to reflect the new location of the *interfaces*, configuration, and log files.

7   Remove all references to the 32-bit *$SYBASE* directories from your *$PATH* definition.

8   Change to the 64-bit *$SYBASE* directory and source the *SYBASE.csh* script (C shell) or *SYBASE.sh* script (Korn shell).

9   Change to the 64-bit *$SYBASE/install* directory and execute:

```
startserver -f RUN_server
```

10 After the 64-bit server starts, run installmaster, installmodel, and installmsg.ebf.

11 If you used dbccdb for dbcc checkstorage, run *installdbccdb*. This re-creates some tables in the dbccdb, and you may lose history data.

12 Drop and re-create all your compiled objects, such as stored procedures, triggers, views, and defaults.

13 If you have partitioned tables, update partition statistics.

14 Run dbcc again on all databases to verify that they run cleanly.

Recovering from a failed upgrade

•   During the upgrade process, the log may get full due to the catalog changes that are made. If so, log in to the new server using isql, and issue:

```
isql> dump tran dbname with no_log
```

This frees the log space, and allow the upgrade process to continue.

•   In some cases, the pre-upgrade space estimations might be insufficient for the data copy phase of upgrade. In this case, you may get an error that there is insufficient space in the system segment for the upgrade. The upgrade process hangs, waiting for space to be provided. You can log in to the new server using isql, and use alter database to increase the size of the database.

```
isql> alter database dbname on device_name = "2m"
```

Note In 12.5.x Adaptive Server, alter database allows you to specify the size to alter the database with the unit-specifier 'm', or 'M'.

Downgrading Adaptive Server

Topic	Page
Downgrading from 12.5.4 release	63
Downgrade instructions for Adaptive Server with encrypted columns	71

Downgrading from 12.5.4 release

This section describes instructions to downgrade from Adaptive Server 12.5.4 to Adaptive Server 12.5.3 or an earlier 12.5.x release. These include:

- Basic downgrade steps

- Post-downgrade tasks, whether or not 12.5.4 features are enabled

- Downgrading to Adaptive Server version 12.5.0.x

- Information regarding downgrading individual features

- Returning to Adaptive Server version 12.5.4

Note Adaptive Server 12.5.4 is not an upgrade release, and you may return to previous versions after using the new features in 12.5.4. Sybase recommends that you do not enable the new features offered in Adaptive Server version 12.5.4 until you are committed to staying with 12.5.4.

Basic downgrade steps

Use these basic steps to return to Adaptive Server version 12.5.3 or earlier. Further steps are necessary if you have enabled 12.5.4 features.

1 Overlay the 12.5.3 or earlier binary as you did when installing 12.5.4.

2 Restart the server with the old 12.5.x binary.

3   Run the 12.5.x version of *installmaster* to return stored procedures to their original version.

These are the minimum tasks needed for rollback. Additional steps are described in the rest of the section.

Note Returning to 12.5.3 ESD #7 is preferable, because 12.5.3 ESD #7 contains fixes for several reported errors.

Post-downgrade tasks

After you perform the basic downgrade steps, changes from the 12.5.4 upgrade still persist.

System stored procedures

After you run *installmaster* on your downgraded server, any changed system stored procedures are returned to their 12.5.x status. Any new stored procedures introduced to Adaptive Server in releases subsequent to 12.5.x are not removed. Attempts to execute such stored procedures against an older binary will have unpredictable results.

Unrecognized tokens or objects

The on-disk structures of stored procedures, triggers, and views may contain statement identity tokens, datatypes and object references that are not understood by the earlier Adaptive Server versions. You must drop all compiled objects that use features introduced into Adaptive Server subsequent to the release to which you are downgrading.

If you are returning to a pre-12.5.4 version of Adaptive Server, Sybase recommends that you return to 12.5.0.3. This is because 12.5.0.3 includes error handling code for unrecognized tokens or objects pertaining to features in 12.5.4. If the 12.5.0.3 Adaptive Server encounters these unknown objects, it handles them with a meaningful error message.

Should you return to a pre-12.5.0.3 version of Adaptive Server, no such soft landing is provided. Using the new features can result in inexplicable behavior, misleading error messages and stack traces.

System tables

After returning to a pre-12.5.4 binary:

- When returning to 12.5.0.x or earlier, the new datatypes date, date null, time, and time null remain in systypes but you cannot use them in 12.5.0.3 and attempts to use them in 12.5.0.2 or earlier will cause a stack trace. See the special procedure for downgrading to Adaptive Server version 12.5.0.x to avoid problems with new datatypes.

- sp_values is dropped and re-created when you run the 12.5.x installmaster, so any new types are eliminated from this table.

- sysmessages still contains all the new error messages, but that is not a problem because they are not used.

Configuration parameters

New configuration parameters have been added in 12.5.4. When starting the previous version of the Adaptive Server, use the old *.cfg* configuration file.

If you forget to run the 12.5.x installmaster before starting Adaptive Server, an error message displays at startup.

Running the 12.5.x version of installmaster removes configuration parameters that belong to 12.5.4, by deleting sysconfigures rows for configuration parameters that do not exist in syscurconfigs. After running installmaster the error messages will no longer appear when you start the server.

Keywords

Two keywords, func and function, were removed in Adaptive Server 12.5.4, making it possible to create identifiers using these names. If you did not use these as identifiers under 12.5.4, no action is needed. If you did, the parser generates a syntax error under 12.5.0.x since it interprets these words as keywords.

See "Additional steps required if you used 12.5.4 features before downgrading" on page 66 for more information.

Downgrading to Adaptive Server version 12.5.0.x

Perform these steps only when you are downgrading to 12.5.0.x. Returning to 12.5.0.3 is preferable, because these versions can handle a number of error conditions that may arise as a result of downgrading. Older version of Adaptive Server cannot handle these situations and results are unpredictable.

Although it is not necessary to restore systypes and syscharsets to their previous state, Sybase provides the store procedure sp_twelve_five_one to do so.

After returning to Adaptive Server version 12.5.0.x, remove the new datatypes from systypes and syscharsets and return these tables to their 12.5.0.x state by performing the following steps:

1   Enable updates to system tables:

```
sp_configure "allow updates", 1
```

2   Remove the 12.5.4 datatypes:

```
sp_twelve_five_one "remove"
```

3   Disable updates to system tables:

```
sp_configure "allow updates", 0
```

This procedure also marks all procedural objects for recompiling, to insure that the pre-12.5.4 Adaptive Server does not encounter 12.5.4 objects or datatypes.

Additional steps required if you used 12.5.4 features before downgrading

If you are rolling back after having used any of the 12.5.4 features, additional steps can be necessary. You should take steps before you downgrade to pre-12.5.4, and others should be performed immediately after downgrading. Areas of concern are listed for each new feature.

In general, no additional steps are required when you are returning to an Adaptive Server version in which the feature was already available. When returning to a server version earlier than the version in which the feature was introduced, you must be aware of the areas of concern listed below.

Encrypted Columns

If you have used the encrypted columns feature of 12.5.4, you must take steps in 12.5.4 before you downgrade. Perform the following. See "Downgrade instructions for Adaptive Server with encrypted columns" on page 71.

Real-Time Messaging

Drop all stored procedures, views, and triggers that use the messaging built-ins for the Real-Time Messaging feature.

For more information about Real Time Messaging, see the *Real-time Data Services Messaging User's Guide*.

New sort orders

Upgrading from 12.5.0.3 to 12.5.4 or later modifies syscharsets. All of the sort orders that were (formerly) destined for use with unichar was rewritten in terms of ID and CSID, although the names remain the same. Other sort orders (that is, any that you loaded with the charset utility or the sqlloc GUI front end) remain unmodified.

While running under 12.5.4, you may configure UTF-8 with a non-binary sort order. If you return to 12.5.0.3 that server fails to start, with a message saying that the character set/sort order combination is not supported. Versions earlier than 12.5.0.3, however, do not have the error handling code; you do not get the message but instead see inexplicable behavior. The most likely symptom is that the server cannot find anything using indexes on the system tables.

To get around this:

- Restart the 12.5.4 server.

- Configure UTF-8 with binary sort order bin_utf8 (ID 50). Do not use any other binary sort order (there are three to choose from); bin_utf8 is the only one known to 12.5.0.3 and earlier servers.

Now when you start 12.5.0.3, syscharsets is still in the 12.5.4 state. If you run sp_helpsort, rewrite syscharsets to a close approximation of what it was before the upgrade.

Note If you execute sp_helpsort now, and then decide to go back to 12.5.4, be aware that although syscharsets is in the 12.5.0.3 state, the upgrade does not run again since it has already run once. Execute sp_helpsort again on return to 12.5.4 to update the syscharsets table.

Unicode parser

If you have taken advantage of unicode parser feature through explicit or implicit use of the U&'syntax' (in a stored procedure, for example), the text of the stored procedure is saved in syscomments. If you roll back to 12.5.0.3 and the server needs to recompile from query text, you get syntax errors.

On another note, if you take advantage of the fact that in 12.5.4 unichar no longer requires UTF-8, you may have stored procedure plans that contain unichar-to-char conversions. In 12.5.4 this is handled correctly on a non-UTF-8 server. In 12.5.0.3 and earlier, however, the assumption was UTF-8; anything else causes the conversion to throw an error and abort the transaction.

date/time datatypes

When you start the 12.5.4 server, the new system types are installed in systypes. If you returned to 12.5.0.3:

*   The new types remain in systypes, but server operation is not affected provided you did not use the new types on the 12.5.4 server

*   You cannot create using date or time because error checks were added

*   If the server encounters date/time tokens, it raises error messages

*   If the server encounters a 12.5.4 procedure that accesses a table with date/time types, a stack trace results.

If you returned to a pre-12.5.0.3 server, nothing prevents you from creating objects using date and time types (since no error handling code exists for this purpose), but this is not advisable because any operations on those objects such as selects or inserts result in stack traces.

If you have created 12.5.4 tables with date or time columns, or procedures with date or time parameters, you need to remove references to these types before rolling back to a pre-12.5.4 server. You can run the following, which identifies the offending tables and procedures, advising you to remove references to them or use alter table to modify the types:

```
sp_date_time_type
```

If sp_date_time_types does not find any date/time objects, it marks procedural objects for recompile, removing all internal references to these new types

SQL derived tables

If you used SQL derived tables before returning to a pre-12.5.4 version server, and you:

*   Create views with SQL derived tables using version 12.5.4 roll back to ASE 12.5.0.3, then try to select from the view, you get an error that derived tables are not supported in that version.

*   Create stored procedures with SQL derived tables in 12.5.4 roll back to ASE 12.5.0.3 or earlier versions and run the stored procedure, you will see an error that the object (SQL derived table) does not exist.

*   Create views with SQL derived tables in 12.5.4 return to a pre-12.5.0.3 version, you get stack trace errors.

