SYBASE®

Installation Guide

Adaptive Server® Enterprise Cluster Edition

15.0.1 ESD #1

[Linux and Sun Solaris]

DOCUMENT ID: DC00760-01-1501-05

LAST REVISED: April 2009

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

Contents

About This Book

The installation guide is for the Cluster Edition of Adaptive Server Enterprise, and provides:

- An overview of the Sybase® Adaptive Server® Enterprise Cluster Edition installation infrastructure.
- Instructions for installing and upgrading Adaptive Server, and installing Backup Server, Monitor Server, XP Server™, Job Scheduler, 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.
- Instructions on how to install Adaptive Server with SySAM 2.0.

Audience

This guide is 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 these chapters:

- Chapter 1, "Overview," provides an overview of Adaptive Server Cluster Edition. It also provides product descriptions and directory layout information.
- Chapter 2, "Installing the Software," describes pre-installation procedures, SySAM related tasks, and how to install server components.
- Chapter 3, "Creating and Starting the Cluster," describes how to configure and start a Sybase shared-disk cluster.
- Chapter 4, "Post-installation tasks," describes how to configure
- Chapter 5, "Installing PC-Client," describes how to install Adaptive Server client products like the Adaptive Server plug-in to Sybase Central.
- Chapter 6, "IO Fencing," describes IO fencing, and how to enable and set up the feature.

- Chapter 7, “Upgrading from an Earlier Version,” describes how to upgrade an existing Adaptive Server to the current version.
- Chapter 8, “Downgrading from Adaptive Server 15.0.2,” describes how to downgrade from Adaptive Server Enterprise 15.0.2.
- Chapter 9, “Downgrading Adaptive Server 12.5.4,”describes how to downgrade from Adaptive Server Enterprise 12.5.4.
- Chapter 10, “Troubleshooting SySAM,” provides information about and solutions to SySAM-related errors.
- Chapter 11, “Troubleshooting,” provides installation error messages and possible solutions to installation and configuration problems.
- Chapter 12, “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, “Manually Setting Up and Managing the Cluster,” describes how to manually configure and manage a shared-disk cluster.
- Appendix C, “A sybcluster Sample Session,” provides a sample sybcluster session.
- Appendix D, “Creating Raw Partitions in Linux,” describes how to upgrade servers with replicated databases.
- Appendix E, “Upgrading Servers with Replicated Databases,” describes how to create partitions, bind partitions for raw disk I/O, and access raw devices from Adaptive Server.
- Appendix F, “Using sybsystemprocs,” describes how to use the sybsystemprocs database.

Related documents

The Sybase Adaptive Server Enterprise documentation set consists of:

- 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 – describes installation, upgrade, and preliminary configuration procedures for all Adaptive Server and related Sybase products.

- *Sybase Software Asset Management Users Guide* – describes asset management configuration concepts and tasks.
- *Users Guide* – describes the new features in Adaptive Server version 15.0, the system changes added to support those features, and changes that may affect your existing applications.
- *Component Integration Services Users Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server Users Guide* – describes how to use Historical Server to obtain performance information for SQL Server and Adaptive Server.
- *Job Scheduler Users 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 Users 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* – datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
  - *Commands* – commands.
  - *Procedures* – system procedures, catalog stored procedures, system extended stored procedures, and dbcc stored procedures.
  - *Tables* – 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 Users Guide* – documents Transact-SQL, the Sybase 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.
- *Unified Agent and Agent Management Console* – describes the Unified Agent, which provides runtime services to manage, monitor and control distributed Sybase resources.
- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Web Services Users 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.
- *FLEXnet Licensing End User Guide* – this Macrovision manual explains FLEXnet Licensing for administrators and end users and describes how to use the tools which are part of the standard FLEXnet Licensing distribution kit from Sybase.
- *SAMreport Users Guide* – this Macrovision manual explains how to use SAMreport, a report generator that helps you monitor the usage of applications that use FLEXnet licensing.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks™ CD, and the Sybase 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 SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Technical Documents Web site, go to Technical Documents at http://www.sybase.com/support/techdocs/.

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.

❖ **Finding the latest information on component certifications**

1. Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.
2. Either select the product family and product under Search by Product; or select the platform and product under Search by Platform.
3. Select Search to display the availability and certification report for the selection.

❖ **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. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the "Technical Support Contact" role to your MySybase profile.

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 san-serif font.
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.

Key	Definition
( )	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.

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Adaptive Server version 15.0 and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

The online help for this product is also provided in HTML, which you can navigate using a screen reader.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

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 installation of Adaptive Server products.

The Cluster Edition

The Cluster Edition allows a single installation of Adaptive Server to operate on multiple nodes as a shared-disk cluster environment with a single-system view. Each server runs on a separate node; referred to as a server instance or instance.The Adaptive Server Enterprise Cluster Edition allows you to “scale” the workload using multiple physical and logical hosts. If instances in the cluster go down, one or more of the still-running instances will take over the workload of the instance that is down.

The cluster manages which instance a client connects to. If one instance is overloaded, the cluster balances the load by shifting clients to other, more available instances.

The primary advantages of the Cluster Edition include:

- Improved availability – the shared-disk nature of the architecture means that applications can continue to run as long as a single cluster member remains viable, even after several others have failed.

- Simple management – does not require data to be repartitioned as cluster membership changes, because data is shared by all instances.

Note The Cluster Edition provides a distributed architecture. Internode communication takes place via a network interconnection, rather than via shared memory. Applications that minimize internode messaging yield the best performance in the Cluster Edition environment.

For more information on Adaptive Server Enterprise Cluster Edition see the *Users Guide to Clusters*.

Single-system presentation

The Cluster Edition supports a single-system presentation. That is, multiple instances that make up the cluster appear to clients as a single system. New client technologies allow clients to connect logically to a cluster while remaining connected physically to individual instances. This logical connection allows Adaptive Server to redirect the client to various instances in the cluster and to dynamically provide the client with high-availability fail over data

Workload management

The Cluster Edition workload manager can customize workload management and failover for each of your business applications so that each performs most efficiently. The logical cluster is the container that allows the workload manager to provide individualized working environments.

Installation steps

Generally, installing the Cluster Edition includes these actions:

1 Pre-installation tasks such as, making sure you have the correct operating system patches and are using the sybase login. See “Pre-installation tasks” on page 23 for more information.

2 Generating and installing the SySAM license. See the SySAM section starting with “Pre-installation tasks for SySAM” on page 15.

3 Verify that the installation is performed to a shared file system location that is visible from all nodes. See “System requirements” on page 3 for details.

4 Verify that the network requirements are in order.

5 Install the software. See Chapter 2, “Installing the Software.”

6 Configure and start the cluster. See Chapter 3, “Creating and Starting the Cluster.”

Adaptive Server Enterprise Cluster Edition options

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 the Cluster Edition

Option	Platform availability	Description
Security and directory services	Linux, Sun	Lightweight directory services and network-based authentication and encryption using SSL and Kerberos.
Partitions	Linux, Sun	Enables semantic partitioning for table row data.
Encrypted Columns	Linux, Sun	Increases both internal and external security parameters and allows for addition of datatypes.

Adaptive Server editions and optional features are unlocked by providing SySAM licenses. For more information, see the *Sybase Software Asset Management Users Guide*.

To determine whether you are running the Cluster Edition enter:

```
select @@version
```

System requirements

The Cluster Edition requires:

- Database devices in the Cluster Edition must support SCSI-3 persistent group reservations (SCSI PGRs). Cluster Edition uses SCSI PGRs to guarantee data consistency during cluster membership changes. Sybase cannot guarantee data consistency on disk subsystems that do not support SCSI PRGs (such a configuration is supported for test and development environments that can tolerate the possibility of data corruption).
- Homogeneous hardware nodes. All nodes must run the same operating system version however, the number of processors and the amount of memory can vary from node to node.
- All database devices, including quorum devices, must be located on raw partitions. You cannot use the Network File System (NFS).
- Raw partitions must be accessible from each node using the same access path. Sybase recommends storage area network (SAN) connected devices.

Note Local user temporary databases do not require shared storage and can use local file systems created as private devices—unlike local system temporary databases, which do require shared storage

- For test environments you can use a single node or machine to run multiple instances of the Cluster Edition in a cluster configuration. However, if you do so, you must use the local file system (not NFS) or SAN Storage for the database devices.
- All hardware nodes must use Network Time Protocol (NTP) or a similar mechanism to ensure their clocks are synchronized.
- All Adaptive Server Enterprise software and configuration files (including the *$SYBASE* directory, the interfaces file) must be installed on a Network File System (NFS) or a clustered file system (CFS or GFS) that is accessible from each node using the same access path. Supported versions of clustered file system are detailed in the next section.
- A high-speed network interconnection (for example, a gigabit ethernet, infiniband) providing a local network connecting all hardware nodes participating in the cluster.
- Sybase recommends that each node in the cluster have three physically separate network interfaces:
  - A public network – for clients to connect.
  - A primary private network – for cluster interconnect traffic.
  - A secondary private network – for cluster interconnect traffic.

The private networks should be physically separated from the public network, and are needed for security, fault-tolerance, and performance reasons. For fault-tolerance, the two private network cards should be on different fabrics so that a cluster will survive network failure.

The private interconnect fabrics should not contain links to any machines not participating in the cluster (that is, all cluster nodes should have their primary interconnect connected to the same switch, and that switch should not be connected to any other switches or routers).

Adaptive Server specifications

Table 1-2: Adaptive Server specifications for Sun Solaris

Hardware and Memory		
Processor	Sparc 64-bit	
Minimum RAM for Adaptive Server	128MB	47,104 2K pages
Default user stack size	84K	Minimum 26K
Minimum RAM per additional user	Approx. 220K	With default stack size, packet size, and user log cache size

Table 1-3: Adaptive Server specifications for Linux

Hardware and Memory	
Processor	AMD Opteron Processor or Intel Xeon with EM64T support
Minimum RAM required for Adaptive Server	128MB
Default user stack size	80KB
Minimum RAM per additional user	Approx. 140KB

Operating system requirements

The Cluster Edition runs only on the Sun Solaris and Linux 64-bit operating systems described in the tables below.

Table 1-4: Operating system requirements for your platform

Platform	Hardware	Operating System	Updates	Preferred RAM
Sun Solaris	Sparc Architecture	Solaris 9	111711-16, 117560-06, 111712-16, 111722- 05	1G
Sun Solaris	Sparc Architecture	Solaris10	120753-05, 120048-03	1G

Platform	Hardware	Operating System	Updates	Preferred RAM
Linux x86-64	x86_64 processor (AMD Opteron or Intel Xeon with EM64T)	RedHat Enterprise Linux 4.0 Kernel: 2.6.9-42.EL Glibc: 2.3.4-2.25 e2fsprogs-1.35-12.4.EL4 (for libuuid.so.1) libaio-0.3.105-2 libaudit-libs-1.0.14-1.EL4 libgcc-3.4.6-3 libstdc++3.4.6-3 (for libstdc++.so.6) libpam.so.0	Update: 4	1G
Linux x86-64	x86_64 processor (Intel Xeon with EM64T)	RedHat Enterprise Linux 5.0 kernel-2.6.18-53.el5 glibc-2.5-18 e2fsprogs-lib-1.39-8.el5 (for libuuid.so.1) libaio-0.3.106-3.2 audit-1.5.5-7.el5 libgcc-4.1.2-14.el5 libstdc++-4.1.2-14.el5 libpam.so.0.81.5 libnsl-2.5	None required	1G
Linux x86-64	x86_64 processor (AMD Opteron or Intel Xeon with EM64T)	SuSE Linux Enterprise Server 9 kernel-smp-2.6.5-7.244 glibc-2.3.3-98.61 e2fsprogs-1.38-4.9 libaio-0.3.102-1.2 libgcc-4.1-4.1.2_20070115-0.2 libstdc++-4.1-4.1.2_20070115-0.2	Update: 3	1G

Platform	Hardware	Operating System	Updates	Preferred RAM
Linux x86-64	x86_64 processor (AMD Opteron or Intel Xeon with EM64T)	SuSE Linux Enterprise Server 10 kernel-smp-2.6.16.46-0.12 glibc-2.4-31.30 e2fsprogs-1.38-25.21 libaio-0.3.104-14.2 libgcc-4.1.2_20070115-0.11 libstdc++-4.1.2_20070115-0.11 compat-libstdc++-5.0.7-22.2	Update: 1	1G
PC-Client	P4 1.0 GHz	WinXP Pro SP2, Win2003 Server SP1, and WinVista Business Edition		512 MB

Note Solaris 10 is shifting to a new method of controlling system resources. shminfo_shmmax is technically obsolete. If you have an entry for it in */etc/system* it will still work. The new way to set this up is to add an entry to the file */etc/project* and reboot Adaptive Server.