Native XML

The XML Services feature of Adaptive Server includes the following new built-in functions and clauses:

*   xmlextract built-in function

*   xmlparse built-in function

*   xmltest predicate

*   for xml clause of select commands

If you create views or stored procedures using one of these features in 12.5.4, then return to 12.5.0.3, you get an error that the feature is not supported in that version. If you store XML documents in:

*   Their original text form, they can be processed on both Adaptive Server 12.5.4 and ASE 12.5.0.3.

- The parsed form generated by the xmlparse built-in function, then return to 12.5.0.3, the parsed form is not accessible. You must first regenerate the text form of the XML documents in 12.5.4.

For example, if a column named xmlindexed contains data generated in Adaptive Server 12.5.4 by the xmlparse built-in function, you can regenerate the text form of the document in the xmlsource column with the following SQL statement that uses the Adaptive Server 12.5.4 xmlextract built-in function:

```
update xmltab
set xmlsource = xmlextract("/", xmlindexed)
```

You may then do one of the following:

- Process the xmlsource column directly with the Java-based XQL processor, using the com.sybase.xml.xql.Xql.query Java-based method, or

- Update the xmlindexed column with the parsed form suitable for processing with the Java-based XQL processor, using the following statement:

```
update xmltab
set xmlindexed = com.sybase.xml.xql.Xql.parse(xmlsource)
```

If you do not want to add the xmlsource column, you can combine these steps with the following SQL statement in ASE 12.5.4:

```
update xmltab
set xmlindexed
= com.sybase.xml.xql.Xql.parse
(xmlextract("/", xmlindexed))
```

Prior to execution of this update statement, the xmlindexed column contains the parsed form of the documents generated by the xmlparse 12.5.4 built-in function. After the update statement, that column contains the parsed form of the documents, generated by the com.sybase.xml.xql.Xql.parse Java-based method, suitable for processing in Adaptive Server12.5.0.3 with the com.sybase.xml.xql.Xql.query Java-based method.

A configuration parameter was added to enable XML in 12.5.4. When you first restart with the prior version, you may see error message 5859 advising you that the parameter is unknown. After running installmaster from the pre-12.5.4 version this warning should no longer appear.

Automatic database expansion

If you installed the 12.5.4 automatic database expansion procedures using installdbextend, and then applied the threshold procedure to one or more database segments, the thresholds might not work properly when applied to the log segment after a downgrade to 12.5.0.3.

To clear all auto-expansion thresholds that might exist on one or more segments before downgrading, use the command:

```
sp_dbextend 'clear', 'threshold'
```

Alternatively, you can disable the entire automatic expansion feature server-wide without changing any existing rules or clearing any thresholds. Execute the following commands using sa_role:

```
use master
go
sp_dbextend 'disable', 'database', 'server-wide'
go
```

This prevents threshold procedures from doing any work even if they were fired at runtime.

Sybase recommends that you leave all the policies and thresholds in place, and simply disable the entire feature server-wide before the downgrade. This simplifies re-enabling automatic expansion if you return to 12.5.4 later.

Downgrade instructions for Adaptive Server with encrypted columns

This section details the steps to be taken for downgrading a 12.5.4 server to earlier versions with regard to encrypted columns.

Back up all your databases and the *$SYBASE* release area prior to the 12.5.4 downgrade.

If you are returning to Adaptive Server version 12.5.3a, you do not need to take any steps to prepare your database for downgrade, whether or not you enabled encrypted columns in your 12.5.4 Adaptive Server.

If you are returning to Adaptive Server version 12.5.3 , returning to 12.5.3 ESD #7 is preferable, because 12.5.3 ESD #7 contains fixes for a number of error conditions.

Depending on whether enable encrypted columns has been configured in your server, you must take a different set of actions before using an older version of Adaptive Server with 12.5.4 databases.

One way to verify that you have never configured encrypted columns is to check whether the system table sysencryptkeys exists in any database. If the catalog does not exist, you have never configured encrypted columns in your server.

Downgrading from a 12.5.4 server to an earlier version of 12.5.x when encrypted columns has not been configured

Use the following procedure when downgrading from a 12.5.4 Adaptive Server that has not been configured for encrypted columns.

1   Start the 12.5.4 server in single-user mode to guarantee that no other user can access Adaptive Server while you are going through the downgrade steps. See the *Utility Guide* for details on starting the server in single user mode.

2   A user with sso_role and sa_role must execute:

        sp_encryption remove_catalog

    This stored procedure removes the new columns added in syscolumns from each database. If sp_encryption is successful in removing the new columns from syscolumns, it also removes the record of the upgrade item from sysattributes in each database. If a database is unavailable, the command prints an error message and exits. You should bring the unavailable database online and execute sp_encryption remove_catalog again.

3   Drop the system stored procedure sp_encryption from the sybsystemprocs database.

4   Shut down the server. You can now use a 12.5.x Adaptive Server binary from a pre-12.5.4 version.

5   Copy the RUN_SERVER file to a 12.5.x release area and modify it to use the data server binary from the 12.5.x release area.

6   Restart the server using the modified RUN_SERVER file.

7   Run the 12.5.x version of installmaster to return system stored procedures to their original version.

Downgrading from a 12.5.4 server to an earlier version of 12.5.x when encrypted columns has been configured

Use the following procedure to downgrade from a 12.5.4 server to an earlier version of 12.5.x when encrypted columns has been configured.

1   If encrypted columns are not currently enabled, the system security officer executes:

```
sp_configure 'enable encrypted columns',1
```

2   Use drop or alter to decrypt all tables with encrypted columns in all databases. The system security officer runs the following command in each database where encryption keys were created to list all encryption keys created in that database:

```
sp_encryption help
```

For each key listed, the system security officer runs the following to see a list of columns encrypted with a particular key:

```
sp_encryption help, <keyname>, 'display_cols'
```

For each encrypted column, one of the following steps must be performed:

a   alter table to decrypt the encrypted column

b   alter table to drop the encrypted columns

c   drop the table containing the encrypted column

After the data encryption has been removed, drop the encryption key.

3   To guarantee that no other user can access Adaptive Server while a system table is removed, restart the server in single-user mode. See the *Utility Guide* for details on starting the server in single-user mode.

4   To remove the sysencryptkeys catalog and the new columns in syscolumns from each database, a user with sso_role and sa_role must execute:

```
sp_encryption remove_catalog
```

If a database is unavailable, the command prints an error and exits. You should bring the unavailable database online and execute sp_encryption remove_catalog again.

If columns encrypted by any key in sysencryptkeys exist, the command does not drop sysencryptkeys, but prints an error or warning and continues with the next database. If sp_encryption is successful in removing sysencryptkeys and the new columns from syscolumns, it also removes these rows from sysattributes in each database:

> * The record of the upgrade item that added sysencryptkeys

> * The record of the upgrade item that added the new columns in syscolumns

> * The system encryption password for the database

5 Drop the system stored procedure sp_encryption from the sybsystemprocs database.

6 Shut down the server. You can now use a 12.5.x Adaptive Server binary from a pre-12.5.4 release area.

7 Copy the RUN_SERVER file to a 12.5.x release area and modify it to use the data server binary from the 12.5.x release area.

8 Restart the server using the modified RUN_SERVER file.

9 Run the 12.5.x version of *installmaster* to return stored procedures to their original version.

To re-enable encrypted columns, when rolling forward from a downgraded 12.5.4 server back to 12.5.4, configure enable encrypted columns. Upon restarting the 12.5.4 server, the sysencryptkeys system table and the new columns in syscolumns are installed in each database.

Replication issues with downgrade

When downgrading a server that has replication enabled on databases that contain encrypted data, you must do one of the following before you start the downgrade procedure:

1 Ensure that all replicated data in the primary database transaction log has been successfully transferred to the standby or replicate database. The process for doing this is application dependent.

2 Using the following commands truncate the transaction log in the primary database, and zero the RS locator for that database in the Replication Server. In the primary database run:

```
sp_stop_rep_agent primary_dbname
dbcc settrunc ('ltm', 'ignore')
dump tran primary_dbname with truncate_only
dbcc settruc ('ltm', 'valid')
```

Shutdown Replication Server. In the RSSD for the Replication Server run:

```
rs_zeroltm primary_servername, primary_dbname
```

CHAPTER 7   **Troubleshooting**

This chapter provides instructions for troubleshooting installation error messages.

If this chapter does not describe the error message you are encountering, see the *Error Messages and Troubleshooting Guide*.

Topic	Page
Overview	75
Error log locations	76
Solutions to common installation problems	77
Troubleshooting SySAM	84
Upgrading compiled objects with dbcc upgrade_object	85

Overview

For Sybase server products, there are two categories of errors:

*   Errors generated by the installation, upgrade, and setup utilities

*   Errors generated by the server (Adaptive Server, Backup Server, and so on)

To determine the cause of an error, first look in the log file of the utility being used, to identify the task the utility was performing when it failed. Then check the server error log. See "Error log locations" on page 76.

Table 7-1 lists possible causes and solutions for common problems that you might encounter during a first-time installation or upgrade. If you continue to have problems, retry the installation or upgrade.

If the installation program or svrbuild unexpectedly quits, or if you cannot correct the problem, see the *Error Messages and Troubleshooting Guide*.

Table 7-1: Troubleshooting guidelines

Problem	Possible cause and solution
The installation program cannot start Adaptive Server.	Failure to start Adaptive Server is generally caused by a lack of available RAM or disk space.
	Make sure you meet RAM requirements. If you have the required RAM, remove, then reinstall all applications to the hard drive and restart the installation process.
	After Adaptive Server is installed, there should be 25MB of free disk space left in the disk drive. Adaptive Server needs approximately 18MB for creating shared memory files.
	Verify that you are logged in as System Administrator. You must log in as an administrator to start Adaptive Server.
	Shut down Monitor Server before restarting Adaptive Server.
After upgrading Adaptive Server, you cannot use svrbuild.	After you begin upgrading a server, you may be unable to use the same svrbuild session for other tasks. Exit and restart svrbuild.
The installation program cannot connect to the upgraded Adaptive Server.	After you begin upgrading a server, you may be unable to use the same svrbuild session for other tasks. Exit and restart svrbuild.
The installation program detects reserved word conflicts.	See "Stopping Adaptive Server after a failure" on page 80.
The upgrade fails.	See "Recovering from a failed upgrade" on page 82.

Error log locations

The information in the error logs helps you determine the reason and possible solution for an error message.

Installation utilities

Table 7-2 lists the default error log locations for the installation, upgrade, and setup utilities.