Note If you install on RHEL5 or SuSE 10, before you start your server (including backup server), set:

- For SuSE10 – `export LD_POINTER_GUARD=1`
- For RHEL5 – `export LD_POINTER_GUARD=0`

Table 1-5: Minimum disk space requirements for your platform

Platform	Typical install	Default databases	Management features	Total disk space requirements
Sun Solaris	891MB	150MB	25MB	1066MB
Linux x86-64	794MB	150MB	25MB	969MB

Table 1-6: Clustered File System

Operating System	Product
Solaris 9, 10	Veritas Clustered File System 5.0
RHEL 4, 5	Global File System 6.1
SuSE Enterprise 9, 10	OCFS2

We recommend that you patch up your system to the latest available OS patches before installing Adaptive Server components.

Contact your operating system provider for any patches recommended for your installation. Do not use a patch that is earlier than the version suggested for your operating system. Use the patch recommended by the operating system vendor, even if it supersedes the patch listed.

Viewing installed patches

Sybase recommends that you install the latest patches on your system. See "Operating system requirements for your platform" on page 5 for more information.

To list all currently installed patches and display the operating system version level, enter:

- For Sun Solaris – enter:

  ```
  showrev -p
  ```

- For Linux – enter:

  ```
  rpm -q -a
  ```

- For Windows – navigate to Settings | Control Panel | System to view the service pack level under the General tab.

Private interconnect

Note The Cluster Edition only supports the UDP network protocol for private interconnects. You cannot use the TCP network protocol for private interconnects.

A private interconnect is an essential component of a shared disk cluster installation. It is a physical connection that allows internode communication. A private interconnect can be a simple crossover cable with ethernet, or it can be a complex proprietary interconnect with a specialized proprietary communications protocol. When configuring more than two nodes, you typically require a switch. The switch enables high-speed communication between the nodes in the cluster.

The interconnect technology you use to connect the nodes should be scalable to handle the amount of traffic the application creates because of contention. The amount of traffic is directly proportional to the amount of inter-instance updates and inter-instance transfers. Sybase recommends that you implement the highest bandwidth, lowest latency interconnect available.

Sybase recommends that Linux environments have an interconnect bandwidth of 1GB Ethernet.

Adaptive Server CE supports the current standards for interconnects. Sybase recommends that you research the available interconnects to find the one that works best for your site.

Table 1-7 compares technologies available for interconnect with Adaptive Server CE.

Table 1-7: Comparing interconnect technologies

Technology	Bandwidth	Full Duplex Bandwidth	Maximum signal length
1GB and 10GB ethernet	1 and 10GB per second	2GB and 20GB per second	Km
Infiniband	2.5 and 10GB per second	5, 20, and 60GB per second	Km

The Cluster Edition supports Inifiniband:

- IP over IB (internet protocol over Infiniband) mode – the server uses a standard IP interface to communicate with the Infiniband interconnect. This is the simplest mode to configure.

Product descriptions and directory layout

Adaptive Server installs multiple products into the *SYBASE* directory. Table 1-8 provides a brief description of the server products installed, and the top-level directory structure.

Table 1-8: Product descriptions and directory layout

Product	Description
The Server Installation Package	Installed into the *ASE-15_0* directory. • Adaptive Server – the database server. • Backup Server – an Open Server-based application that manages all database backup (dump) and restore (load) operations. • XP Server– an Open Server application that manages and executes extended stored procedures (ESPs) from within Adaptive Server. • Job Scheduler – provides a job scheduler for Adaptive Server. Job Scheduler components are located in their own directory at *ASE-15_0/jobscheduler/*.
Shared Disk Cluster Administration Tools	• Sybase Central™ 4.x is a Java-based framework used by the system management tools. Installed into the *shared* directory • Adaptive Server plug-in. Sybase Central plug-in that provides cluster configuration and complete administration functionality. Installed into *ASEP* directory. • sybcluster – interactive command line interface for configuring and managing the cluster installed in the *$SYBASE/$SYBASE_ASE/install* directory. • Job Scheduler templates and utilities • Unified Agent server – provides a container for mid-tier management application logic accessed by the ASE plug-in and sybcluster. It is installed in the *UAF-2_5* directory. • Agent Management Console – Sybase Central plug-in that manages the Unified Agent. It is installed into the *AMCP* directory. • Interactive SQL– interactive command line interface for configuring and managing the cluster. It is installed into the *$SYBASE/ASE-15_0/bin* directory. • JRE – the Java Runtime Environment (JRE) is a runtime Java virtual machine used to execute Java-based programs such as Sybase Central. Installed into the *shared* directory • The Cluster Edition Administration Utility is installed to *$SYBASE/SDCADMIN-15_0*

Product	Description

Software Developer Kit (SDK)	Installed into the *OCS-15_0* directory.
	• Connectivity:
	  • Open Client (Client-Library, dblib)
	  • Embedded SQL™/COBOL 15.0
	  • XA interface for Adaptive Sever
	Installed into the *DataAccess* directory:
	• ODBC (Windows and Linux only) – a driver used to connect to Adaptive Server from ODBC based applications.
	• OLEDB (Windows only) – a provider used to connect to Adaptive Server from OLE DB-based applications.
	• ADO.NET (Windows only) – a provider used to connect to Adaptive Server from .NET based applications.

Language Modules	Installed into the *$SYBASE/locales* and *$SYBASE_ASE/locales* directories. 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.
Sybase Software Asset Management (SySAM)	Installed into the *SYSAM-2_0* directory. Provides asset management of Adaptive Servers and optional features.
jutils-2_0	A collection of Adaptive Server utility programs including: • jisql – a JDBC-based GUI isql tool. • ribo – a tool to trace TDS traffic between client program and Adaptive Server.
jConnect	Installed into the *jConnect-6_0* directory. Provides a Java Database Connectivity (JDBC) driver for Adaptive Server.

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 1-9: Products descriptions

Product	Description

Software Developer Kit (SDK)	Installed into the *%SYBASE_OCS%* directory.
	• Connectivity:
	  • Open Client (Client-Library, dblib)
	  • Embedded SQL™/COBOL 15.0
	  • XA interface for Adaptive Sever
	Installed into the *DataAccess* directory:
	• ODBC (Windows and Linux only) – a driver used to connect to Adaptive Server from ODBC based applications.
	• OLEDB (Windows only) – a provider used to connect to Adaptive Server from OLE DB-based applications.
	• ADO.NET (Windows only) – a provider used to connect to Adaptive Server from .NET based applications.

PC-Client Administration Tools	• Sybase Central™ 4.x is a Java-based framework used by the system management tools. Installed into the *Shared* directory
	• Adaptive Server plug-in. Sybase Central plug-in that provides cluster configuration and complete administration functionality. Installed into *ASEP* directory.
	• sybcluster – interactive command line interface for configuring and managing the cluster installed in the *%SYBASE%\%SYBASE_ASE%\/install* directory.
	• Job Scheduler templates and utilities
	• Unified Agent server – provides a container for mid-tier management application logic accessed by the ASE plug-in and sybcluster. It is installed in the *UAF-2_5* directory.
	• Agent Management Console – Sybase Central plug-in that manages the Unified Agent. It is installed into the *AMCP* directory.
	• Interactive SQL– interactive command line interface for configuring and managing the cluster. It is installed into the *%SYBASE%\%SYBASE_ASE%\bin* directory.
	• JRE – the Java Runtime Environment (JRE) is a runtime Java virtual machine used to execute Java-based programs such as Sybase Central. Installed into the *%SYBASE%\Shared\Sybase Central 4.3\win32* directory
	• The Cluster Edition Administration Utility is installed to *%SYBASE%\SDCADMIN-15_0*
	• jutils-2_0 - A collection of Adaptive Server utility programs including:
	  • jisql – a JDBC-based GUI isql tool.
	  • ribo – a tool to trace TDS traffic between client program and Adaptive Server.

Language Modules	Installed into the *%SYBASE%\locales* directory. Provides system messages and date/time formats.

Product	Description
Character Sets	Installed into the *charsets* directory. Provides character sets available for use with Adaptive Server.
jConnect	Installed into the *jConnect-6_0* 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.
- SAMreport – an asset management reporting tool that helps you monitor the usage of SySAM enabled Sybase products.

Product licensing with SySAM 2.0

Starting with Adaptive Server Enterprise 15.0, the Sybase Software Asset Management System (SySAM) implementation has changed. With these changes, SySAM configuration is no longer optional. Review the updated SySAM implementation and plan your SySAM deployment before you install Adaptive Server Enterprise.

- For more complex installations, and for general information about SySAM 2.0, see the *Sybase Software Asset Management Users Guide*.

Warning! SySAM-2.0-enabled products work for an initial period without a license. If you do not install a valid license during this grace period, the product ceases to function.

CHAPTER 2

Installing the Software

This chapter describes the component contents, system and platform requirements for Cluster Edition. In addition it will guide you through the set up and the installation of the software.

Contents

The Cluster Edition includes the following:

- The Cluster Edition of Adaptive Server Enterprise
- PC-Client

Pre-installation tasks for SySAM

For simple installations follow the instructions in this book. No additional information should be necessary.

The Adaptive Server installation guide (this document) has the following information about SySAM-related tasks:

- Installation instructions are described in “Installing components with the Installer in GUI mode” on page 26.
- If you have problems, see Chapter 10, “Troubleshooting SySAM.”
- For information about using SySAM for asset management, see *Sybase Software Asset Management Users Guide*.

Before you install Adaptive Server, you should:

1 Decide the SySAM license model you will use. The choices are:

- The unserved license model – gets licenses directly from the license file. If you are using an unserved license, you should simply save the license to the machine on which you have installed Adaptive Server.
- The served license model – uses a license server to manage the allocation of your licenses to multiple machines. See the following for more information.

For information about served and unserved licenses, and which model is best suited for your environment, see Chapter 2, "Choosing a License Model" in the *Sybase Software Asset Management Users Guide*.

2 If you plan to use a served license, you need a license server. Decide whether it will be an existing license server or a new license server. The license server need not be on the same machine or running on the same operating system and architecture.

- If you are going to use an existing license server, you must know the server host name and port number.
- For information about creating a new license server before installing Adaptive Server, see "Installing a new license server" on page 21.

Note There can be only one instance of a SySAM license server running on a given machine. If you want to setup a SySAM 2.0 license server on a machine that is already running a SySAM 1.0 license server, the old license server must be migrated to 2.0. A migrated license server can serve licenses for both SySAM 1.0 and SySAM-2.0-enabled products.

Please see Appendix F of the *Sybase Software Asset Management Users Guide* for instructions on migrating a license server.

For information about choosing a license server host, see the "Choosing a license server host" section in Chapter 2, "Choosing a License Model" of the *Sybase Software Asset Management Users Guide*.

For information about license server administration, see Chapter 4, "License Server Administration" in the *Sybase Software Asset Management Users Guide*.

3 Sybase recommends that you get your license files from the Sybase Product Download Center (SPDC) at https://sybase.subscribenet.com before you begin installing Adaptive Server. For information about accessing SPDC, see "Accessing SPDC" on page 17, and for information about generating your licenses at SPDC, see "Generating licenses at SPDC" on page 18.

Accessing SPDC

Once you have purchased a Sybase product, typically you are expected to download the product and generate its licenses from the online Web portal, the Sybase Product Download Center (SPDC.)

To access SPDC when you have purchased Sybase products from Sybase:

1 Once you have ordered a Sybase product you receive a welcome e-mail that contains a URL to the SPDC site, as well as a user name and password.

Note If you have purchased your Sybase software from a Sybase reseller, you will receive a web key rather than an e-mail.

2 Click on the URL, and when prompted, enter the designated user name and password. Your login is typically your e-mail address. If you have forgotten your password, use the password finder on the SPDC login page.

Getting your host ID

To get your host ID:

1 Change to *$SYBASE/SYSAM-2_0/bin*. Where *$SYBASE* is the Sybase installation directory.

2 To get the host ID enter:

```
./lmutil lmhostid
```

Note On some platforms, the host ID is derived from the network adapter address. If your machine has multiple network adapters, `lmutil lmhostid` returns one host ID for each network adapter. The output may look similar to:

```
The FLEXlm host ID of this machine
```

```
is ""0013023c8251 0015c507ea90""
Only use ONE from the list of hostids.
```

Choose **one** of these host IDs. Sybase recommends using the value associated with the primary wired Ethernet adapter. Do not use values associated with internal loopback adapters. If you cannot determine which host ID to use from the `lmutil lmhostid` output, use the native operating system command to get additional details to help make the determination. See to the SPDC FAQ titled,"What's my Host ID," or "Appendix A" of the *FLEXnet Licensing End User Guide* for the exact commands for your platform.