Table 7-2: Error log locations for installation utilities

Utility	Error log location
InstallShield	*$SYBASE/log.txt*
srvbuild	*$SYBASE/$SYBASE_ASE/init/logs/srvbuildMMDD.NNN*
srvbuildres	where *MM* is the month, *DD* is the date, and *NNN* is a three-digit number identifying the srvbuild session
sqlloc	*$SYBASE/$SYBASE_ASE/init/logs/sqllocMMDD.NNN*
sqllocres	
sqlupgrade	• *$SYBASE/$SYBASE_ASE/init/logs/sqlupgradeMMDD.NNN*
sqlupgraderes	• *$SYBASE/$SYBASE_ASE/upgrade/upgrade.NNN* (a temporary file created by the upgrade process)

Sybase servers

Table 7-3 lists the default error log locations for each Sybase server.

Table 7-3: Error log locations for Sybase servers

Server	Default error log path and file name
Adaptive Server	*$SYBASE/$SYBASE_ASE/install/servername.log*
Backup Server	*$SYBASE/$SYBASE_ASE/install/servername_back.log*
Monitor Server	The directory from which Monitor Server is started. The error log file name is *ms.log*.
XP Server	The Adaptive Server error log.

Solutions to common installation problems

If this section does not describe the problem you are experiencing, see the *Error Messages and Troubleshooting Guide*.

Unable to use X-Windows

If the setup and configuration utilities do not display correctly, you may have to adjust the resolution on your monitor.

To change to a smaller font size, issue the following UNIX commands:

```
% cd $SYBASE/ASE-12_5
```

```
% chmod +w xappdefaults
% cd xappdefaults
% chmod +w *
% foreach i(*)
? cat $i | sed -e "s/140/100/g" | sed -e "s/^#D/D/g" | sed -e "s/^#S/S/g" > p
? mv p $i
? end
%
```

The installation utilities will now use approximately 25 percent less window space.

Cannot eject the CD from the drive

If you cannot eject the CD from the drive:

- Check to see whether the CD drive path is the current directory (pwd) in a UNIX terminal window. If it is, change (cd) to another directory.

- Check for sybhelp processes. If these processes exist, kill them using the UNIX kill command.

DISPLAY environment variable not set correctly

This error message means that the DISPLAY environment variable on the *remote* machine is not set correctly to display InstallShield to your *local* machine:

```
The DISPLAY environment variable is not set correctly.
```

To correct the problem, enter the following command at the UNIX prompt of the *remote* machine, enter the following, where *host_name* is the name of the machine on which you want InstallShield to appear (that is, on your *local* machine):

For C shell:

```
setenv DISPLAY host_name:0.0
```

For Bourne shell:

```
DISPLAY=host_name:0.0; export DISPLAY
```

Client not authorized to connect to server

If you run InstallShield and you get this error message, it means the remote machine does not have permission to display the user interface on the local machine where you start working:

```
Xlib: connection to "host_name" refused by server
Xlib: Client is not authorized to connect to Server
xhost: unable to open display "host_name"
```

To correct the problem, enter the following command at the UNIX prompt of your *local* machine, where *remote_machine* is the machine on which you are running InstallShield:

```
xhost +remote_machine
```

Restart InstallShield.

Address already in use

The following message from srvbuild means you entered a port number that is already in use:

```
kernel: ninit: bind, Address already in use
```

To correct the problem, enter a different port number on the srvbuild window. The command netstat -a produces a list of port numbers in use.

Adaptive Server fails to start

The shared memory of the operating system may not be set high enough.

To correct the problem, see Chapter 2, "Installing Adaptive Server" for instructions on adjusting the shared memory value. Restart the installation or upgrade process.

InstallShield fails to start

If InstallShield does not start, restart it, using:

```
"-is:javaconsole"
```

This displays any error messages to the console which user can then correct.

Cannot start XP Server

You may receive the following message from XP Server, when it is invoked by xp_cmdshell or some other extended stored procedure:

```
Msg 11018, Level 16, State 1:
Procedure "xp_cmdshell", Line 2:
XP Server must be up for ESP to execute.
(return status = -6)
```

Verify that there is an XP Server entry in Adaptive Server's sysservers table. If you created XP Server in a different srvbuild session than Adaptive Server and you did not specify a related Adaptive Server, srvbuild cannot update the sysservers table. You also need to check verify that the XP server exists in the interfaces file or LDAP server.

Use sp_addserver to add an entry to the sysservers table.

Stopping Adaptive Server after a failure

If the installation or upgrade session fails after you start Adaptive Server, use the shutdown command:

1   Log on as "sa".

2   Shut down Adaptive Server using the shutdown with nowait command. Using the with nowait option stops the Adaptive Server immediately, without waiting for currently executing SQL statements to finish:

```
1> shutdown with nowait
2> go
```

Recovering from a failed installation

If the installation does not succeed, the installation program displays error messages. Review the error messages and your Adaptive Server error log to determine the cause of the installation failure. For default error log locations, see Table 7-3 on page 77.

If installation fails after files are created

If the installation program quits while you are configuring Adaptive Server:

1   View the contents of the log file generated by Adaptive Server. For default
     error log locations, see Table 7-3 on page 77.

2   Take any suggested actions to correct the problem.

3   If the installation fails *after* the installation program has created any
     operating system files, such as the master device or system procedures
     device files, delete those files.

4   If the installation fails *after* the installation program starts the Adaptive
     Server that you are attempting to install, shut down that server.

5   Use asecfg to restart the configuration.

Troubleshooting resource file installations

If you encounter problems during the build, configuration, or upgrade process
of Adaptive Server using the srvbuild[res], sqlloc[res], or sqlupgrade[res], it may
be that these utilities did not allow enough time for Adaptive Server to shut
down properly.

You can set the environment variable SYBSHUTWAIT to force the utilities to
wait for Adaptive Server to shut down. For example, this command forces the
utility to wait for 2 minutes to allow Adaptive Server to shut down before
proceed with the next task:

```
% setenv SYBSHUTWAIT 120
```

If Adaptive Server fails the pre-upgrade test, sqlupgrade displays:

```
Server SERVER_NAME failed preupgrade eligibility test.
See log for more information.
```

1   From the Upgrade window, select Exit.

2   Examine the log file created in the *$SYBASE/$SYBASE_ASE/init/logs*
     directory to find out why Adaptive Server or SQL Server failed the pre-
     upgrade eligibility test.

     After you resolve any problems, shut down Adaptive Server or SQL
     Server and use sqlupgrade to complete the upgrade session.

Recovering from a failed upgrade

If the upgrade process fails, the installation program displays error messages. Review the error messages and the Adaptive Server error log to determine the cause of the upgrade failure. For default error log locations, see Table 7-2 on page 77.

Restoring from backup

You may need to restore your databases due to a failed upgrade.

If you think the upgrade failure or its cause may have damaged your databases, restore the databases from backups. For information on restoring databases, see the *System Administration Guide*.

If you are concerned about the possible corruption of your databases, exit sqlupgrade, but do not attempt to restart the upgrade session until you have restored the databases from backup. After restoration is complete, retry the upgrade.

Re-running the upgrade

Whether you can safely re-run the upgrade depends on when the failure occurred in the upgrade process. If the failure occurs while the message: "Starting to upgrade Adaptive Server" is displayed, it is safe to re-run the upgrade program.

1 Try to fix the problem that caused the upgrade to fail.

2 Run the upgrade again.

If the upgrade fails:

* *Before* returning the message "Setting upgrade version to 12.5," you may need to restore your latest database backup, and restart the upgrade.

* *After* returning the message "Setting upgrade version to 12.5," it is not necessary to restart the upgrade. The installation utility considers the upgrade to be complete.

Also, it is not necessary to restore a database from a backup unless that database failed during the upgrade.

Recording the upgrade manually

If Adaptive Server did not finish recording the upgrade in the sysattributes table before the failure occurred.

1   Fix the problem that caused the failure.

The first error message indicates the cause of the failure. If you can, solve the problem and proceed to step 2. For example, you can usually correct an 1105 error with a dump transaction command. However, you may need to refer more complex problems to Sybase Technical Support.

2   Execute the following SQL statements to allow Adaptive Server to complete recording the upgrade:

```
1> declare @dbname varchar(30)
2> select @dbname = min(name)
3> from sysdatabases
4> while @dbname is not null
5> begin
6> online database @dbname
7> select @dbname = min(name)
8> from sysdatabases
9> where name > @dbname
10> end
```

If the cause of the failure is known

If the error logs or messages clearly indicate the cause of failure, and you do not believe your databases were damaged, you can attempt to fix the problem and re-run the upgrade immediately.

1   Exit the sqlupgrade program.

2   Perform the necessary actions to fix the problem.

For example, if the error log indicates that the upgrade failed because your existing databases do not contain enough space, use the alter database command to increase the available space.

3   You may need to shut down Adaptive Server.

Shutting down the server enables the installation program to start the server and re-run the upgrade session.

4   Start sqlupgrade again.

5   Select Upgrade Adaptive Server, and proceed with the upgrade.

If the cause of the failure is unknown

If the upgrade process fails again, and you cannot determine the cause of failure, check the error log file to find out when and where the upgrade failed, and contact Sybase Technical Support.

By default, the log file is located in *$SYBASE/$SYBASE_ASE/install/errorlog*.

Troubleshooting SySAM

When you start Adaptive Server with SySAM support, problems acquiring licenses or contacting the asset management software appear in the Adaptive Server error log file, *lmgrd.log* (*$SYBASE/$SYBASE_SYSAM/log*).

Fixing license certificate input errors

When you purchase licenses for Sybase Adaptive Server products, you are issued a Sybase Software Asset Management Certificate. The certificate has the following information for each product:

- Order Number

- Feature Name

- Feature Count

- Software Version

- Authorization Code

- Product Description

This information is used by SySAM to build the license file, with new licensed features appended to the end of the file. If license keys are not working, it may be that the information entered on the License Entry application (lmgr) was typed incorrectly. In this case, you can check the license key in question (for example, ASE_HA) to make sure the information is correct. If an error was made, you can correct it using a text editor.

Here is a sample license key entry from a *license.dat* file with information on how to map the information in the license file to the information that appears on the Sybase Software Asset Certificate.

```
INCREMENT ASE_HA SYBASE 12.5 permanent 5A234B567C890\
```

```
OVERDRAFT=10000   ck=0   SN=30181-4
```

- INCREMENT - indicates that the count of this license key will be added to the count of any other license keys for the same feature (all SYBASE license keys use INCREMENT).

- ASE_HA - is the Feature Name entered.

- SYBASE - identifies this license key as belonging to the SYBASE vendor daemon (case sensitive, must be uppercase).

- 12.5 - the Feature Version entered.

- permanent - indicates this license has no expiration date (some license keys have a date instead.)

- 5 - is the Feature Count entered.

- A234B567C890 - is the Authorization Code entered.

- OVERDRAFT=10000 - indicates that you can check out more features than you are licensed for, but you will be warned that you have overused your license (case sensitive, must be uppercase)

- ck=0 - the checksum (it is not used and is usually 0).