3 Save the host ID so you can use it later at SPDC. Your host ID is platform-specific, but may be similar to:

- On Linux – `00400516E525`
- On Solaris – `170a3472`

Generating licenses at SPDC

This section provides information about how to generate a license for a served and an unserved license model. To generate your license, regardless of license model:

1 At the Welcome screen at SPDC, you can view all of the product families to which you have access. Select the Adaptive Server product family to generate a license.

2 The Cluster Edition is listed under available products. Select the Cluster edition for the operating system you are using.

3 The license agreement displays. You must agree to the terms and conditions in the license in order to generate a license. To do so, click I Agree.

Note The license agreement only displays the first time you attempt to download a product. Once you have agreed to the license, you will not encounter the license agreement for all subsequent downloads of that product.

4 At the product download page, click the License Keys link.

5 The License Information page displays. Select the license you wish to generate. When you are selecting the license you wish to generate, consider:

- The license *type* for the license you wish to generate. For more information about license types, see the "License types" in Chapter 3 "Getting and Using your License" of the *Sybase Software Asset Management Users Guide*.
- If you are unsure what you have licensed, your company's purchase order should have the product name, edition, and license type.

6 Once you've determined the license you wish to generate, select it via the Generate radio button on the left side of the screen. Scroll to the bottom of the page, and click Select to Generate.

7 Step one of the SySAM license generation wizard asks you to choose between a served license and an unserved license. If you choose a served license, continue with the steps in "Generating a served license" on page 19, and if you choose an unserved license, continue with the steps in "Generating an unserved license" on page 20.

Note Some Sybase Products or specific license types do not give you a choice of license model to use, and therefore do not display this page. If this is the case, proceed with the SySAM license generation wizard to generate your license.

❖ **Generating a served license**

1 Select Served License and click Next.

2 Indicate the quantity of licenses you wish to generate and click Next.

3 Enter the License Server Host ID and optionally the host name and port number:

a Sybase recommends that you provide the host name here to ease future license administration.

b A port number is not required unless you are using a 3-node redundant cluster. A valid number is any unused port number between 0 and 64000. On UNIX, choose a port greater than 1024, since those less than 1024 are privileged port numbers. If no TCP/IP port number is specified, one of the default ports in the range of 27000 and 27009 is used.

c If you want to generate a license for a 3-node redundant cluster, then you must specify the host ID, host name, and port number for the three nodes of the cluster. The port number is not optional for this configuration and should be outside of the 27000 to 27009 range.

d Be sure to get your server host ID before you began the process of generating your licenses. For information about how to get the host ID, see "Getting your host ID" on page 17. You have the option to specify redundant server information if you choose to configure your servers for three-server redundancy.

4 Click Generate.

5 Once you have generated the license, you can:

a Download the license file by clicking the Download License File button.

b Print a paper copy of the license by clicking the Print Friendly button.

c Return to the license information page to generate additional licenses.

6 Once you have generated all the licenses you need, save them to the *licenses* directory in your license server installation.

Note You must save your license files with a *.lic* extension, or SySAM 2.0 will not recognize them.

Sybase recommends that you set up the license server and required licenses before you install your SySAM-2.0-enabled Sybase product.

❖ **Generating an unserved license**

1 Select Un-served License and click Next.

2 Select the number of machines you want to license and click Next. You can generate licenses for a maximum of ten machines at a time.

3 Enter the host ID, and optionally the host name for each machine for which you want to generate a license. You should have determined the host ID before you began to generate your licenses. For information on getting your host ID and host name, see "Getting your host ID" on page 17.

For some license types, you are asked to enter the number of CPUs or licenses for the machine where this license will be used.

4 Click Generate.

5 Once you have generated the license, you can:

- Download the license file by clicking the Download License File button.
- Print a paper copy of the license by clicking the Print Friendly button.
- Return to the license information page to generate additional licenses.

6 Once you have generated all the licenses you need, save them to the specific location required by your product.

Note You must save your license files with a *.lic* extension, or SySAM 2.0 does not recognize them.

Installing a new license server

Note If you have selected a served license model, you must install a license server.

For information about operating system requirements, see "System requirements" on page 3.

Perform the following steps to install a new license server.

1 Install the license server.

a Launch the Sybase installer program by entering:

```
./setup
```

b The Welcome Window displays. Click Next.

c Accept the license agreement.

d Enter or select the destination directory.

e Select the Custom installation.

f Select only "SySAM License Server" for installation.

g Deselect other components, to install the license server only.

h Click OK for the installer to install the license server.

Note The license server cannot be started until there is at least one valid served license installed in the *licenses* directory.

2 Get the host ID for the machine on which you will be running the license server. For information on how to generate the host ID, see "Getting your host ID" on page 17.

3 Go to the Sybase Product Download Center (SPDC) at https://sybase.subscribenet.com.

4 Generate the licenses for the products you want to install. For instructions on generating your licenses, see "Generating licenses at SPDC" on page 18.

5 From command prompt, navigate to installed directory to source the environment variables.

- SYBASE.csh

6 Copy the license file in the *$SYBASE/SYSAM-2_0/licenses* directory on the network license server machine.

7 Refresh or restart the license server:

a Change to:

$SYBASE/$SYBASE_SYSAM/bin

b If the license server is not already started, start it, by entering:

```
sysam start
```

If the license server is already started, use the reread command to make the license server read the new license files:

```
sysam reread
```

8 Validate that the license daemon is running by entering:

```
sysam status
```

You can expect to see output for a running server that is similar to:

```
lmutil - Copyright (c) 1989-2006 Macrovision Europe
Ltd. and/or Macrovision Corporation. All Rights
Reserved. Flexible License Manager status on Wed
5/24/2006 15:04

License server status: 27000@mysysamserver
```

```
License file(s) on keyserver:
/opt/sybase/SYSAM_0/licenses/mysysamserver_60302031
5.lic
```

For more information, see Chapter 10, “Troubleshooting SySAM.”

Pre-installation tasks

Before installing the Cluster Edition:

1 Install operating system patches, if required.

2 Create a “sybase” account on your system to perform all installation tasks. This account should own all devices and files, and must have permission to read and write all devices the cluster uses.

   Create all devices with this “sybase” account.

3 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.

4 If you are using a local account local to just one physical box then create the sybase user on each machine and make sure it has the same user ID.

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

   The *$SYBASE* location must be on a shared file system, accessible from all cluster nodes using the same path.

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

7 Verify that your network software is configured.

   The Cluster Edition implicitly requires that your network be configured for the nodes you plan to include in the cluster.

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

8 Adjust the operating system shared memory parameter for your platform

Sun Solaris

To adjust shared memory segments of the operating system, add the following line to the configuration file */etc/system*, where *x* is the number of shared memory segments:

```
set shmsys:shminfo_shmseg=x
```

- The method of controlling system resources is to add an entry to the file */etc/project* such as in the following example, where project.max-shm-memory=(privileged,17179869184,deny) is the name for the project parameter:

```
project-sybase:200:For use by Sybase:sybase:sybase:
project.max-shm-memory=\(privileged,17179869184,deny)
```

  - privileged – is a threshold value on a resource control that constitutes a point at which local actions can be triggered or global actions such as logging in to a machine can occur. Three privilege levels exist:
    - basic – can be modified by owner of the calling process
    - privilege – can be modified by privileged callers. In specifying the threshold value of "privileged," you can use the abbreviation "priv," such as in the following, where "17179869184" is the threshold value (16GB) on the resource control:

      ```
      project.max-shm-memory=(priv, 17179869184, deny)
      ```

    - system – is fixed for the duration of the operating system instance

    You can set setting project.max-shm-memory while the system is running by using the prctl command. The rctladm command can be used to set things permanently.

    The default value for project.max-shm-memory is 25% of the physical memory on the system. The maximum value is UINT64_MAX, which works out to 18446744073709551615 bytes, so essentially this is limited only by the size of physical memory.

- deny – means attempts to use more than 16GB are denied.
- Adjust shared memory segments. Do this if Adaptive Server can not get large memory as a single segment or if backup server stripes fail due to insufficient segments.

  Depending on the number and types of devices you use for backup (dump) and recovery (load), you may need to adjust the shared memory segment parameter in the operating system configuration file to accommodate concurrent Backup Server processes. The default number of shared memory segments available for process attachments is 6.

Adaptive Server may allocate shared memory segments after start-up if any reconfiguration through sp_configure requires additional memory. You may need to account for these additional segments. Allocate the maximum memory you will make available to Adaptive Server, by using the allocate max shared memory configuration parameter. See the *System Administration Guide* for more information.

Linux

- The operating system shared memory default for most Linux releases is 32MB. The minimum required by Adaptive Server is 64MB for a default server with 2K pages. A higher value is required if you plan to increase the Adaptive Server 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 at runtime, where *nnn* is 1073741824 for 1GB of shared memory:

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

  To guarantee that this value is applied every time the system is started, edit the */etc/sysctl.conf* file.

- On Linux RHEL Update4.0 and greater, Adaptive Server running on multiple engines requires the security feature Exec-Shield to be disabled.

  To disable Exec-Shield:

  a Add the following lines in */etc/sysctl.conf:*

  ```
  kernel.exec-shield=0
  kernel.exec-shield-randomize=0
  ```

  b Enter the following for the action to take effect:

  ```
  /sbin/sysctl -P
  ```

  For additional information, see the Red Hat Web site at http://www.redhat.com/f/pdf/rhel/WHP0006US_Execshield.pdf.

  If the server fails to restart after you have adjusted the SHMMAX parameter, you may also need to increase the value of another Kernel parameter, SHMALL, which is the maximum amount of shared memory that can be allocated. You can modify its value in the */etc/sysctl.conf* file.

  The syntax for increasing SHMALL is:

  # /sbin/sysctl -w kernel.shmall=nnn

Installing the Cluster Edition

Warning! If you already have a working Adaptive Server on your system, **do not** install the Cluster Edition in the same *$SYBASE* directory.

The Cluster Edition includes new major versions of Adaptive Server and many of the supporting components. If you install the Cluster Edition into the same directory as earlier versions of the same components, the Cluster Edition overwrites the older versions.

Sybase recommends that you install the Cluster Edition in its own directory. This directory must not contain Adaptive Server version 15.0.x. If you must install other products into the same directory, install the Cluster Edition last.

Installing other products on top of the Cluster Edition will likely result in one or more products not working correctly.

The instructions in this section are for using the installer in GUI mode. For information on alternate methods of installation see the Appendix, "Alternative Installation Methods."

Installing components with the Installer in GUI mode

1 Download and extract the Adaptive Server install image from the secure download site.

2 Navigate to the directory where you have extracted the software and start the installer by entering:

```
./setup
```

3 The Welcome screen displays. Click Next.

4 When the license selection window displays, select the most appropriate location from the list, read the license terms, and then click "I agree" to proceed.

If the country you are located in is not listed, select the most appropriate area.

5 Click Next.

6 The Install directory window appears. Click Next to accept the default of:

```
/opt/sybase
```

You can also enter a directory for installation. If you specify a directory that does not already exist, the installer creates it for you. Sybase recommends that you install the Cluster Edition in its own directory.

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

- Typical
- Full
- Custom

8 If you select Custom, the Products and Features Selection window displays.

- Select the products you want to install by checking the box next to each product name.
- After you have made your selection, click Next.

Note Some features are dependent on other features; therefore, you cannot unselect some items without first unselecting others.

9 Before proceeding to the next window, the installer verifies the selections and checks for dependencies and available disk space. If there is not enough disk space an error message appears.

10 If you are installing this Adaptive Server into a directory with an existing Adaptive Server installation, the installer checks to see if the existing license authorizes use of the Adaptive Server version being currently installed. If the license does not authorize the use of this version, then a warning is displayed asking if you want to proceed with the installation.

- You should proceed only if you have an updated license that authorizes use of this version.

Note Note: Failure to apply an updated license will cause Adaptive Server to stop working after this updated version is installed.

11 The Sybase Software Asset Management License Server screen allows you to designate where Adaptive Server will find its licenses. Where you choose to store your license depends upon whether you have selected a served or unserved license model.