- SN=30181-4 - the Order Number entered (case-sensitive, the SN part must be uppercase.)

Many of the fields in the license key are case sensitive; you must take care when editing a license key. The only time you should edit a license key is when the values have been entered incorrectly and the key is reported as invalid. Making any change to a license key that is valid makes the license key invalid.

The file is located in *$SYBASE/$SYBASE_SYSAM/licenses/license.dat*.

Upgrading compiled objects with *dbcc upgrade_object*

Adaptive Server version 11.9.3 introduced the process of upgrading compiled objects based on their source text. Compiled objects are:

- Check constraints

- Defaults

- Rules

- Stored procedures (including extended stored procedures)

- Triggers

- Views

The source text of each compiled object is stored in the syscomments table, unless it has been manually deleted. When you upgrade the server, the existence of the source text in syscomments is verified during that process. However, the compiled objects are not actually upgraded until they are invoked.

For example, if you have a user-defined stored procedure named list_proc, the presence of source text for list_proc is verified when you upgrade to Adaptive Server 12.5.x. The first time list_proc is invoked after the upgrade, Adaptive Server detects that the list_proc compiled object has not been upgraded. Adaptive Server recompiles list_proc, based on the source text in syscomments. The newly compiled object is then executed.

Upgraded objects retain the same object ID and permissions that they used before being upgraded.

Compiled objects for which the source text was hidden using sp_hidetext are upgraded in the same manner as objects for which the source text is not hidden. For information on sp_hidetext, see the *Reference Manual: Procedures*.

Note If you are upgrading from 32-bit installations to use a 64-bit Adaptive Server, the size of each 64-bit compiled object in the sysprocedures table in each database increases by approximately 55 percent when the object is upgraded. The pre-upgrade process calculates the exact size. Increase your upgraded database size accordingly.

To ensure that compiled objects have been upgraded successfully *before* they are invoked, you can upgrade them manually using the dbcc upgrade_object command. For details, see "Finding compiled object errors before production" on page 86.

Finding compiled object errors before production

Changes made in earlier versions of Adaptive Server may cause compiled objects to work differently in version 12.5.x and later. You can use dbcc upgrade_object to find the following errors and potential problem areas that may require manual changes to achieve the correct behavior:

- Reserved word errors

- Missing, truncated, or corrupted source text
- Quoted identifier errors
- Temporary table references
- select * potential problem areas

After reviewing the errors and potential problem areas, and fixing those that need to be changed, you can use dbcc upgrade_object to upgrade compiled objects manually instead of waiting for the server to upgrade the objects automatically. For details, see "Using dbcc upgrade_object" on page 90.

Reserved word errors

If dbcc upgrade_object finds a reserved word used as an object name in a compiled object, it returns an error, and that object is not upgraded. To fix the error, either manually change the object name or use quotes around the object name and issue the command set quoted identifiers on. Then, drop and re-create the compiled object.

For example, suppose you load a database dump from Adaptive Server 11.5 into Adaptive Server 12.5.x and the dump contains a stored procedure that uses the word "lock." When you run dbcc upgrade_object on that stored procedure, the command returns an error because, although "lock" was not reserved in version 11.5, it became a reserved word in version 11.9.2. With this advance notice, you can change the stored procedure and any related tables before they are used in a production environment.

Missing, truncated, or corrupted source text

If the source text in syscomments was deleted, truncated, or otherwise corrupted, dbcc upgrade_object may report syntax errors. If the source text was not hidden, you can use sp_helptext to verify the completeness of the source text. If truncation or other corruption has occurred, drop and re-create the compiled object.

Quoted identifier errors

dbcc upgrade_object returns a quoted identifier error if:

- The compiled object was created in a pre-11.9.2 version with quoted identifiers active (set quoted identifiers on).

• Quoted identifiers are not active (set quoted identifiers off) in the current session.

To avoid this error, activate quoted identifiers before running dbcc upgrade_object. When quoted identifiers are active, you must use single quotes instead of double quotes around quoted dbcc upgrade_object keywords.

If quoted identifier errors occur, use the set command to activate quoted identifiers, and then run dbcc upgrade_object to upgrade the object.

For compiled objects created in version 11.9.2 or later, the upgrade process automatically activates or deactivates quoted identifiers as appropriate.

Note Quoted identifiers are not the same as literals enclosed in double quotes. The latter do not require you to perform any special action before the upgrade.

Temporary table references

If a compiled object such as a stored procedure or trigger refers to a temporary table (#temp *table_name*) that was created outside the body of the object, the upgrade fails, and dbcc upgrade_object returns an error. To correct this error, create the temporary table exactly as expected by the compiled object, then execute dbcc upgrade_object again. You need not do this if the compiled object is upgraded automatically when it is invoked.

select * potential problem areas

In Adaptive Server version 11.9.3 and later, the results of a select * clause in a stored procedure, trigger, or view that was created in an earlier version of Adaptive Server may be different from what you expect.

For more information about the changes, see the *Reference Manual: Commands*.

If dbcc upgrade_object finds a select * clause in the outermost query block of a stored procedure, it returns an error, and does not upgrade the object.

For example, consider the following stored procedures:

```
create procedure myproc as
  select * from employees
go
create procedure yourproc as
  if exists (select * from employees)
    print "Found one!"
```

```
go
```

dbcc upgrade_object returns an error on myproc because myproc includes a statement with a select * clause in the outermost query block. This procedure is not upgraded.

dbcc upgrade_object does not return an error on yourproc because the select * clause occurs in a subquery. This procedure is upgraded.

Determining whether *select* * should be changed in views

If dbcc upgrade_object reports the existence of select * in a view, compare the output of syscolumns for the original view to the output of the table, to determine whether columns have been added to or deleted from the table since the view was created.

For example, suppose you have the following statement:

```
create view all_emps as select * from employees
```

Before upgrading the all_emps view, use the following queries to determine the number of columns in the original view and the number of columns in the updated table:

```
select name from syscolumns
  where id = object_id("all_emps")
select name from syscolumns
  where id = object_id("employees")
```

Compare the output of the two queries. If the table contains more columns than the view, and retaining the pre-upgrade results of the select * statement is important, change the select * statement to a select statement with specific column names. If the view was created from multiple tables, check the columns in all tables that comprise the view and rewrite the select statement if necessary.

 Warning! Do not execute a select * statement from the view. Doing so upgrades the view and overwrites the information about the original column information in syscolumns.

Another way to determine the difference between the columns in the view and in the new tables is to run sp_help on both the view and the tables that comprise the view.

This comparison works only for views, not for other compiled objects. To determine whether select * statements in other compiled objects need to be revised, review the source text of each compiled object.

Using *dbcc upgrade_object*

Syntax
```
dbcc upgrade_object [ ( dbid | dbname
    [, ['database.[owner].]compiled_object_name' |
     'check' | 'default' | 'procedure' | 'rule' |
     'trigger' | 'view'
    [, 'force' ] ] ) ]
```

where:

- *dbid* - specifies the database ID. If you do not specify *dbid*, all compiled objects in the current database are upgraded.

- *dbname* - specifies the database name. If you do not specify *dbname*, all compiled objects in the current database are upgraded.

- *compiled_object_name* - is the name of a specific compiled object you want to upgrade. If you use the fully qualified name, *dbname* and *database* must match, and you must enclose the fully qualified name in quotes. If the database contains more than one compiled object of the same name, use the fully qualified name. Otherwise, all objects with the same name are parsed, and if no errors are found, upgraded.

- check - upgrades all check constraints and rules. Referential constraints are not compiled objects and do not require upgrading.

- default - upgrades all declarative defaults and the defaults created with the create default command.

- procedure - upgrades all stored procedures.

- rule - upgrades all rules and check constraints.

- trigger - upgrades all triggers.

- view - upgrades all views.

  The keywords check, default, procedure, rule, trigger, and view specify the classes of compiled objects to be upgraded. When you specify a class, all objects in that class, in the specified database, are upgraded, provided that dbcc upgrade_object finds no errors or potential problem areas.

- force - specifies that you want to upgrade the specified object even if it contains a select * clause. Do not use force unless you have confirmed that the select * statement will not return unexpected results. The force option does not upgrade objects that contain reserved words, contain truncated or missing source text, refer to nonexistent temporary tables, or do not match the quoted identifier setting. You must fix these objects before they can be upgraded.

Note If set quoted identifiers is on, use single quotes around the keywords. If set quoted identifiers is off, you can use either double quotes or single quotes.

Examples

```
dbcc upgrade_object
```

Upgrades all compiled objects in the active database.

```
dbcc upgrade_object(listdb, 'procedure')
```

Upgrades all stored procedures in the listdb database. Single quotes are used around procedure because set quoted identifiers is on.

```
dbcc upgrade_object(listdb, "rule")
```

Upgrades all rules and check constraints in the listdb database. Double quotes are used around rule because set quoted identifiers is off.

```
dbcc upgrade_object(listdb, list_proc)
```

Upgrades all stored procedures named list_proc in the listdb database.

```
dbcc upgrade_object(listdb,
  "listdb.jkarrik.list_proc")
```

Upgrades the stored procedure list_proc, which is owned by the login "jkarrik".

```
dbcc upgrade_object(master,
  "listdb.jkarrik.list_proc")
```

Returns an error because the value of *dbname* is master and the value of *database* is listdb. These values must match.

Permissions

Only the Database Owner or a System Administrator can execute dbcc upgrade_object. The Database Owner can upgrade his or her own objects in the database.

Upgraded objects retain the same owner that they had prior to being upgraded.

Increasing the log segment size

You can specify that all compiled objects of a particular class should be upgraded in one execution of dbcc upgrade_object; for example, you can upgrade all triggers by using the trigger keyword. However, even though you use only one dbcc command, the upgrade of each object is recorded in a separate transaction; the old row is deleted from sysprocedures and a new row is written. Therefore, if you run dbcc upgrade_object on a large number of compiled objects, your system may run out of log space. Increase the size of the log segment in the databases in which you plan to run this command, to allow sufficient room to log all the upgrades.

Error reporting

To send all the output from dbcc upgrade_object to the window, a System Administrator can execute dbcc traceon(3604). Sybase recommends that you use this command if you think the output of error messages might overflow the error log.

Using database dumps in upgrades

Upgrading using dump and load

You can load pre-12.5 database dumps and transaction logs and upgrade the databases.

Some issues of which you should be aware:

*   Upgrading requires space for copying data and logging changes to the system tables during the upgrade process. If the source database in the dump was nearly full, the upgrade process might fail due to insufficient space. While this is expected to be uncommon, you can use alter database to extend the free space in the event of insufficient-space errors.

*   After reloading an older dump, run sp_checkreswords from the new installation on the loaded database to check for reserved words.

Upgrading compiled objects in database dumps

When you load a database dump that was created in an earlier version than the current Adaptive Server, you are not required to perform the pre-upgrade tasks before loading the dump. Therefore, you will not receive any notification if the compiled objects in your database dump are missing their source text. After loading a database dump, run sp_checksource to verify the existence of the source text for all compiled objects in the database. Then, you can allow the compiled objects to be upgraded as they are executed, or you can run dbcc upgrade_object to find potential problems and upgrade objects manually.

For information on using sp_checksource, see the *Reference Manual: Procedures*.

Determining whether a compiled object has been upgraded

To determine whether a compiled object has been upgraded, do one of the following:

- Look at the sysprocedures.version column. If the object was upgraded, this column will contain the number 12500.

- If you are upgrading to a 64-bit pointer size in the same version, look at the sysprocedures.status column. It will contain a hexadecimal bit setting of 0x2 to indicate that the object uses 64-bit pointers. If the bit is not set, the object is a 32-bit object, which means the object has not been upgraded.

Removing Adaptive Server

With this version of Adaptive Server, the installation technology has changed from that used for Adaptive Server 12.0 and 12.5. The current installer is based on InstallShield Multi-Platform Edition and uses the built-in uninstall capability. The basic premise of the uninstaller is that only actions that were done by InstallShield are removed during an uninstallation process. This means that files or registry entries created after installation are not removed by the uninstaller and must be removed by the customer after the uninstall is completed. Where possible these actions are identified in this chapter.

Heading	Page
Uninstalling Adaptive Server 12.5.4	95
Removing an existing Adaptive Server	96

Uninstalling Adaptive Server 12.5.4

Note The UninstallDeletePanel wizard prompts you to remove user files in GUI and console uninstalls. However, this option is not available in silent uninstall.

1   Shut down all servers prior to running the uninstall program. See "Starting and Stopping Servers" in the *Configuration Guide*.

2   The uninstall program is located in the following product specific directories:

   •   *uninstall/ASESuite*

   •   *uninstall/PCClient*

   •   *uninstall/SDK*

   •   *uninstall/OpenServer*

   Launch the uninstall program by executing the following steps:

a   Change to the *SYBASE* directory.

b   Source the *SYBASE.csh* or *SYBASE.sh* as appropriate.

c   Execute the uninstall program by entering:

```
./_uninstall/ASESuite/uninstall
```

3   Once the Uninstall Welcome window displays, click Next to continue.

4   The component selection window displays with everything selected. Deselect components by clicking the check box and erasing the check mark. Click Next to continue.

5   A summary window displays what is about to be uninstalled. Click Next to continue.

6   A window displays indicating that the uninstallation process is being performed. There is no progress bar.

> **Note** One or more pop-up displays may occur indicating that a file was modified after it was installed and requesting confirmation that it should be deleted.

7   When the uninstall process has completed a final window displays. Click Finish to exit the uninstall program.

There are five directories that the uninstall program cannot delete. These are *_JVM* (the Java VM, which is needed by the uninstall program) and the directories where the uninstall programs are located. You must delete these manually. The uninstall program does not delete database devices which may be in the *SYBASE* directory or elsewhere. Unless you are planning to reinstall Adaptive Server and re-use the database devices, you should delete them.

Removing an existing Adaptive Server

1   To remove an existing Adaptive Server, from a command line prompt, enter:

```
rm servername.*
```

2   Change to the directory *$SYBASE/$SYBASE_ASE/install* and run the following commands:

```
rm RUN_servername.*
```

```
rm servername.*
```

3   Edit *$SYBASE/interfaces*, to remove all references to the Adaptive Server.

4   If you used operating system files for database devices, remove those.

A P P E N D I X   A   Alternative Installation Methods

Installation on consoles (no-GUI mode)

In cases where there is no graphics display device available, or for some other reason you wish to run the installer without the graphical user interface (GUI) InstallShield can be launched in console or no-GUI mode. When invoking the installer setup program add the -console command line argument. In cases where InstallShield launches automatically, select Cancel to cancel the GUI install and then launch the setup program from a terminal or console. Then:

1   Change the default directory to the CD.

2   Run the setup program with the -console command line argument:

```
setup -console
```

The flow of the installation is identical to a regular GUI installation, except that the display is written to a terminal window and responses are entered using the keyboard.

Silent installation

A silent installation (sometimes referred to as an unattended install) is done by running InstallShield and providing a response file that contains answers to all of InstallShield's questions. There are two methods of generating a response file for InstallShield. The first method is to run InstallShield in GUI mode and record all of the responses into a response file. This response file then be edited with an editor to customize the responses if needed.

The second method is to run InstallShield and have it write out a template through the file in an editor and provide responses for each question. InstallShield is then invoked with the response file and all questions that would be been asked are answered by reading from the response file.

To perform the silent installation using the resource file that you created by running InstallShield through the installation process:

1   Change default directory to the CD.

2   Run the setup program with the -options-record *filename*:

        setup –options-record session1.txt

3   Run through the installation saving all of the responses.

4   Edit the response file to customize it for your needs.

5   Run the setup program with the -option filename and -silent command line arguments. Additionally you must include the -W argument that indicates that you accept the license agreement.

        ./setup -options "/tmp/neals_install.txt" -W
        SybaseLicense.agreeToLicense=true -silent

To use a template response file, skip step three and alter step two above to do the following:

        Setup –options-template <response-file-name>

In either case it is possible to custom configure the servers as well as being able to select a typical install and use the default configuration values.

Except for the absence of the GUI screens all actions of InstallShield are the same and the result of an installation in silent mode is exactly the same as one done in GUI mode with the same responses.

Note The value for `-options`, `-options-template`, and `-options-record` must specify an absolute path.

Configuration using resource files

You can create an Adaptive Server or Backup Server, using values specified in a resource file that defines the attributes for the server.

Resource files are ASCII format template files that contain configuration variables. To use the template files, edit the resource file, replace variables with desired values, and execute the svrbuildres utility. The svrbuildres utility uses the variables to create servers.

Resource files allow you to create servers in batch mode. You can create servers on multiple systems from the same resource file by editing a few values in that file. Within a single resource file, you can create only one server at a time.

The server files must already exist on the computer before the servers can be created. Use procedures in Chapter 2, "Installing Adaptive Server."

To install an Adaptive Server or Backup Server using resource files:

1   Edit a resource file as described in "Configuring from a resource file" on page 101.

2   Execute the srvbuildres utility, using the edited resource file as described in "Using srvbuildres" on page 104.

Configuring from a resource file

You can install Adaptive Server via a resource file. You can use either:

•   The sample resource file provided with the software, or

•   The resource file that is created by the srvbuild utility.

Editing a sample resource file

Sample resource files for creating Adaptive Server and Backup Server are included in your Adaptive Server distribution in:

$SYBASE/$SYBASE_ASE/init/sample_resource_files.

Edit these files to specify the attributes for the servers you want to create.

Note the following:

•   If you use USE-DEFAULT as the Adaptive Server name, the utility substitutes the name of the computer on which it is running for the server name. If you use USE-DEFAULT, do not create Backup Server entries in the resource file. They are created automatically using the default name.

- To create a server with a different name, change the Adaptive Server name and use the new name with the "_BK" extensions for Backup Server. For example, the Backup Server for PIANO should be PIANO_BK.

- The master device size must be at least 30MB.

- The system procedure device and sybsystemprocs database sizes must be at least 132MB.

- The auditing entries in the resource file are not supported and will not be processed by the utility. To enable auditing, see Configuring Adaptive Server Enterprise.

Editing a resource file created by *srvbuild*

You can create a resource file by running the srvbuild utility. A new file containing the values you specified for the Adaptive Server or Backup Server is written to:

$SYBASE/$SYBASE_ASE/init/logs/srvbuildMMDD.VVV-servername.rs.

Edit these files to specify the attributes for the servers you want to create.

Warning! Be sure to edit the resource file generated by srvbuild to change attributes, such as device names, that may be different on another system.

Resource file attributes for Adaptive Server

Table A-1 shows the Adaptive Server resource file attributes, their default values, and other options.

The attributes in bold type are *required*.

All alpha values are case-sensitive.

The prefix of the attribute name varies, depending on whether the resource file was created by srvbuild or by sybinit (from a pre-11.5 SQL Server). The prefix is ignored by srvbuildres when processing the resource file.

The attribute names in Table A-1 are the attribute names provided in the sample resource file included in your Adaptive Server distribution.

Table A-1: Resource file attributes for Adaptive Server

Attribute	Default value [other options]
sybinit.release_directory	The value of *$SYBASE* at your site

Attribute	Default value [other options]
sybinit.product	sqlsrv
sqlsrv.server_name	*server_name*
sqlsrv.new_config	yes (required value)
sqlsrv.do_add_server (to *interfaces* file)	yes [no]
sqlsrv.network_protocol_list	tcp [spx]
sqlsrv.network_hostname_list	*hostname*
sqlsrv.network_port_list	*port_number*
sqlsrv.master_device_physical_name	*path_and_name_of_master_device*
sqlsrv.master_device_size	30 (MB)
sqlsrv.master_database_size	6 (MB)
sqlsrv.errorlog	*$SYBASE/$SYBASE_ASE/install/server_name.log*
sqlsrv.do_upgrade	no (required value)
sqlsrv.sybsystemprocs_device_physical_name	*path_and_name_of_sybsystemprocs_device*
sqlsrv.sybsystemprocs_device_size	132 (MB)
sqlsrv.sybsystemprocs_database_size	100 (MB)
sqlsrv.sybsytemdb_device_physical_name	*path_and_name_of_ sybsystemdb_device*
sqlsrv.sybsystemdb_device_size	5 (MB)
sqlsrv.sybsystemdb_database_size	5 (MB)
sqlsrv.default_backup_server	*server_name*_BS

Resource file attributes for Backup Server

Table A-2 shows the Backup Server resource file attributes, their default values, and other options.

The attributes in bold type are *required*.

All values are case-sensitive.

The prefix of the attribute name varies, depending on whether the resource file was created by srvbuild or by sybinit (from a pre-11.5 Adaptive Server). The prefix is ignored by srvbuildres when processing the resource file.

The attribute names in Table A-2 are the attribute names provided in the sample resource file included in your Adaptive Server distribution.