Note For information about served and unserved licenses, and about choosing the best license server for your environment, see Chapter 2, "Choosing a License Model" of the *Sybase Software Asset Management Users Guide*.

- You must specify whether the licenses will be obtained from a license server. If you answer:
  - Yes – you must know the host name of the machine where the license manager is running, and the port number if you have used a non-default port number.
  - No – you are using unserved licenses. You will be reminded to download and install the license file after you have finished the installation.

Note When you download the license file, copy it to *$SYBASE/$SYBASE_SYSAM/licenses.*

12 The Product Edition and License type panel is displayed next. Select the Product Edition you are installing and the License type under which you have licensed Adaptive Server.

Note For this version, there is only one product edition choice, "Cluster Edition."

13 The Sybase Software Asset Management Notification screen asks you to configure your server for e-mail notification. When configuration is enabled, designated users receive information about license management events requiring attention.

Provide the following information:

- SMTP server host name
- SMTP server port number
- E-mail Return Address
- Recipient e-mail addresses
- Message severity that triggers e-mail messages

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

Verify that you have selected the correct type of installation. Click Next to proceed.

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

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

Click Finish to exit the installer and go on to the next step.

17 Adaptive Server Enterprise Cluster Edition files are on your computer. Next, follow the steps in Chapter 3, “Creating and Starting the Cluster.”

CHAPTER 3

Creating and Starting the Cluster

This chapter describes how to configure and start a Sybase shared-disk cluster. Follow the instructions in this chapter after following instructions in Chapter 2, “Installing the Software.”

Sybase recommends that you use the Adaptive Server plug-in or the sybcluster utility to create and manage your shared-disk cluster.

Note You can also configure and manage the cluster manually. See Appendix B, “Manually Setting Up and Managing the Cluster.”

Overview

Complete these steps in this order to configure and start the Cluster:

1 Read “Before you begin” on page 32.

2 Set the *$SYBASE* environment and start the Unified Agent on each node of the cluster. See “Setting the environment and starting the Unified Agents” on page 34.

3 Configure the cluster using one of the following:

- The Adaptive Server plug-in. See “Creating the cluster server using the Adaptive Server plug-in” on page 37.

    - The sybcluster utility. See “Creating the cluster using sybcluster” on page 44.

4 (Optional) Configure the auxiliary servers: Monitor Server, XP Server, Backup Server, and Job Scheduler. See “Configuring auxiliary servers” on page 51.

5 If the installation and startup fails you must follow the steps at the end of this chapter before re-installing.

Once configured, each instance of Adaptive Server in the cluster shares:

- All databases and database devices (for example, all instances share the same master database)
- A common *$SYBASE* directory
- All server binaries (for example *dataserver*) and all scripts (for example *installmaster*)
- All configuration files, including the cluster configuration file
- The *interfaces* file
- A quorum device used to coordinate the instances with the cluster

Before you begin

Sybase recommends that you plan your Adaptive Server shared-disk cluster before you begin the configuration process.

Before you begin, determine:

- The location of the *$SYBASE* installation directory.
- The cluster name.
- The number of instances and the instance names.
- The Domain Name Service for the network on which the cluster will run. The Unified Agent and the sybcluster utility will not function properly if the DNS entries are not entered correctly for each node.
- The number of nodes in the cluster and the node names. Sybase recommends that you use one node for each of the instances in the cluster with two or more engines per instance.

- The raw devices to be used by each database device. For the Cluster Edition, you must create all devices, database devices, and quorum devices as raw devices on shared disks.

  Note Local user temporary databases do not require shared storage and can use local file systems created as private devices.

- The range of port numbers used by each instance to exchange messages with other instances via the private interconnection. Choose unique port numbers that are not in use by other applications.

  Note The Adaptive Server plug-in and sybcluster supply default values if your system does not have an existing standard.

- The query or listening port number for each instance. Choose unique port numbers that are not in use by other applications.
- The IP addresses of the private primary and secondary network cards on each node. Only UDP network protocol is supported at this time.

Making disk devices accessible on Linux

You must configure all disk devices used by the Cluster Edition so that they are accessible from all the nodes in the cluster.

The account used to start the cluster requires permission to read and write to all of the disk devices.

The cluster should have write permission on some of the "*/dev/sg**" files that correspond to configured database devices, as the driver expects write permission for SCSI-3 PGR commands used in I/O fencing, and SCSI generic drivers require write permission.

❖ **Making disk devices accessible**

1 Use the UNIX ls -l list command to verify paths and file permissions. You can use the UNIX dd utility to verify that the devices can be read and written to by the Sybase account.

2 Use the chmod change mode or chown change owner commands to correct the write access to "*/dev/sg**" files.

The write access permissions of these files may change to root only after restarting the machine.

Creating a simulated cluster on a single node

Sybase recommends that you configure each instance on a different node ie. one node for each instance. However, for a testing environment, you can create a simulated cluster on a single node with all instances running on that node. For best performance make sure the number of engines in all instances running on a single node does not exceed the number of CPUs on that node.

Additionally, Sybase strongly recommends tuning the runnable process search count parameter to *'3'* in such an environment. This can be set using the sp_configure stored procedure.

Setting the environment and starting the Unified Agents

The Unified Agent enables the distributed management of the cluster. Set the *$SYBASE* environment variables and start a Unified Agent on each node that will host instances in the cluster.

If another application is already using port 9999 see step 3.

❖ **Starting the Unified Agents**

From the *$SYBASE* directory of each instance:

1 Source the *SYBASE.sh* file if you are using the Bourne shell or C shell.

- C shell:

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

- Bourne shell:

```
. ./SYBASE.sh
```

2 Start the Unified Agent:

```
$SYBASE_UA/bin/uafstartup.sh &
```

Note The Unified Agent creates *$SYBASE_UA/nodes/<node_name>/log/agent.log*, and sends agent output to that file.

3 To optionally specify the UAF port number, enter the following, where *port_number* is any available network port:

```
$SYBASE_UA/bin/uafstartup.sh -port port_number
```

You can change the port number at the command line with the command:

```
uafstartup.sh -port port_number
```

UAF automatically updates its configuration to start on the new port

Creating and starting the cluster

Create the cluster in one of these ways:

- Adaptive Server plug-in to Sybase Central – see "Starting and stopping Sybase Central" on page 37.
- sybcluster utility – see "Creating the cluster using sybcluster" on page 44.
- Manually, using the dataserver binary – see Appendix B, "Manually Setting Up and Managing the Cluster."

Note The Adaptive Server plug-in and sybcluster do not create a *run_server* file when they create a cluster. After you have created a cluster using the Adaptive Server plug-in or sybcluster, you must start this cluster and each instance using the Adaptive Server plug-in or the sybcluster utility. You cannot start this cluster from the command line using *run_server* files.

Worksheet for creating a cluster

Use Table 3-1 to help you plan your cluster. It lists all the information you need to create a cluster using the Adaptive Server plug-in or sybcluster. Default values are in square brackets. For an example of a completed worksheet, see "A sybcluster Sample Session" on page 173.

Table 3-1: Configuration values for creating a cluster

Parameter	Default value	Value
Cluster name		
Number of instances		
Maximum number of instances	4	
Master Device		
Full path and name for the master device		

Parameter	Default value	Value
Size of master device [80MB]	60MB	
Size of master database [60MB]	30MB	
Instance page size in kilobytes	2	
Sybase System Procedure Device		
Full path and name for sybsystemprocs device		
Size of sybsystemprocs device (MB)	150	
Size of sybsystemprocs database	132	
System Database Device		
Full path and name of system database device		
Size of system database device (MB)	6MB	
Size of system database (MB)	6MB	

Does this cluster have a private network	Y	
Does this cluster have a secondary private network	Y	
Quorum Device		
Full path and name for the quorum device		
Trace flags		
Verify file locations		
The location of the home directory	*$SYBASE*	
Full path to environment variable script	*$SYBASE/SYBASE.sh*	
Path to *$SYBASE_ASE*		
Path to interfaces file directory	*$SYBASE*	
Path to the dataserver configuration file	*$SYBASE/cluster_name.cfg*	
Instance Information		
Node name		
Instance name		
Query port number for instance1		
Primary protocol address for instance1		
Secondary protocol address for instance1		
Local System Temporary Database		
Local system temporary database device name (Enter the name of the Adaptive Server database device for the local system temporary database.)		
Path to local system temporary database device		
Local system temporary database device size (MB)		
Local system temporary database name		
Local system temporary database size (MB)		

Parameter	Default value	Value
Node name		
Instance name		
Query port number for instance2		
Primary protocol address for instance2		
Secondary protocol address for instance2		
Local System Temporary Database		
Local system temporary database device name (Enter the name of the Adaptive Server database device for the local system temporary database.)		
Path to local system temporary database device		
Local system temporary database device size (MB)		
Local system temporary database name		
Local system temporary database size (MB)		

Node name		
Instance name		
Query port number for instance 3		
Primary protocol address for instance3		
Secondary protocol address for instance3		

Creating the cluster server using the Adaptive Server plug-in

The Adaptive Server plug-in helps you perform complex administration tasks without the need to remember the syntax of Transact-SQL commands or system stored procedures.

Starting and stopping Sybase Central

To start Sybase Central:

- On UNIX – run *$SYBASE/shared/sybcentral43/scjview.sh.*
- On Windows – select the shortcut from the Start menu.

To stop Sybase Central, select File | Exit.

Registering the Adaptive Server plug-in

The Adaptive Server plug-in is registered in Sybase Central as part of the server installation. However, if the Adaptive Server plug-in is not correctly registered, you can manually register the Adaptive Server plug-in:

- On UNIX – run *$SYBASE/ASEP/bin/registerASEP*.
- On Windows – run *%SYBASE%\ASEP\bin\registerASEP.bat.*

To register the Adaptive Server plug-in manually:

1 From Tools | plug-ins, select Register. A registration wizard appears.

2 Select "Register a plug-in by specifying a plug-in registration file."

3 Click Browse.

4 Navigate to *$SYBASE/ASEP/bin* (*%SYBASE%\ASEP\bin* on Windows) and select *ASEPlugin.jpr*. Follow the instructions in the wizard to register the Adaptive Server plug-in.

Creating a cluster with the Adaptive Server plug-in

The Adaptive Server plug-in includes a wizard that steps you through the process of creating, starting and stopping a cluster. For complete instructions for using the Adaptive Server plug-in to configure and manage the cluster see "Administering Clusters with Adaptive Server plug-in" in the *Users Guide to Clusters*. Adaptive Server plug-in does not provide upgrade support, however, sybcluster does. If you are upgrading to the Cluster Edition see the Upgrade Chapter.

To start the Create Cluster wizard:

1 Select the Adaptive Server Enterprise icon in the left pane.

2 Select the Utilities tab in the right pane.

3 Right Click on Create Cluster to start the Create Cluster wizard.

Or you can select Tools | Adaptive Server Enterprise | Create Cluster to start the Create Cluster wizard.

Follow the instructions in the Create Cluster wizard. Complete these steps using the values you indicated on Table 3-1 on page 35.

1 Enter the name of the cluster.

2 Select the platform from the drop-down list.

3 Select the maximum number of instances you plan to use for this cluster.

4 Click Next.

5 Discovery settings – you can change the settings the Adaptive Server plug-in uses to find cluster nodes to use in the cluster. For more information see "Administering Clusters" in the *Users Guide to Clusters*.

6 Select the hosts that participate in the cluster.

Enter the unified agent user name and password.

Select Next.

7 Quorum device path – enter the full path to the quorum device. Click Next.

8 Enter:

- Sybase home – the Adaptive Server release directory (also the value of *$SYBASE*).
- ASE home – the full path to the ASE-15_0 directory (also the value of *$SYBASE/$SYBASE_ASE*).
- Interfaces directory – path to the directory containing the interfaces file.
- Cluster configuration file – full path to the cluster configuration file.
- Sybase environment script – full path to the script that you use to set the environment variables.
- Start parameters – list any startup parameters for the cluster.
- Trace flags – list any trace flags you want used at startup.

Click Next.

9 Enter the master device information:

- Master device path – full path to the master device.

Note By default, the Adaptive Server plug-in uses the *$SYBASE/data/cluster_name* directory for the master device; however, the plug-in does not create this directory. If this directory does not already exist, you must either create it now before you can continue, or change the value to an existing path.

- Device size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).
- Database size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).

- Page size – page size of the instance; 2-, 4-, 8-, or 16K (the default is 2K)

Click Next.