Table A-2: Resource file attributes for Backup Server

Attribute	Default value [other options]
sybinit.release_directory	$SYBASE
sybinit.product	bsrv
bsrv.server_name	*server_name*_back
bsrv.do_add_backup_server (to *interfaces* file)	yes [no]
bsrv.network_protocol_list	tcp [spx]
bserv.network_hostname_list	*hostname*
bsrv.network_port_list	*port_number*
bsrv.language	us_english [chinese, french, german, japanese, spanish]
bsrv.character_set	iso_1[iso_1, cp850, cp437, deckanji, ascii_8, eucgb, eucjis, mac, roman8, sjis, utf8]
bsrv.tape_config_file	*$SYBASE/$SYBASE_ASE/ backup_tape.cfg*
bsrv.errorlog	*$SYBASE/$SYBASE_ASE/ server_name_back.log*

Using *srvbuildres*

To execute srvbuildres at the UNIX prompt enter:

```
$SYBASE/$SYBASE_ASE/bin/srvbuildres -r resource_file
```

where *resource_file* specifies the resource file containing the attributes that describe the server to build.

Note Unlike srvbuild, the srvbuildres utility does not require any X libraries.

When you have completed resource file installation, go to Chapter 3, "Post-Installation Tasks."

Creating Raw Partitions

Overview

Adaptive Server 12.5 and later allows you to create and mount its database devices on raw bound devices for raw disk I/O. Raw disk I/O has performance advantages since it enables direct memory access from user address space to the physical sectors on the disk, omitting needless memory copy operations from the user address space to the kernel buffers.

Raw disk I/O also assumes that logical and physical I/O are simultaneous, and writes are guaranteed to flush to the disk when the system write() call returns.

When preparing a raw partition device, follow these guidelines:

- Do not initialize a database device on the partition that contains your Sybase installation software. Doing so destroys all existing files on that partition.

- A raw partition designated for use by Sybase cannot be mounted for use by the operating system for any other purpose, such as for file systems or swap space.

- After a Sybase configuration utility or the disk init command has initialized a portion of a partition as a database device, the entire partition cannot be used for any other purpose. Any space left on the partition beyond the size specified for the device becomes inaccessible, unless you drop and re-create the device.

- For best performance, place the Sybase software and all disk devices, including the master device, on the same machine.

- To avoid any possibility of using a partition that contains the partition map, do not use cylinder 0.

- Place the database device on a character device, because the Adaptive Server recovery system needs unbuffered system I/O.

To determine whether a device is a block device or a character device, run this command on the */dev/raw* directory:

```
ls -l
```

"b" (block) or "c" (character) appears in the left-most column.

❖ **Choosing a raw partition**

1 Determine which raw partitions are available.

2 Determine the sizes of the raw partitions.

3 From the list of available raw partitions, select a raw partition for each device.

4 Verify with the operating system administrator that the partition you have chosen is available.

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

Note For more information on choosing a raw partition, see your operating system documentation.

Creating the partitions

Specific system administration is required before raw devices can be enabled and used. The available tools to configure devices depend on the distribution.

Physical disk space needs to be allocated in partitions on the disks where you want to set up raw devices. The physical I/O subsystem can be on either SCSI or EIDE devices.

Note Creating the partitions can be done with the Linux default fdisk(8) utility. You must have "root" privileges to use the command fdisk. Refer to the fdisk(8) man pages for a complete description of the command.

This example shows how to set up partitions as raw devices, on four SCSI disks in the system—sda, sdb, sdc, and sdd.

• Start fdisk on */dev/sdd:*
```
# fdisk /dev/sdd
```

The system returns:

```
The number of cylinders for this disk is set to 8683
```

```
....
Command (m for help):
```

• Enter "p" to print the current partition lay-out. The output is:

```
Disk /dev/sdd: 64 heads, 32 sectors, 8683 cylinders
Units=cylinders of 2049 * 512 bytes

Device Boot   Start    End     Blocks   Id  System
/dev/sdd1       1     7499    7678960   83  Linux
/dev/sdd2     7500     8012    525312   82  Linux swap
/dev/sdd4     8013     8683    687104    5  Extended
```

This sample shows the extended partition (sdd4) has 687104 free blocks, starting from 8013 and ending at 8683. The remaining partitions can be assigned later.

The next example assigns an additional partition for raw bound disk I/O:

1   Use the n command to create a new partition, and enter "l" at this prompt for logical.

```
Command (m for help):n
Command action
l    logical (5 or over)
p    primary partition (1-4)
```

2   Accept the default by pressing Enter on the keyboard when you are prompted:

```
First cylinder (8013-8683, default 8013):
```

3   Accept the default by pressing Enter on the keyboard when you are prompted:

```
Last cylinder or +size or +sizeM or +sizeK (8013-
8683, default 8683): 8269
```

4   Use the t command, enter "5" at this prompt:

```
Partition number (1-8):5
```

5   Enter "60" at this prompt:

```
Hex code (type L to list codes): 60
```

6   The output is:

```
Changed system type of partition 5 to 60 (Unknown)
```

7   Repeat the above steps to create four partitions for raw device I/O.

Verify the setup using p to print the full partition table before writing it out. Make sure that there are no overlapping partitions and the type for the unassigned partitions is Unknown type 60.

The partition table can now be written to disk and you can quit the fdisk(8) utility.

Before continuing, restart your system to verify that the kernel running is in sync with the new disk partition layout. This is reported during system start when probing for disk spindles.

```
# /sbin/reboot
```

Red Hat Raw device administration

Red Hat Enterprise Linux is fully equipped with the administration tools to set up raw devices and administrate them during system start. To set up a Red Hat Enterprise Linux system the partitions are administrated in the file */etc/sysconfig/rawdevices*.

This is a plain text file contains comments and examples for possible configurations, as follows:

```
# raw device bindings

# format: rawdev major minor
#         rawdev blockdev
# example: /dev/raw/raw1 /dev/sda1
#          /dev/raw/raw2 8 5
/dev/raw/raw1 /dev/sdd1
/dev/raw/raw2 /dev/sdd2
/dev/raw/raw3 /dev/sdd3
/dev/raw/raw4 /dev/sdd4
```

1   Once created, you need to bind the raw devices. This can be done by starting them from */etc/rc.d/init.d/rawdevices*.

```
[root@legolas init.d]# cd /etc/rc.d/init.d

[root@legolas init.d]# sh rawdevices start

Assigning devices:

/dev/raw/raw1  -->   /dev/sdd5

     /dev/raw/raw1:    bound to major 3, minor 5
```

```
/dev/raw/raw2  -->   /dev/sdd6

     /dev/raw/raw2:     bound to major 3, minor 6
/dev/raw/raw3  -->   /dev/sdd7

     /dev/raw/raw3:     bound to major 3, minor 7
/dev/raw/raw4  -->   /dev/sdd8

     /dev/raw/raw4:     bound to major 3, minor 8
done
```

2   To guarantee that the raw device binding occurs during any restart, use the chkconfig(8) utility.

```
# /sbin/chkconfig rawdevices on
```

SuSE raw device administration

The raw disk partitions are administered in the */etc/raw* file. This is a plain text file containing comments and examples for possible configurations:

```
# /etc/raw
#
# sample configuration to bind raw devices
# to block devices
#
# The format of this file is:
# raw<N>:<blockdev>
#
# example:
# ---------
# raw1:hdb1
#
# this means: bind /dev/raw/raw1 to /dev/hdb1
#
# ...
raw1:sda7
raw2:sda8
raw3:sda9
```

Once created, you need to bind the raw devices. You do this by starting them with the script */etc/init.d/raw*:

```
# cd /etc/init.d
# sh raw start
```

```
bind /dev/raw/raw1 to /dev/sdb1... done
bind /dev/raw/raw2 to /dev/sdb2... done
bind /dev/raw/raw3 to /dev/sdb3... done
...
```

Use the chkconfig(8) utility to guarantee that the raw device binding occurs during any restart:

```
# /sbin/chkconfig raw on
```

Accessing raw devices from Adaptive Sever

Once the partitions are created and the devices are bound for raw disk I/O, Adaptive Server is able to use them. If Adaptive Server is running as user sybase, then apply read, write, and owner permissions to the */dev/raw/raw#* device entries and the raw bound control device */dev/rawctl* for user sybase.

See the chown(1), chgrp(1), and chmod(1) commands to apply correct permissions.

*   To verify your settings, query the device binding using the raw command, enter:

    ```
    $ raw -qa
    ```

    The output should be:

    ```
    /dev/raw/raw1:  bound to major 3, minor 5

    /dev/raw/raw2:  bound to major 3, minor 6

    /dev/raw/raw3:  bound to major 3, minor 7

    /dev/raw/raw4:  bound to major 3, minor 8
    ```

*   To verify your settings, query the device binding using the permissions command, enter:

    ```
    <username@servername.sybase.com:4> raw -qa
    ```

    The output should be:

    ```
    Cannot open master raw device '/dev/rawctl'
    (Permission denied)
    ```

Using the raw devices, Adaptive Server and the installation and configuration utility, srvbuild, detects and presents their size automatically. Just enter the absolute path to the raw device when creating the master, sybsystemprocs, sybtempdb, or any other device.

Adaptive Server Enterprise

APPENDIX C   Upgrading Servers with Replicated Databases

Preparing to upgrade servers with replicated databases

After you have upgraded, you will no longer be able to scan any part of the transaction log that existed before the upgrade, so you must follow the following process if your server contains replicated primary databases (this includes replicated RSSDs). This procedure will help to ensure that all replicated data from a replicated database has made it safely to the replicate database.

Warning! It is not sufficient to simply get the replicated data into the Replication inbound queue, you cannot rebuild the inbound queue after the upgrade.

The procedures described here do not upgrade Replication Server itself. For information on upgrading Replication Server, see your Replication Server documentation.

The database upgrade procedure consists of:

* Suspending transaction processing and replication activities.

* Draining transaction logs for primary databases.

* Draining the Replication Server System Database (RSSD) log.

* Disabling the log truncation point.

After upgrading to version 12.5.x, complete the post-upgrade tasks to reenable database replications functions.

For more information, see the *Replication Server Reference Manual* and the *Replication Server System Administration Guide*.

 Warning! As a safeguard, perform a dump database and a dump transaction before executing the procedures in the following sections.

To determine whether your existing server contains replicated databases:

1   Use isql to connect to the Server you are upgrading.

2   Run the following command in each database (including system databases):

```
1> dbcc gettrunc
2> go
```

3   If the command returns "1" for "ltm_trunc_state" in any database, replication is enabled in that database.

Suspending transaction processing and replication activities

To suspend replication of and transaction activity in the databases:

1   Verify that subscriptions that are being created by the create subscription command, with primary data in the databases being upgraded, have reached the "valid" state at the primary Replication Server before you begin the upgrade. Use the check subscription command to find the "valid" state.

Defer the upgrade process while the subscriptions are being created so that Replication Server does not interfere by attempting to access the database being upgraded.

Make sure that no users create subscriptions for data in the database you are upgrading until the upgrade procedure is finished.

2   Execute rs_helproute in each Replication Server System Database (RSSD) being upgraded.

The status of all existing routes should be "Active." If any routes are not active, resolve them before continuing. See the Replication Server documentation for help in diagnosing and fixing the problem with the route, and then go to step 3.

3   Shut down all applications that use the databases you are upgrading.

4   Use the admin who command in Replication Server to find the existing
    Data Server Interface (DSI) connections to the data server being upgraded.

5   Suspend all DSI connections to the non-RSSD databases you are
    upgrading by entering the following command in Replication Server for
    each database:

    ```
    1> suspend connection to dataserver.database
    2> go
    ```

6   Leave the DSI connections to the RSSD databases running.

Draining the transaction logs for primary databases

For each primary database you are upgrading, ensure that Replication Server
completely processes the pre-upgrade log.

To drain the transaction logs:

1   Wait for all remaining transactions to be replicated.

2   Run the following Replication Server command:

    ```
    1> admin who, sqm
    2> go
    ```

3   Find the entry that corresponds to the inbound queue for this database by
    looking for the Info field for the queue_number:queue_type entry. For an
    inbound queue, the queue type is 1. Note the Last Seg.Block entry for the
    queue.

4   Open a queue dump file by executing the following Replication Server
    command, where *file_name* is the name of the file to which you will dump
    the queue:

    ```
    1> sysadmin dump_file, "file_name"
    2> go
    ```

5   Use isql to update one row in a single replicated table in the primary
    database:

    ```
    1> update table set column = column
    2> where key = unique_value
    3> go
    ```

The update command helps track whether all modifications to the replicated database have been sent to the Replication Server.

> **Note** In Replication Server 10.1 or later, choose a table that does not use the replicate minimal columns clause, or use the alter replication definition command...replicate all columns command to change the replication definition before updating the row. If you alter the replication definition, be sure to change it back after you complete this upgrade procedure.

6 In the primary Replication Server, execute the admin who, sqm command until the last segment:block entry for the inbound queue changes.

7 Execute the following Replication Server command to dump the last block of the inbound queue to the dump file you created in step 3:

```
1> sysadmin dump_queue, queue_number,
2> queue_type, last_seg, block, 1
3> go
```

Use the *queue_number*, *queue_type*, *last_seg*, and *block* values found in the output of the last admin who, sqm command.

8 Use Notepad or another text editor to examine the dump file to make sure it contains the transaction that corresponds to the update you performed in step 4.

9 Repeat steps 5–7 until the transaction that corresponds to the update is in the dump file.

10 Log in to the Replication Server and suspend the Log Transfer connection from that database:

```
1> suspend log transfer from server.database
2> go
```

11 If you are using Rep Agent, log into the Adaptive Server, and stop the Rep Agent:

```
1> use database
2> go

1> sp_stop_rep_agent database
2> go
```

12 If you are using LTM, shut down the LTM.

After draining the transaction logs, do not allow any other activity in the databases. If activity does occur, you will need to redrain the logs.

Draining the RSSD transaction log

If the Replication Server has routes to other Replication Servers, you must ensure that Replication Server processes all transactions in the RSSD transaction log before you upgrade the databases.

To see whether the transaction log has been processed completely, create a replication definition in the primary Replication Server and then watch for it to appear in the replicate Replication Server's RSSD. When the replication definition is in the replicate RSSD, you can assume that the log is processed fully.

To ensure that the RSSD log is processed:

1 Log in to the primary Replication Server and create a temporary replication definition:

```
1> create replication definition rep_def_name
2> with primary at dataserver.database
3> (column_a int)
4> primary key (column_a)
5> go
```

The data server and database names must be valid, but the replication definition does not have to reference an actual table.

2 Log in to the replicate RSSD (not the primary RSSD) and execute the following query to find out if the replication definition has arrived from the primary RSSD:

```
1> select * from rs_objects
2> where objname = "rep_def_name"
3> go
```

If this select statement returns rows, the last replication definition created in step 1 has been sent successfully to the replicate RSSD. This means that the transaction log has been drained.

3 Log in to the replicate Replication Server and suspend the Log Transfer connection from the primary RSSD:

```
1> suspend log transfer from server.database
2> go
```

4 If you are using Rep Agent, log in to the Adaptive Server, and stop the Rep Agent:

```
1> use database
2> go
```

```
1> sp_stop_rep_agent database
2> go
```

5   If you are using LTM, shut down the LTM.

Disabling the secondary truncation point

When you upgrade a primary database, the Rep Agent or the Log Transfer Manager (LTM) must not be running, and the secondary truncation point should be turned off for the duration of the upgrade. The Rep Agent or Log Transfer Manager should already be shut down (from the previous steps).

For each primary database and replicated RSSD, disable the secondary truncation point:

1   If this is a replicated RSSD, log in to the Replication Server of the RSSD, and issue:

```
1> sysadmin hibernate_on, 'Replication Server'
2> go
```

2   Disable the secondary truncation point in the Adaptive Server database that is being upgraded by issuing:

```
1> use database
2> go

1> dbcc settrunc('ltm', 'ignore')
2> go
```

If the dbcc settrunc command fails, make sure that the Rep Agent or LTM are not running. When the Rep Agent and LTM are disabled, repeat this step.

APPENDIX D   **Post-Upgrade Tasks**

Restoring functionality in Adaptive Server

To reset or enable options or functionality you had to change before the upgrade:

1   Reset the configuration parameters.

   If you changed any configuration parameters before upgrading, use sp_configure to set them back to their previous values.

2   Use sp_dboption to reset any database options you disabled before upgrading.

3   After you complete the upgrade, but before you use the upgraded Adaptive Server, verify that all scripts developed at your site point to Adaptive Server 12.5.x.

4   Verify procedure cache allocation after upgrading.

   This section is for information only. Adaptive Server ensures that procedure cache size after upgrade is the same as before upgrade, unless the original size was less than the default value.

   If you are upgrading from a version of Adaptive Server that is earlier than version 12.5.x, procedure cache percent has been changed to procedure cache size. Adaptive Server now allocates memory dynamically and determines the size of the procedure and data caches in terms of megabytes. During the upgrade process, Adaptive Server converts the procedure cache percent to an absolute value and sets procedure cache size based on this value.

For example, if procedure cache percent is configured as 20 percent of 200MB of memory, then the absolute value of procedure cache percent is 40MB. During the upgrade process, Adaptive Server converts procedure cache percent at 20 percent to procedure cache size at 40MB. If the value for procedure cache size is less than the default value, Adaptive Server sets procedure cache size to the default value. For example, if procedure cache percent is set to 10 percent in the example above, and the absolute value of 10 percent is less than the default value of procedure cache size, then procedure cache size is set to the default value.

With the introduction of dynamically reconfigured memory configuration parameters in Adaptive Server 12.5.x, an increase in Adaptive Server's memory use does not decrease the size of the procedure cache or the data cache. That is, if your default data cache is set to 40MB, and you increase your procedure cache, the default data cache is still 40MB.

For more information about sp_configure, see the *Reference Manual*.

5   Check the procedure cache requirements.

Stored procedures, triggers, and other compiled objects require more memory to run than older versions. The memory required to run a stored procedure increased by 20 percent between versions 10.x and 11.5. Adaptive Server 12.5.x needs approximately 4 percent more procedure cache than version 11.5 for the server to maintain the same performance.

You may increase the procedure cache size during runtime, using sp_configure. You can use the verify option in sp_configure to verify any changes you make to the configuration file without having to restart Adaptive Server. The syntax is:

```
sp_configure "configuration file", 0, "verify",
"full_path_to_file"
```

For example:

```
sp_configure "configuration file", 0, "verify",
"/work2/Sybase/ASE125.cfg"
```

For more information about sp_configure and sp_sysmon, see the *Reference Manual* and the *Performance and Tuning Guide*. For more information about configuring memory, see the *System Administration Guide*.

6   Verify data cache allocation after upgrading. This section is for information only. Adaptive Server ensures that all the data cache sizes after upgrading are the same as before upgrade.

In pre-12.5 versions of Adaptive Server, the default data cache available depended on the amount of memory available to Adaptive Server. If the default data cache was set to 2MB, Adaptive Server allocated 2MB to the default data cache. If, after memory was allocated to all user-defined data caches and to the procedure cache, there was more memory available, that memory was allocated to the default data cache. However, if there was not enough memory available for Adaptive Server to set up the user-defined caches and the procedure cache, then memory was taken from the default data cache regardless of the designated configuration parameter. Therefore, it was possible to set default data cache to the default value and have all remaining available memory allocated to the default data cache.

Adaptive Server 12.5.x treats the default data cache size as an absolute value and sets it in the *config* file. The string DEFAULT in the *config* file has a different meaning for default data cache size in Adaptive Server 12.5.x.

In pre-12.5 Adaptive Server, it meant all the memory left after allocating memory for other configurations, procedures, and user-defined named caches. In Adaptive Server 12.5.x, it means a default value of 8MB.

During the upgrade process, Adaptive Server ensures that the default data cache size remains the same. Therefore, during the pre-upgrade process, the size of the default data cache is obtained and written to the configuration file as an absolute value, not as "DEFAULT." This enables Adaptive Server 12.5.x to have the same default data cache size as before the upgrade. If this size is less than the default size of default data cache of 8MB, then Adaptive Server 12.5.x allocates a default data cache of size 8MB.

7   If you unmirrored devices, remirror them, using the disk remirror command.

8   If you used two-phase commit in 11.9.x, run the script to install the two-phase commit tables:

```
isql -Usa -Psa_password -Sserver_name
-i$SYBASE/$SYBASE_ASE/scripts/installcommit
```

You do not need to run this script if you do not use two-phase commit.

Re-enabling Replication Server

If you disabled replication before the upgrade, you must re-enable replication. To do so, first remove any older format log records from your database, and then reenable replication.

Removing old log records

Use the dump tran command to dump the database and transaction logs to remove the older format log records from your database. This prevents Replication Server from accessing the pre-upgrade portion of the transaction logs.

For example, to dump the sales database:

```
1> use master
2> go
1> dump database sales to dumpdev
2> go
```

Reenabling replication

Follow the steps in this section to reenable replication after upgrading the primary databases and primary RSSDs.

For each primary database and RSSD:

1   Start Adaptive Server if it is not already running.

2   Log in to the server.

3   Clear the locator for the database by executing the following command in the RSSD for this database:

```
1> use RSSD
2> go
3> rs_zeroltm dataserver, database
4> go
```

4   Enable the truncation point for the database.

```
1> use database
2> go
1> dbcc settrunc("ltm", "valid")
2> go
```

5   Restart the Replication Servers and LTMs.

6   Resume the DSI connections that were suspended before the upgrade by executing the following Replication Server command for each suspended database:

```
1> resume connection to dataserver.database
2> go
```

The replication system is now ready for Adaptive Server 12.5.x, and applications can resume.

If you installed any Sybase client products, such as Open Client, use the dsedit utility to edit your *interfaces* file and to specify the servers to which you want to connect.