10 Enter the system procedures (sybsystemprocs) information:

- System procedures device – enter the full path to the sybsystemprocs device.
- Device size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).
- Database size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).

Click Next.

11 Enter the system database (sybsystemdb) information:

- System database device – full path to the sybsystemdb device.
- Device size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).
- Database size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).

Click Next.

12 Verify the following or change to appropriate values:

- Network Protocol – select the UDP network protocol.
- Starting Port – select a starting port number for the instances. Make certain this port number is not already in use.
- Select Private Network or Public Network.
- Check the box if you have a secondary network available.

13 Click Validate Ports to make sure the ports are not currently in use.

Click Next.

14 The Adaptive Server plug-in specifies the server instances currently configured. Click Add to add instances to the list.

15 Enter the following for *each* instance you are adding to the cluster:

- Instance name – the name of the instance.
- Node – select the node.

- Query port – select the port number for the query port entry in the interfaces file.

  Select Validate Port to make sure this port is available.

- Primary – enabled if you select a private network, host name or protocol address for the primary private connect.
- Secondary – enabled if you selected a secondary address , host name or protocol address for the secondary private connect.
- Log file – full path to the errorlog for this instance.
- Start parameters – list any startup parameters for the instance.
- Traceflags – list any trace flags you want used at startup.

Select OK to accept the cluster instance properties, then select Next.

16 Specify the local system temporary databases and devices. The plug-in lists the default values for the local temporary databases. Accept the defaults or select each table cell to change its value.

To add more devices, click Create Device. Enter:

- Device name – name of the device.
- Device path – full path to the database device.
- Device size – size in megabytes (MB), gigabytes (GB), or terabytes (TB).

  Click OK.

Click Next.

17 To automatically start the cluster after the wizard completes, check the appropriate box when the cluster is created. If this option is not selected the cluster will be left in a shutdown state after the Create Cluster wizard completes.

Click Next to create this cluster.

Adaptive Server plug-in displays another window displaying the server messages as it creates the cluster.

Adding logical clusters

The logical cluster is an abstract representation of one or more instances in a physical shared-disk cluster that is used to manage workload, failover and client application access to the cluster. Each logical cluster has a set of instances it runs on and can have a set of instances to which it fails over. Routing rules direct incoming connections to specific logical clusters based on an application, user login, or server alias supplied by the client. Other rules can restrict the logical cluster to bound connections or allow any authenticated connection to access it.

This section describes how to create a logical cluster using Adaptive Server plug-in. For a more detailed description of the logical cluster, see the "Managing the Workload" chapter of the *Users Guide to Clusters*".

See the *Users Guide to Clusters* for information about working with logical clusters after they are created or defined.

To add logical clusters:

1 Select Logical Clusters from the Workload Management folder.

2 Select Add Logical Cluster. The Adaptive Server plug-in starts the Logical Cluster wizard.

3 Name the logical cluster. Select a name that is representative of the job this logical cluster will perform. For example, SalesLC.

   Click Next.

4 Select the instances that make up the logical cluster.

   a Select Add to see a list of available instances. The Add Instance to Logical Cluster window lists their instance name, ID, and state (whether it is online or offline). These instances are also known as the "base" instances.

   b Select the instances you want to add. Hold the Control key down to select multiple instances. Click OK.

   c Click Next.

   To remove an instance from this list, highlight its name and select Remove.

5 Add the failover server instances. These are instances on which the logical cluster will run if one or more of the base instances fail. Any instance in the physical cluster can be a failover resource. Workload management capabilities let you group and configure resources to specify failover order and precedence.

a Select Add for a list of available failover instances.

b The Adaptive Server plug-in displays the Add Failover to Logical Cluster window, which allows you to select:

- The Failover Group – failover groups let you specify preference and order as to which failover instances are to be used. Lower-numbered groups are used first.

  Select the number for the failover group to which you want to associate these failover instances. Click OK.

- The instance – the Add Failover Instance to Logical Cluster window lists the instances available to be failover instances.

  Select the instances. Hold the Control key down to select multiple instances. Click OK.

c Click Next.

6 Add routed applications, logins, and aliases. Routing rules allow you to specify the logical cluster you want specific applications, logins, and aliases to connect to. See the *Users Guide to Clusters* for more information about routing.

The "Routed applications, logins, and aliases" window lists the Name and Type of currently defined routes. To add additional routes for:

- Applications – select Add Application Route. Specify the name of the application in the Application Route window. Click OK.
- Logins – select Add Login Route. Select the login name from the list in the New Login Binding window (hold the Ctrl key to select multiple names). Click OK.
- Aliases – Select Add Alias Route. Specify the alias name in the Alias Route window. Click OK.

To drop a listed route, select the route name and Click Drop Route.

Click Next.

7 Define the options for the logical cluster (for more information about these options, see the "Managing the Workload" chapter of the *Users Guide to Clusters*.

- System view – an instance view means that monitoring and informational tools such as sp_who, sp_lock and monitoring tables display information only for the instance on which they are running. A cluster view means that they display information for all instances in the cluster.
- Automatically start logical cluster – select this option to determine whether you want the logical cluster started when the cluster starts.
- Failover mode – Select either “instance” or “group” from the drop-down list to specify whether you want the instances brought online as a group or individually.
- Fail to any – specifies whether any instance can be a failover resource or only a specific instance can be a failover resource.
- Down routing mode – specifies how client connections are routed if the logical cluster designated by the routing rule is not available. The options are:
  - system – sends un-routable connections to the system logical cluster.
  - open – sends unroutable connections to the open logical cluster.
  - disconnect – disconnects unroutable connections.

    See the *Users Guide to Clusters* for more information.
- Logical cluster roles – select this option to indicate whether this logical cluster assumes an open role, meaning that all connections not routed to a logical cluster via an explicit routing rule are routed to the current open logical cluster. When you create a new cluster, the system logical cluster is automatically designated the open logical cluster. You can reassign the open role to another logical cluster. However, only one open logical cluster can exist per physical cluster.

Click Next.

8 Select Finish to build the logical cluster.

Creating the cluster using *sybcluster*

This section describes how to use sybcluster to configure and start the cluster. Make sure you first follow the instructions in:

- “Before you begin” on page 32

- "Setting the environment and starting the Unified Agents" on page 34

You can view a complete sybcluster session that includes answers to all the prompts in the Appendix C, "A sybcluster Sample Session."

Note Use sybcluster to create and configure a shared-disk cluster. To create logical clusters, use the Adaptive Server plug-in or system stored procedures described in the "Managing the Workload" chapter of the *Users Guide to Clusters*.

❖ **Configuring the cluster**

1 Start sybcluster. For example:

```
sybcluster -U uafadmin -P -C mycluster -F
"blade1,blade2,blade3"
```

See "The sybcluster Utility" chapter in the *Users Guide to Clusters* for complete syntax and usage information.

2 Enter:

```
create cluster
```

sybcluster prompts you for all necessary information, one parameter at a time. If there is a default value, sybcluster displays it in the command prompt. To accept the default, press Enter. Otherwise, enter the correct value, then press Enter.

3 sybcluster prompts for the following. See "Worksheet for creating a cluster" on page 35.

- The name of the cluster to create if you did not set the default cluster on the command line.
- The maximum number of instances to create for the cluster.
- The number of nodes in the cluster.

sybcluster displays a list of available host machines. In this list are all nodes specified by the -F and -d parameters on the sybcluster command line with a configured and running Unified Agent as set up in "Setting the environment and starting the Unified Agents" on page 34. Select nodes from this list.

Note You can create a simulated cluster by hosting all instances on a single node. For best performance, make sure that the total number of engines across all instances running on the single node does not exceed the number of CPUs, or cores, on the host machine.

Additionally, Sybase strongly recommends tuning the runnable process search count parameter to *'3'* in such an environment. This can be set using the sp_configure stored procedure.

- The number representing the cluster node. sybcluster numbers the remaining nodes in the cluster starting with this number.
- The full path for the quorum device. For example, */dev/raw/raw11*.
- Trace flags.
- The full path for the master database device, including the device name. For example, */dev/raw/raw12*.
- The size of the master database device.
- The size of the master database.
- The page size of the master database, in kilobytes.
- The full path to the system procedure database device, sybsystemprocs. For example, */dev/raw/raw13*.
- The size of the system procedure database device.
- The size of the system procedure database.
- The full path to the Sybase system database device, sybsystemdb. For example, */dev/raw/raw14*.
- The size of the system database device.
- The size of the system database.
- Does the cluster have a private network? (Y or N)
  - If you enter Y to a private network, sybcluster asks:

Do you have a secondary network?

Note If you answer yes (Y), sybcluster asks (in step 4, when configuring the instances) for the protocol address or address of the private network interface on the host system for each instance in the cluster.

- If you enter "N" to a private network, sybcluster asks for:

  The port number to which this range applies.

  Enter a starting port number. sybcluster calculates the number of additional port numbers required so that the instances can exchange messages, and reserves that number of ports, starting with the port number you provide.

  Note Adaptive Server uses several sockets for each interconnection. The number of ports required for each instance is 5 multiplied by the maximum number of instances.

  Make sure the starting port number and the additional port numbers sybcluster requires are not used by another application.

- The *$SYBASE* home directory. For example, */remote/var/sybase*.
- The full path to the ".sh" environment shell script. For example, /*remote/var/sybase/SYBASE.sh*.
- The Adaptive Server home directory. For example, /*remote/var/sybase/ASE-15_0*.
- **Note** Ensure that the interfaces file does not contain the cluster or instance information. It will be added by sybcluster during the Create Cluster process

  The directory containing the interfaces file. For example, */remote/var/sybase*. sybcluster adds the correct cluster and instance information during configuration.
- The full path to the dataserver configuration file. For example, */remote/var/sybase/mycluster.cfg*. sybcluster must be able to find this file during configuration.

4 sybcluster prompts for values for each instance, one node at a time:

- Node name, if you are configuring more than one node.

- Instance name.
- Query port number for the instance. Ensure that it is available and not in use by other applications.
- The primary protocol address for the instance (if you answered Y to a private network). For example, 10.0.1.1.
- The secondary protocol address for the instance (if you answered Y to a secondary private network). For example, 10.0.1.2.
- Local system temporary database device- The name of the Adaptive Server database device for the local system temporary (LST) database.
- The LST device path. For example, */dev/raw/raw15*.
- The LST device size.

  Note If you place all LST databases on the same device, the device size must be large enough for all LST databases.

- The LST database name.
- The LST database size.

5 sybcluster asks if you want to add another instance. If you enter Y, sybcluster repeats step 4 for the next instance.

6 sybcluster asks if you want to save the cluster configuration you have entered.

If you enter Y, sybcluster stores the configuration in an XML file that you can edit and replay using the sybcluster command:

```
create cluster cluster_name file file_name
```

See Appendix C, “A sybcluster Sample Session.”

7 sybcluster asks if you want to create the cluster.

If you enter Y, sybcluster configures the cluster using the information you have entered. This may take several minutes.

Verifying the cluster configuration

❖ **Verifying that the cluster is running and that you can connect to it**

1 If sybcluster is not running, start it. Enter:

```
sybcluster -U uafadmin -P -C cluster_name -F "node_name[:port_num]
[,node_name[:port_num]]..."
```

This statement identifies the default cluster and the Unfied Agents on each node in the cluster. If you do not enter this information on the sybcluster command line, you can enter it in the following step. See the *Users Guide to Clusters* for more information.

2 Connect to the cluster. Enter:

```
connect
```

3 Start the cluster. Enter:

```
start cluster
```

4 Verify that the cluster is running. Enter:

```
show cluster status
```

5 Verify the cluster configuration. Enter:

```
show cluster config
```

Starting and stopping a cluster using *sybcluster*

❖ **Starting a cluster**

1 If sybcluster is not running, start it. The syntax is:

sybcluster -U uafadmin -P -C *cluster_name*
-F "*node_name*[:*port_num*]
[,*node_name*[:*port_num*]]..."

For example, to start “mycluster” on “blade1”, “blade2”, and “blade3”, enter:

```
sybcluster -U uafadmin -P -C mycluster
-F "blade1,blade2,blade3"
```

2 Connect to the cluster. For example:

```
connect
```

3 Start the cluster:

```
start cluster
```

❖ **Shutting down a cluster**

- Enter:

```
shutdown cluster
```

Adaptive Server waits for all statements and transactions to complete before shutting down the cluster.

Starting the cluster when the operating system starts

It is possible to configure the host system operating system to automatically start the Adaptive Server cluster when the operating system starts. This is done by starting the Unified Agent on the host system and then executing the sybcluster commands to start the cluster using a shell script. The following steps are offered as an example:

1 Start UAF

2 Verify that UAF is started successfully.

3 Execute sybcluster and pass a command file to sybcluster containing commands to start the instances on the system.

4 The command file is passed using the sybcluster -i command-line parameter and will look something like this:

```
connect to asece15
start instance asece1
quit
```

5 The sybcluster command line will look something like:

```
sybcluster -U uafadmin -P -F host1:9999,host2:9999 -i asece1_startup
```

where:

- asece1_startup is the command file shown above.

Cleaning up after failed installation

If an unexpected error occurs during configuration of the cluster, some files or operating system processes may be left. It is important to remove these before attempting to create the cluster again.

❖ **Cleanup steps**

1. Terminate srvbuildres or dataserver processes if either are left running.
2. Stop the UAF agents on all nodes.
3. Remove *$SYBASE_UA/nodes/*/plugins/<cluster_name>*
4. Remove any entries for the cluster or cluster instances from the *interfaces* file.
5. Restart the UAF agents.

Configuring auxiliary servers

This section contains instructions for configuring the optional auxiliary servers:

- Backup Server
- Job Scheduler
- XP Server
- Monitor Server

Using Backup Server in a clustered environment

Backup Servers in a Shared Disk Cluster environment are available across all nodes of the cluster. A cluster has a single Backup Server. You can configure the Backup Server on any node of the cluster, and you can provide additional hosts and port numbers on which Backup Server can start if its current node goes down.

You can execute the dump and load commands from any node in the cluster. Once issued, the local instance services the dump and load commands and routes them to the cluster's Backup Server. The instances attempt to connect to Backup Server using the order specified in interfaces file. If Backup Server is not running on any of the nodes in the cluster, the instance on which you issued the dump or load command starts Backup Server, if Backup Server is configured to run on this node in the interfaces file.

You can install and start Backup Server using the sybcluster utility, which prompts you for required information and then performs the installation start up for you. You can also perform all of these steps manually. Both methods are described in this section.

The Backup Server binary (*$SYBASE/$SYBASE_ASE/bin/backupserver*) is installed when you install Adaptive Server version 15.0.1 CE.

Using *sybcluster* to configure Backup Server

To configure Backup Server using sybcluster:

1 Start sybcluster. For example, to start sybcluster and specify Unified Agents on the nodes “blade1”, “blade2”, and “blade3”, enter:

```
sybcluster -U uafadmin -P -F "blade1:1234,blade2:2345,blade3:3456"
```

Note Unified Agents must be running on all nodes in the cluster to verify that the selected ports are available.

2 Connect to the cluster. For example, to connect to “mycluster”, enter:

```
connect to mycluster
```

3 If the cluster is not running, start it. Enter:

```
start cluster
```

4 To create the Backup Server for the cluster, enter:

```
create backupserver
```

Adaptive Server prompts you for these values:

- The Backup Server name. The default value is “*cluster_name*_BS”. For example, “mycluster_BS”.
- The backup server listening port for each node in the cluster.

After creating backup server, you can set optional configuration values by editing the Backup Server configuration file, which is *$SYBASE/<backupserver_name>.cfg*.

Manually installing and starting the Backup Server

To configure the Backup Server manually:

1. Add the Backup Server's information to the interfaces file. Include an entry for each node on which you want Backup Server to start. This example adds a Backup Server named "Cluster_BS":

   ```
   Cluster_BS
        master tcp ether blade1 <port1>
        query tcp ether blade1 <port1>
        master tcp ether blade2 <port2>
        query tcp ether blade2 <port2>
        master tcp ether blade3 <port3>
        query tcp ether blade3 <port3>
   ```

   If the enable backupserver HA configuration option is enabled, then the Backupserver is started on whichever node the instance on which the dump or load command was executed on is running. If enable backspserver HA is not enabled, then the Backupserver is started wherever the end-user decides to start it. If this node is not available, it attempts to start on blade2, and so on.

2. Add the Backupserver to the sysservers table using the sp_addserver stored procedure. For example:

   ```
   sp_addserver SYB_BACKUP, rpcserver, SYB_BACKUP
   ```

3. Add a runserver file for Backup Server. Create the file in the same directory you created the runserver file for each of the instances (typically, *$SYBASE/$SYBASE_ASE/install*). For example, the following is the contents of the runserver file for the Cluster_BS backup server:

```
#!/bin/sh
#
# Backup Server Information:
#  name:                          Cluster_BS
#  errorlog: /sybase/ase1501ce/ASE-15_0/install/backup.log
#  interfaces: /sybase/ase1501ce/ASE-15_0/interfaces/interfaces
#  location of multibuf: /sybase/ase1501ce/ASE-15_0/bin/sybmultbuf
#  language:                      us_english
#  character set:                 iso_1
#  tape configuration file: /sybase/ase1501ce/ASE-15_0/backup_tape.cfg
#
#
/sybase/ase1501ce/ASE-15_0/backupserver -SCluster_BS \
-e/sybase/ase1501ce/ASE-15_0/install/backup.log \
-I/sybase/ase1501ce/interfaces\
-M/sybase/ase1501ce/ASE-15_0/bin/sybmultbuf -Lus_english -Jiso_1 \
-c/sybase/ase1501ce//Cluster_BS.cfg
```

4 'enable backupserver HA' enables the cluster for high availabilty Backup Server. This functionality starts a new Backup Server on the current node if the current node is configured to run Backup Server and no active Backup Server is available.

Disabling this functionlity does not start Backup Server automatically when no active Backup Server is available.

Run sp_configure 'enable backupserver HA' to configure the cluster for a high availability Backup Server functionality:

- sp_configure 'enable backupserver HA', 1 – enables the cluster for a Backup Server.
- sp_configure 'enable backupserver HA', 0 – disables the cluster for a Backup Server functionality.

Configuring Job Scheduler

To set up and start the Job Scheduler, follow directions in Chapter 2, "Configuring and Running Job Scheduler," in the *Job Scheduler Users Guide*. The following section contains changes for the Cluster Edition and replaces "Installing Job Scheduler."

Installing Job Scheduler for the Cluster Edition

All instances in the cluster share a single Job Scheduler. You can set up Job Scheduler so that, in the event the instance on which it is running fails, Job Scheduler can fail over to another node.

❖ **Installing Job Scheduler**

1 Create a device called sybmgmtdev with a size of at least 90MB on a shared raw device that is accessible to all instances in the cluster.

2 Run the *installjsdb* script:

```
isql –Usa –Psa_password –Sservername
–i $SYBASE/$SYBASE_ASE/scripts/installjsdb
```

Note You must have the directory with the location of the isql executable (*$SYBASE/$SYBASE_OCS/bin*) in your path.

The *installjsdb* script looks for the sybmgmtdb database. If it exists, the script creates Job Scheduler tables and stored procedures. If it does not exist, the script looks for a sybmgmtdev device on which to create the sybmgmtdb database, tables, and stored procedures.

Note If the *installjsdb* script finds neither a sybmgmtdev device nor a sybmgmtdb database, it creates a sybmgmtdb database on the master device. Sybase strongly recommends that you remove the sybmgmtdb database from the master device to make recovery easier in the case of a disk failure.

3 Create a directory services entry for the JS Agent in the interfaces file using either dscp, dsedit, or a text editor as appropriate. Sybase suggests that you name the entry "*clustername*_JSAGENT".

To enable HA failover, the JS Agent entry must contain master and query rows for each node in the cluster. For example, to add a JSAGENT entry for the cluster "mycluster" with two nodes, the syntax might be:

```
mycluster_JSAGENT
    master tcp /dev/tcp node_name1 17780
    query tcp /dev/tcp node_name1 17780
    master tcp /dev/tcp node_name2 16780
    query tcp /dev/tcp node_name2 16780
```

The node name must match the name returned by the uname -n command executed at the UNIX prompt. For example, on host "linuxxml1," uname -n returns the value "linuxxml1.sybase.com," and on host "linuxxml2", uname -n returns the value "linuxxml2.sybase.com". The correct entry for JS Agent is:

```
mycluster_JSAGENT
    master tcp /dev/tcp linuxxml1.sybase.com 17780
    query tcp /dev/tcp linuxxml1.sybase.com 17780
    master tcp /dev/tcp linuxxml2.sybase.com 16780
    query tcp /dev/tcp linuxxml2.sybase.com 16780
```

Note You must specify a port not currently in use.

See the *System Administration Guide, Volume 1* for more information on directory services.

4 Using sp_addserver, create an entry in the sysservers table for the cluster. For example:

```
sp_addserver SYB_JSAGENT, null, mycluster_JSAGENT
```

See the *Reference Manual: Commands* for more information on sp_addserver.

5 Enable Job Scheduler:

```
sp_configure "enable job scheduler," 1
```

6 To start Job Scheduler, you can either restart the server, or execute:

```
use sybmgmtdb
go
sp_js_wakeup "start_js", 1
go
```

7 To determine the instance on which Job Scheduler is running, query the global variable *@@jsinstanceid*.

```
select @@jsinstanceid
go
```

Configuring XP Server

You can configure an XP Server for each instance in the cluster using the sybcluster configure xpserver command, which performs the configuration steps for you. You can also configure XP Servers manually, and perform the configuration steps yourself.

Configuring XP Server using *sybcluster*

To configure XP Servers using sybcluster:

1 Start sybcluster. For example, to start sybcluster and specify Unified Agents on the nodes “blade1”, “blade2”, and “blade3”, enter:

```
sybcluster -U uafadmin -P -F "blade1:1234,blade2:2345,blade3:3456"
```

2 Connect to the cluster. For example, to connect to “mycluster”, enter:

```
connect to mycluster
```

3 Start the cluster. Enter:

```
start cluster
```

4 To configure the XP Servers, enter:

```
create xpserver
```

Adaptive Server prompts you for the XP Server port number for each instance.

Configuring XP Server manually

If you configure XP Server for the cluster, you must configure an XP Server for each instance. For each instance:

1 Create an entry for XP Server in your interfaces file for each instance, indicating node name and port number.

Enter the XP Server name in capital letters in this format:

INSTANCE_NAME_XP

The syntax for interfaces file entry is:

*INSTANCE_NAME*_XP
master tcp ether *node_name port_number*
query tcp ether *node_name port_number*

For example:

```
ASE1_XP
master tcp ether blade1 32077
query tcp ether blade1 32079
```

2 Use sp_addserver to enter a row in the sysservers table for each XP Server. Thereafter, the Cluster Edition automatically starts the server. For example:

```
sp_addserver ASE1_XP, NULL, ASE1_XP
```

See the *Utility Guide* for more information about the xp server utility.

Configuring Monitor Server

You can configure a Monitor Server for each instance in the cluster using the sybcluster create monitorserver command.

To configure a Monitor Server for each instance

1 Start sybcluster.

For example, to start sybcluster and specify Unified Agents on the nodes “blade1”, “blade2”, and “blade3”, enter:

```
sybcluster -U uafadmin -P -F "blade1:1234,blade2:2345,blade3:3456"
```

2 Connect to the cluster. For example, to connect to “mycluster”, enter:

```
connect to mycluster
```

3 Start the cluster. Enter:

```
start cluster
```

4 To configure the Monitor Server, enter:

```
create monitorserver
```

For each instance, Adaptive Server prompts you for:

- The Monitor Server port number
- A user name
- A password

CHAPTER 4

Post-installation tasks

This chapter explains post-configuration tasks for setting up the cluster.

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 Cluster Edition as “sa” and set a password using sp_password, where “null” is the default password and *new_password* is the password that you are assigning to the “sa” account:

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

For greatest security, Sybase recommends that you create a password with at least six characters, with a combination of letters and numbers.

If you change the password for the system administrator, or choose to use a different account than the system administrator to administer Adaptive Server, you must also modify the Unified Agent configuration with the new password or account information. You can change the Unified Agent administrator account using the sybcluster 'set cluster login' command. See “The sybcluster Utility” chapter in the *Users Guide* for syntax and usage information.

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 61
- "Installing the interpubs database" on page 62
- "Installing the jpubs database" on page 63

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 4-1 lists the scripts that you can use to install the sample databases.

Table 4-1: 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 2K, 4K, 8K, and 16K 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 the server instance.

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.

Use isql to log in to the instance. From the *$SYBASE/$SYBASE_ASE/scripts* directory, run the script:

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

Where:

- *server_name* represents the destination server for the database
- *script_name* is the 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 10 MB 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), Roman 8, 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 cluster, 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 the cluster'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 60.