For more information about establishing client/server connections, see the *Open Client Configuration Guide* or Chapter 4, "Installing Sybase PC-Client Products."

Restoring replication after upgrade

Restore the replication functionality:

1   Log in to the RSSD for each replicated primary, and for each replicated RSSD, and issue:

```
1> use RSSD_name
2> go

1> rs_zeroltm dataserver, database
2> go
```

2   Log in to each replicated primary, and replicated RSSD and issue:

```
1> use database
2> go

1> dbcc settrunc ('ltm', 'valid')
2> go
```

3   Restart the Replication Server.

4   If the database is also used as a RSSD, resume the Replication Server connection to the RSSD by issuing the following command to the Replication Server:

```
1> sysadmin hibernate_off, 'Replication Server'
2> go
```

Make sure to specify the same string that was specified during the 'hibernate_on' command.

5   Log in to the Replication Server, and resume the Log Transfer connection for each replicated primary and each replicated RSSD:

```
1> resume log transfer from server.database
2> go
```

If this is a replicated RSSD, you must log in to the replicate Replication Server.

6   If you are using Rep Agent, log in to the Adaptive Server and restart the Rep Agent:

```
1> use database
2> go

1> sp_start_rep_agent database
2> go
```

7   If you are using LTM, restart the LTM.

Reenabling auditing

Reenable all auditing options for stored procedures using the audit options you recorded during pre-installation. Reenter the audit options using sp_audit.

Adaptive Server 12.5.x and later includes the following global audit options for sp_audit:

•   security

•   dbcc

The functionality provided by these obsolete sp_configure "auditing", 0 options is now covered by the sp_audit security global audit option:

•   server boot

•   role toggle

These options are turned off by default and are not turned on even if you used sp_configure "auditing", 0 to turn on all global audit options in a version earlier than 11.5.

To turn on these options, use:

```
1> sp_audit security "on"
2> go
1> sp_audit "dbcc" "on"
2> go
```

For more information, see the *Reference Manual: Procedures*.

If any of these options were turned off before the upgrade, reset the security option to on to achieve the same auditing actions.

Adaptive Server Enterprise

APPENDIX E   **Using sybsystemprocs**

> **Note** sysprocsdev is the default system name for this device. However, it is frequently referred to as the sybsystemprocs device, as in the Adaptive Server attribute window, since it stores the sybsystemprocs database.

Verify that the sybsystemprocs database is large enough. For an upgrade, the recommended minimum size for sybsystemprocs is the larger of 132MB, or enough free space to accommodate the existing sybsystemprocs database, and the largest catalog that is to be upgraded, plus an additional 10 percent of the largest catalog's size. The additional 10 percent is for logging upgrade changes.

You may need more space if you are adding user-defined stored procedures. 120MBs accommodate additional internal data structures, but does not account for the possibility of a large number of user-defined system procedures.

If your sybsystemprocs database does not meet these requirements and you have enough room on the device to expand the database to the required size, use the alter database command to increase the database size.

Use sp_helpdb to determine the size of the sybsystemprocs database:

```
1> sp_helpdb sybsystemprocs
2> go
```

Use sp_helpdevice to determine the size of the sysprocsdev device:

```
1> sp_helpdevice sysprocdev
2> go
```

If the db_size setting is less than the required minimum, you must increase the size of sysprocdev.

Increasing the size of the *sybsystemprocs* database

If your current sybsystemprocs database does not have the minimum space required, there are two options for creating a new database with sufficient space for the upgrade:

- You can drop the old database and the device and create a new sysprocsdev device, or

- You can leave the database and old device alone and add a new device large enough to hold the additional megabytes, and alter the sybsystemprocs onto the new device. Sybase recommends that you expand the current device.

❖ **Enlarging the sybsystemprocs database**

1 If you do not have a current backup, create one.

2 In isql, use alter database to increase the size of the sybsystemprocs database. For example:

```
1> use master
2> go
1> alter database sybsystemprocs on sysprocsdev=40
2> go
```

In this example, "sysprocsdev" is the logical name of the existing system procedures device and "40" is the number of megabytes of space to add.

If the system procedures device is too small, you may receive a message similar to the following when you try to increase the size of the sybsystemprocs database:

```
Could not find enough space on disks to extend
database sybsystemprocs
```

If there is space available on another device, you can expand sybsystemprocs to a second device, or initialize another device that is large enough. For instructions on creating a larger sybsystemprocs device, see "Increasing device and database capacity for system procedures" on page 129.

3 To verify that Adaptive Server has allocated more space to sybsystemprocs, issue:

```
1> sp_helpdb sybsystemprocs
2> go
```

When the system procedures database is large enough to accommodate the increased size of the sybsystemprocs database, continue with the other pre-upgrade tasks.

Increasing device and database capacity for system procedures

If you cannot fit the enlarged sybsystemprocs database on the system procedures device, increase the size of the device and create a new database.

This procedure involves dropping the database. For more information on drop database, see the *Reference Manual: Commands*.

Warning! This procedure removes all stored procedures you have created at your site. Before you begin, save your local stored procedures using the defncopy utility. See the *Utility Guide* for more information.

❖ **Creating a larger system procedures device (*sysprocsdev*)**

1   Determine which device or devices you need to remove.

Warning! Do *not* remove any device that is in use by database other than sybsystemprocs, or you will destroy that database.

```
select d.name, d.phyname
from sysdevices d, sysusages u
where u.vstart between d.low and d.high
and u.dbid = db_id("sybsystemprocs")
and d.status & 2 = 2
and not exists (select vstart
    from sysusages u2
    where u2.dbid != u.dbid
    and u2.vstart between d.low and d.high)
```

Where *d.name* is the list of devices to remove from sysdevices and *d.phyname* is the list of files to remove from your computer.

The not exists clause in this query excludes any devices that are used both by sybsystemprocs and other databases.

Note the names of the devices; you will need them in the steps.

2   Drop sybsystemprocs:

```
use master
go
drop database sybsystemprocs
go
```

3   Remove the device or devices using sp_configure, enter the following for
The where clause contains the list of device names returned by the query
in step 1:

```
sp_configure "allow updates", 1
go

delete sysdevices
where name in ("devname1", "devname2", ...)
go

sp_configure "allow updates", 0
go
```

> **Note** Each device name must have quotes. For example, "*devname1*",
> "*devname2*", and so on.

If any of the named devices are OS files rather than raw partitions, use the
appropriate OS commands to remove those files.

4   Remove all files for the list of *d.phyname* returned in step 1.

> **Note** File names cannot be complete path names. If they are relative paths,
> they are relative to the directory from which your server was started.

5   Find another existing device that meets the requirements for additional
free space, or use a disk init command similar to the following to create an
additional device for sybsystemprocs, where */sybase/work/* is the full,
absolute path to your system procedures device.:

```
1> use master
2> go

1> disk init
2> name = "sysprocsdev",
3> physname = "/sybase/work/sysproc.dat",
4> size = 51200
```

```
5> go
```

Note Server versions 12.0.x and later accept but do not require
`"vdevno=number"`. In server versions earlier than 12.0.x, the number for
vdevno must be available. For information about determining whether
vdevno is available, see the *System Administration Guide*.

The size you provide should be the number of megabytes of space needed
for the device multiplied by 512. disk init requires the size to be specified
in 2K pages. In this example, the size is 112MB (112 x 512 = 57344). For
more information on disk init, see the *Reference Manual: Commands*.

6   Create a sybsystemprocs database of the appropriate size on that device,
    for example:

```
1> create database sybsystemprocs on sysprocsdev =
112
2> go
```

7   Run the installmaster script in the *old* Sybase installation directory, as
    follows:

```
isql -Usa -Ppassword -Sserver_name -i/old_dir/
scripts/installmaster -oinstallmaster.out
```

Adaptive Server Enterprise

Backup Server tape device support

Backup Server tape device support

Linux supports multiple tape device strategies and hardware. Backup Server supports only physical tape backup hardware which works per the "mtio" software layer interface.

This includes all generic SCSI-1/-2 tape hardware. Before configuring the backup devices within Adaptive Server you must understand and configure the system tape initialization properly.

Backup Server supports variable block and async tape storage. Log in as "root" to enable these at system start-up. A configuration file */etc/stinit.def* should be in place and a command should be run to force the generic st(8) interface into the desired mode.

For any Backup Server configured tape unit set the below:

```
noblklimits=0 # Force variable block mode.
can-bsr # The unit does handle back space records.
async-writes=0 # Desired 0 - No async tape operations (adviced).
async-writes=1 # Desired 1 - Async tape operations are enabled.
two-fms=1 # Write two (2) end-of-file markers.
buffering=0 # Disable buffering - must if async = 1.
```

This example configures a DDS-[1234] 4mm tape unit:

```
[/etc/stinit.def]
# A compressing DAT (DDS-1-DC or DDS-[234])
manufacturer=HP model = "C1533A" {
scsi2logical=1 can-bsr noblklimits=0 async-writes=0
two-fms=1 buffering=0 read-ahead=0
mode1 blocksize=0 compression=1
mode2 blocksize=1024 compression=1
mode3 blocksize=0 compression=0
mode4 blocksize=1024 compression=0
}
```

Tape units are first accessible per their entries under */dev*. Rewind devices are required. The first unit presents itself as nst0; thenext as nst1,and so on.

You should understand device mappings as described within the man-page towards st(8)reference page.

By default, Adaptive Server configures two backup devices within sysdevices named:

- logical device 'tapedump1' towards physical device '/dev/nst0'.

- logical device 'tapedump2' towards physical device '/dev/nst1'.

Index

A

Adaptive Server
   described   7
   installing multiple   100
   installing via resource file   100
   language, changing   22
   more information about   22
   RAM required by   12
   System Administrator   29
   verifying environment for upgrade   48
Administrator
   Sybase System   14
alter database command   127
applications
   changing after renaming objects   49
   testing current   49
auditing
   feature   124
authorization code
   troubleshooting   84

B

Backup Server
   installing via resource file   100
   more information about   22
batch installation of servers   100

C

character devices
   located in /dev directory   105
commands
   alter database   127
   showserver   27
conflicting object names   50
connection problems

   analyzing   14
connectivity   42
conventions
   used in this manual   xii
current applications
   testing   49

D

database devices   30
   raw partitions for, preparing   105
databases
   image data for sample   31
   interpubs   32
   jpubs   32
   optional   30
   sample   33
default
   network protocols   41
directories
   installation   36
   specifying new for installation   48
disk init command   105
drivers
   ODBC   36
dscp utility   41
dsedit utility   41, 123

E

Enterprise Edition package   5
environment variables
   PATH   35
   TEMP   36
error log files
   troubleshooting and   80
extended stored procedures (ESPs)   22

Adaptive Server Enterprise

Adaptive Server Enterprise