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 Cluster Edition 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.

CHAPTER 5

Installing PC-Client

Overview

The PC-Client installer contains client applications and utilities you can use to access the cluster to execute queries or administer the cluster. It also contains the Sybase Open Client/ Open Server Software Developers Kit, which you can use to develop applications that access the cluster, and ODBC, OLE DB, and ADO.NET clients.

Client applications, such as Sybase Central and Open Client, are used to access Adaptive Server.

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 68.

The Sybase PC-Client CD includes:

- Software Developer’s Kit (SDK) for Windows platforms. SDK includes:
  - Open Client (CT-Library, DB-Library)
  - ASE plug-in
  - DBISQL
  - Embedded SQL/C (ESQL/C)
  - Embedded SQL/Cobol (ESQL/Cobol)

- Extended Architecture (XA)
- jConnect for JDBC
- ASE ODBC driver by Sybase
- ASE OLE DB provider by Sybase
- ASE ADO.NET Data provider by Sybase
- Language modules

Before you begin

Have at least an extra 5MB of disk space, a *\temp* directory, and set the 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 components

The PC-Client CD contains several components 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 components that can be installed from the CD and allows you to install them one at a time.

Table 5-1: Minimum System requirements for PC-Client on WindowsXP

Windows OS	Hardware	Minimum RAM
WinXP Pro SP2, Win2003 Server SP1, and WinVista Business Edition	P4 1.0 GHz	512 MB

❖ **Installing PC-Client components for Windows**

PC-Client components can be installed on client computers.

1 Verify that your computer has sufficient disk space for each product.

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

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

4 Select Run from the Windows Start menu to start the installation program, and enter the following, where *X* is the directory into which you have copied the PC-Client installation files:

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

5 Select Client Components 15.0.1.

6 When the installer starts, the Welcome window displays. Click Next.

7 When the Sybase license agreement displays, select a country, click I Agree, then click Next.

8 Enter the directory path and click Next.

9 Select the type of installation to be performed.

- Typical Install – installs the default components that most users need.
- Full Install – installs every component on the CD.
- Custom 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 Custom Install, next specify the components to install.

Components installed in a typical installation are automatically selected. You may select or unselect components from this list.

11 Click Next.

The Summary window displays every component to be installed by the 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 either click Previous to return to the previous window and change your selections, or Cancel to quit the installer.

12 Click Next. The installer displays a progress indicator as it installs components.

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

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 must know where the others reside on the network. This information is stored in the *interfaces* file on Windows or in a Lightweight Directory Access Protocol (LDAP) server.

Configuring *libtcl.cfg* for LDAP

Windows

Use the *libtcl.cfg* files to specify an LDAP server name, port number, DIT base, user name, and password to connect to an LDAP server.

The default *libtcl.cfg* file is located in *%SYBASE%|%SYBASE_OCS%\ini*.

In its simplest form on 32-bit LDAP drivers, the *libtcl.cfg* file is in this format:

```
[DIRECTORY]
ldap=libsybdldap.dll
```

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 and 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 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! Enter the LDAP URL on a single line.

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

For example:

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

2 Verify that the appropriate environment variable points to the required third-party libraries. The Netscape LDAP SDK libraries are located in *%SYBASE%\%SYBASE_OCS%\dll*

The Windows PATH environment variable must include this directory.

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

a From Windows, select Start | Programs | Sybase | Connectivity | Open Client Directory Service Editor.

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%\ini\objectid.dat*.

e Click Add New Network Transport.

- Select the transport type.
- Enter the host name.
- Enter the port number.

f Click OK twice to exit the dsedit utility.

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

Adding a server to the *interfaces* file

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 *interfaces* file on your PC-Client computer:

1 From Windows, select Start | Programs | Sybase | Connectivity | Open Client Directory Service Editor.

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.

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
- *4100* – is the port number the server is using to "listen" for clients.

Note Windows 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 *interfaces* file with any standard ASCII text editor.

---

7 To find the IP address for a machine, use ypmatch on a UNIX box, ipconfig on Windows.

Note You can use Windows Directory Replication to replicate *interfaces* in multiple locations. See the Microsoft documentation for information. You can also use directory services, which are stored in the Registry.

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

Testing the Sybase Central installation

After you install Adaptive Server, Sybase Central, and the Java Runtime Environment, test the installation and network connections:

1 Select Start | Programs | Sybase | Sybase Central v4.3.

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

3 Log in using the default user ID, “sa”, without a password. If you have changed the password, use the new password.

4 Select the Adaptive Server to which you want to connect.

5 To disconnect from a server, select Tools | Disconnect, or right-click the server icon to which are connected and select Disconnect.

6 Exit Sybase Central.

CHAPTER 6

IO Fencing

Introduction

The shared-disk cluster can detect the presence of a noncooperating instance and remove it from the cluster. In rare situations, however, the cluster cannot stop the noncooperating instance from writing to the shared disk, even though the instance is no longer part of the cluster. For example, if an instance has been removed from the cluster, but has not released resources and shut down, it may still be capable of writing to the shared disk. Use IO fencing to prevent the noncooperating instance from making data writes.

Note Sybase cannot guarantee data integrity unless you enable IO fencing. If you do not use IO fencing, data loss or other data corruption may occur in rare instances. Only test and development environments that can accept this risk should be deployed without IO fencing.

In the Cluster Edition, Sybase supports the SCSI-3 Persistent Group Reservation (PGR) feature of SCSI-3 devices to provide IO fencing. PGR is the SCSI-3 standard for managing disk access in an environment where a single disk is shared by multiple hosts for read and write access.

The IO fencing provided by the SCSI-3 PGR feature operates only on devices, not on partitions. For example, */dev/sda1* and */dev/sda2* are partitions of the device */dev/sda*. A fencing operation targeted to a raw device bound to */dev/sda1* affects all partitions of */dev/sda*, so any file systems or other applications (including another Adaptive Server) using partitions on that device are also affected. For this reason, the device must be used exclusively by the cluster instance.

Enabling IO fencing

To enable IO fencing:

- Each instance in the cluster must run on a separate node.
- All storage devices (disks) that hold database devices must support the SCSI-3 standard and cannot be partitioned.

Note SCSI-3 PGR functionality is available only for a physical SCSI disk device or a virtual disk device exported by a storage area network (SAN). Partitioning such a device at the operating system level does not provide SCSI-3 PGR on each partition. Rather, the SCSI-3 PGR, and thus the fencing support, is shared among all partitions on the device. Adaptive Server cannot fence database devices independently.

Sybase recommends that you do not create database devices on partitions that use the same device as they cannot be fenced independently.

The quorum must reside on its own device. Adaptive Server does not allow you to create database devices on the quorum device.

In addition, if device partitions are used outside the cluster, any fencing performed by the cluster denies access to those other partitions for the external applications or file systems placed on them.

IO fencing is based on a device driver API that is targeted to a specific device driver. The device driver is typically called the fence device. Use operating system commands to create the fence device on each of the nodes running the cluster.

To turn on the enable i/o fencing configuration parameter, enter:

```
sp_configure "enable i/o fencing", 1
```

The SCSI-3 PGR feature is platform-dependent, and all devices used by the Cluster Edition should have this functionality. See your operating system documentation for complete syntax and setup information.

As part of the cluster creation process, both the Adaptive Server plug-in and sybcluster let you check that each device is enabled for IO fencing. You can also run the qrmutil utility:

```
$SYBASE/$SYBASE_ASE/bin/qrmutil
```

Note Configuring IO fencing affects all devices with IO fencing capability system-wide.

Setting up IO fencing

This section discusses how to set up IO fencing for different platforms.

Setting up IO fencing on Linux

To enable IO fencing on Linux:

- Use operating system commands to load a SCSI generic, (sg), driver on each node running the shared-disk cluster. Loading the driver automatically creates */dev/sg** files.
- Grant read and write permission to each instance for all */dev/sg** files created on that node by the sg driver.

  For example, if the database device on the raw partition */dev/raw/raw1* is mapped to */dev/sg3*, grant write permission on the */dev/sg3* file.

Make sure the Cluster Edition has write access to */dev/sg* files that correspond to configured database devices. In addition, the fence device must have write access to */dev/sg* files containing SCSI-3 PGR commands.

The path for the fence device may be different from the corresponding physical device path. For example, the database device might be */dev/raw/raw1*, the physical device */dev/sda*, and the fence device */dev/sg0*.

Note Whenever the sg driver is loaded at system startup (or through user command), the system re-creates the */dev/sg** files. Make sure that permissions are granted appropriately on these files as part of the startup process. Otherwise, the system administrator must manually reset permissions each time the system starts.

Setting up IO fencing on Solaris

To enable IO fencing on Solaris, the UNIX user starting the Cluster Edition must be granted SYS_DEVICES inheritable privileges to enable access to the raw devices (*/dev/raw/raw#* or */dev/rdsk/c#t#d#s#*) used by the Cluster Edition for database and quorum devices. SYS_DEVICES allows the Cluster Edition to execute the SCSI-3 PGR commands used for IO fencing.

Permanent SYS_DEVICES privileges can be granted to the user's set of inheritable privileges. For example:

```
sudo ppriv -s I+sys_devices $$
```

Temporary SYS_DEVICES privileges can be granted to the current user's shell process. For example:

```
usermod -K defaultpriv=basic, sys_devices mylogin
```

See your operating system documentation for complete syntax and usage information.

CHAPTER 7

Upgrading from an Earlier Version

Overview

You can upgrade to the Cluster Edition from:

- 12.5 through 12.5.3 ESD #6
- 15.0 through 15.0.1 ESD #3

You cannot directly upgrade from 12.5.4 including ESDs nor can you directly downgrade from 15.0.2. If you have any of these versions and want to install the Cluster Edition you must first:

- Downgrade from 12.5.4 to 12.5.3 ESD #6 by following the instruction in “Downgrading Adaptive Server 12.5.4.”Downgrade from 15.0.2 to 15.0.1 ESD #3 by following the instructions in “Downgrading from Adaptive Server 15.0.2.”
- Once you have completed the downgrade process, you must follow the steps outlined in this chapter for upgrading to the Cluster Edition.

To upgrade from 15.0.1 to the Cluster Edition, you must use the upgrade procedures described in this manual. For more information on migrating from unsupported version see “Migrating to the Cluster Edition from Unsupported Editions” on page 99.

Only upgrades to and from the same page size are supported. Use bcp to re-create the schema and load data from one page size to another.

You can upgrade the server manually, or use the sybcluster utility, which performs many of the preupgrade and upgrade steps for you.

- To use sybcluster to upgrade the server, go directly to “Using sybcluster to upgrade to the Cluster Edition” on page 90.
- To upgrade the server manually, follow the instructions in this and succeeding sections.

See “Migrating to the Cluster Edition” on page 97 for information about migration.

Manually upgrading Adaptive Server consists of six processes:

1 Perform pre-upgrade tasks.

2 Back up your database.

3 Install the new server into its own installation directory.

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

4 Run preupgrade from the Cluster Edition release location.

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

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

Note You see advisory warnings when configuration parameters are not set to the default. These do not require any action.

6 Perform post-upgrade tasks.

System catalog changes during upgrade

The Cluster Edition introduces new system catalogs and changes to existing catalogs.

The catalog upgrade in the Cluster Edition may affect your existing application. If you are upgrading, see the "System Changes" chapter of the *Users Guide* for a complete list of catalogs that are affected.

Statistics in system tables

When you upgrade a server to the Cluster Edition, index and table-level statistics are inaccurate.

Execute update index statistics on the following system tables:

- sysobjects
- sysindexes
- syscolumns
- systypes
- syslogins
- sysusers

Pre-upgrade tasks

Before you begin the upgrade process, install the Cluster Edition onto your system in a new location.

Note The upgrade process modifies the sysaudits tables in the sybsecurity database. Sybase recommends that you archive any auditing data and truncate these tables before you upgrade. This reduces the chances of a failed upgrade due to lack of space in the sybsecurity database.

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. This list summarizes the preupgrade tasks; there is additional information for some of these tasks in the sections that follow.

1 Check system and upgrade requirements.

2 Check the location of the *RUN_server* file.

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 Store procedure text in the syscomments table is required for upgrade. See "Procedure text is required for upgrade" on page 81.

5 Resolve reserved words using quoted identifiers. See "Reserved words" on page 82.

6 Verify users are logged off.

7 Check for database integrity.

8 Back up the databases.

9 Dump the transaction log.

10 Ensure that master is the default database for the "sa" user.

11 Prepare the database and devices for upgrade.

- Disable auditing
- Archive auditing data and truncate auditing tables.
- Disable disk mirroring.

  Note Adaptive Server Enterprise 15.0.1 Cluster Edition ESD#1 does not support disk mirroring.

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

12 If the *$SYBASE* directory is not the same as the old *$SYBASE* directory, copy the following files from earlier versions of Adaptive Server to their corresponding Cluster Edition installation:

- *$SYBASE/interfaces*
- *$SYBASE/<servername>.cfg*

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

Checking system and upgrade requirements

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

2 Determine whether the version of your server is upgradeable to the Cluster Edition.

3 Verify that you unloaded Adaptive Server 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.

4 Verify that your operating system is at the proper version level and has all operating system patches needed for the Cluster Edition.

Checking the *runserver* file location

Verify the name and location of the *runserver* file. Be sure the *runserver* file for your current server is located in *$SYBASE/$SYBASE_ASE/install/RUN_server_name*.

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 deleted text from syscomments, drop and then recreate the stored procedures to again add the text.

Sybase recommends that you use the sp_hidetext stored procedure to hide text, rather than deleting it.

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. 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 Before performing the upgrade, use sp_renamedb to change the name of any user database that is a reserved word.

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.

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

See the *Reference Manual:Building Blocks* for a complete list of reserved words.

Running a reserved word check

Run the reserved word check on the Cluster Edition:

1 Install the Cluster Edition version of *installupgrade* (where *$SYBASE* and *$SYBASE_ASE* are the values for the Cluster Edition):

```
isql -Usa -Ppassword -Sserver_name
-i$SYBASE/$SYBASE_ASE/scripts/installupgrade
```

2 Install the Cluster Edition version of *usage.sql*:

```
isql -Usa -Ppassword -Sserver_name
-i$SYBASE/$SYBASE_ASE/upgrade/usage.sql
```

3 Log in to the Cluster Edition and execute sp_checkreswords on all of databases. For example, log in to the instance and execute:

```
use sybsystemprocs
go
sp_checkreswords
go
```

4 Correct any errors from the reserved word check.

Addressing reserved words conflicts

Use sp_dboption to set the database to single-user mode, and then run sp_renamedb, specifying the new name. See the *Reference Manual:Building Blocks* 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.
- Use brackets around the identifier. For example:

```
create table [table] ( [int] int, [another int] int )
```

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*.

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 preupgrade, the utility scans the system catalogs and calculates how much additional space you need to perform the upgrade.

If you did not run preupgrade, you must manually calculate how much free space to add to your system catalogs and databases.As a general rule, double the size of the largest catalog that you are going to upgrade, and add approximately 10 percent more. 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**

Use sp_helpdb or Sybase Central 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 The model database cannot be larger than the tempdb. First increase the size of tempdb, then increase the size of the model database.

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

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 large enough, you do not need to make any changes. If the database has insufficient space for upgrade, increase the size with alter database. Use preupgrade to determine the database size.

If you do not have a *sybsystemdb* database

If you do not have a sybsystemdb, and will not be using two-phase commit, create a sybsystemdb with a minimum of 4MB. The size you need depends on your page size.

sybsystemprocs

sybprocsdev is the default name for this device. However, it is frequently referred to as the sybsystemprocs device in Adaptive Server.

Verify that the sybsystemprocs database is large enough. For an upgrade, the recommended minimum size for sybsystemprocs is the larger of 124MB, 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.

Manually upgrading to the Cluster Edition

Note Start the Cluster Edition in a single instance until the upgrade is complete.

1 Back up all old databases.

2 Start the earlier version of Adaptive Server:

- Move to the old $SYBASE directory

```
cd $SYBASE
```

- Source *SYBASE.sh* (Bourne shell) or *SYBASE.csh* (C shell).

```
source SYBASE.csh
```

- Execute the runserver file:

```
$SYBASE/$SYBASE_ASE/install/RUN_server_name
```

3 In another window, change to the new *$SYBASE* directory

4 Source *SYBASE.sh* (Bourne shell) or *SYBASE.csh* (C shell) in the new *$SYBASE* directory:

```
source SYBASE.csh
```

5 Run the preupgrade test on the old server using the preupgrade utility, located at *$SYBASE/ASE-15_0/upgrade*. The server configuration resides in *$SYBASE/$SYBASE_ASE* in the old release directory.

Do not change the default packet size from 512 to 2048 until after the upgrade is complete.

Note If during pre-upgrade the default network packet size is set to 2048 then you cannot login to finish the pre-upgrade on a 12.5.x server because there is no way to tell preupgrade to use 2048 bytes as a packet size.

a Execute the following, where *password* is the system administrator's password:

```
$SYBASE_ASE/upgrade/preupgrade -Sserver_name -Ppassword
```

b Correct all errors from the output of the preupgrade test. Re-run preupgrade until it succeeds without errors.

c Restart the old Adaptive Server, if required.

6 Run the reserved word check on the Cluster Edition:

a Install the Cluster Edition version of *installupgrade:*

```
isql -Usa -Ppassword -Sserver_name
-i$SYBASE/$SYBASE_ASE/scripts/installupgrade
```

b Install the Cluster Edition version of *usage.sql*:

```
isql -Usa -Ppassword -Sserver_name
-i$SYBASE/$SYBASE_ASE/upgrade/usage.sql
```

c Log in to the old Adaptive Server and execute sp_checkreswords on all databases:

```
use sybsystemprocs
go
sp_checkreswords
go
```

d Correct any errors the reserved word check reveals.

7 Shut down the old Adaptive Server.

8 Create the cluster input file. For example the file 'mycluster.inp':

```
[cluster]
name = mycluster
max instances = 2
master device = /dev/raw/raw101
config file = /sybase/mycluster.cfg
interfaces path =
traceflags =
primary protocol = udp
secondary protocol = udp

[management nodes]
hostname = blade1
hostname = blade2

[instance]
id = 1
name = ase1
node = blade1
primary address = blade1
primary port start = 38456
secondary address = blade1
secondary port start = 38466
errorlog = /sybase/install/mycluster_ase1.log
interfaces path = /sybase/
traceflags =
```

```
additional run parameters =
```

For an example of what this input file must contain, see “Creating the cluster input file” on page 159 for more information.

9 Copy the old Adaptive Server configuration file (*server_name.cfg*) from the old *$SYBASE* directory to the new *$SYBASE* directory

10 Add an additional entry to the interfaces file for each of the instances in your cluster input file (described in Step 9). See “Configuring the interfaces file” on page 163 for more information.

11 Determine the raw device used for the quorum device. For the version of the Cluster Edition, use a raw device on shared disks. Do not use a file-system device.

12 Create the quorum device and start the new instance with the old master device:

```
$SYBASE/$SYBASE_ASE/bin/dataserver\
--instance=ase1\
--cluster_input=mycluster.inp\
--quorum_dev=/dev/raw/raw101
--buildquorum
```

Note The *instance_name* you indicate with the --instance parameter must be the name of one of the instances in the new cluster, and the interfaces file must contain an entry for this instance. Any additional options such as -M must be present in the RUN_FILE as dataserver won’t read them from the quorum. For complete dataserver documentation see the *Users Guide*.

13 Run the upgrade utility:

```
$SYBASE/$SYBASE_ASE/upgrade/upgrade -S
instance_name -Ppassword
```

The *instance* you include with the --instance parameter is the name of the instance in your cluster that was started in step 13.

14 Log in to the instance. Create the local system temporary database devices and local system temporary databases for each of the instances in your cluster. The syntax is:

create system temporary database *database_name* for instance *instance_name* on *device_name* = *size*

See “Setting up local system temporary databases” on page 166 for more detailed information.

15 Shut down the instance. Log in to the instance with isql and issue:

```
shutdown instance_name
```

16 Restart the cluster.

```
$SYBASE/$SYBASE_ASE/bin/dataserver \
--instance=ase1\
--quorum_dev=/dev/raw/raw101\
```

17 Log in to the Cluster Edition and execute sp_checkreswords on all of databases. For example, log in to the instance and execute:

```
use sybsystemprocs
go
sp_checkreswords
go
```

18 Correct any errors from the reserved word check.

19 Copy and modify the old *run_server* file to new directory. You must edit it to point to binaries in the correct *$SYBASE* directories:

- Add this argument to the *run_server* file:

  --quorum_dev=<*path to the quorum device*>

- Remove these options, as the information is now stored in the quorum device.
  - -c
  - -i
  - -e

See “Creating the runserver files” on page 165 for more information.

20 Start the instance:

```
cd $SYBASE/$SYBASE_ASE/install
startserver -fRUN_server_name
```

21 Install the system procedures:

```
isql -Usa -Ppassword -Sserver_name
-i$SYBASE/$SYBASE_ASE/scripts/installmaster
```

22 If Adaptive Server includes auditing, run *installsecurity*:

```
isql -Usa -P password -S server_name
-i$SYBASE/$SYBASE_ASE/scripts/installsecurity
```

23 Run *installcommit*:

```
isql -Usa -Ppassword -Sserver_name
-i$SYBASE/$SYBASE_ASE/scripts/installcommit
```

Using sybcluster to upgrade to the Cluster Edition

This section describes the upgrade process using the sybcluster utility. sybcluster performs many of the upgrade steps for you. You can also upgrade to the Cluster Edition by performing all the upgrade steps yourself. See "Manually upgrading to the Cluster Edition" on page 86.

sybcluster is a command line–based utility that allows you to create and manage a cluster. sybcluster uses the Unified Agent Framework (UAF) to "plug in" to the Unified Agent on each node in the cluster. The Unified Agent processes the sybcluster commands that let you manage the cluster. See the *Users Guide* for detailed information about sybcluster and the UAF.

This section describes three task sets for upgrading your server:

1 Before you begin – complete these tasks before you upgrade your server.

2 Verifying the server for upgrade – make sure that your server is ready for upgrade.

3 Upgrading the server – upgrade the server to Adaptive Server Cluster Edition version 15.0.1 ESD#1.

In addition, after completing the upgrade, perform the tasks described in "Post-upgrade tasks" on page 96. These tasks must be performed whether you upgrade the server manually or using sybcluster.

Before you begin

Before you start the upgrade:

- Back up old databases.
- Install the Cluster Edition. Make sure that the nonclustered Adaptive Server that is to be upgraded and the Adaptive Server Cluster Edition version 15.0.1 ESD#1 installation directories are located on the same machine. This is the upgrade machine.

- Review Chapter 3, “Creating and Starting the Cluster,” and the “Worksheet for creating a cluster” on page 35 for a list of the information you need for the upgrade.

Verifying the server for upgrade

This procedure tests to make sure the server is ready for the upgrade process. It does not actually perform any upgrade steps.

❖ **Using sybcluster to verify the server for upgrade**

1 Start sybcluster. For example, enter:

```
sybcluster -U uafadmin -P -F "mynode:8999"
```

Where:

- -U – is the login name for the Unified Agent. “uafadmin” is the default value.
- -P – is the Unified Agent password. The default password for “uafadmin” is null or blank.
- -F – specifies the node on which the Adaptive Server is running and the listening port for the Unified Agent. The default value is “9999”.

  In this example, the node name is “mynode” and the UAF listening port is “8999”.

  Note You must specify the node on which the upgrade is to take place.

2 Start the verification. From the sybcluster command line, enter:

```
upgrade server server_name checkonly
```

For example:

```
upgrade server myserver checkonly
```

3 Respond to the upgrade server checkonly prompts. Default values are displayed in square brackets. The prompts are:

a Enter the name of the cluster: [*server_name*]

b Enter the existing Sybase installation directory for server *server_name*:

Enter the complete path to the Sybase installation directory. For example, enter "*/myserver/sybase15*".

c Enter the name of the subdirectory containing the ASE installation for server *server_name*. For example, enter "ASE-15_0".

d Enter the name of the subdirectory containing the OCS installation for server *server_name*. For example, enter "*OCS-15_0*".

e Enter the name of an sa login on server *server_name*. For example, enter "sa".

f Enter a password:

Enter the password for the sa login.

g Enter the maximum number of instances: [4]

h How many nodes (hardware hosts) will participate in this cluster. [1]

i Enter the number representing the cluster node 1. [1]

j Enter the full path to the quorum disk.

k Enter any traceflags.

l Enter the full path to the existing master device.

m Does this cluster have a private network. [Y]

n Enter the port number from which this range will be applied. [15100]

o Enter the SYBASE home directory.

Note sybcluster provides default locations for steps 15 through 19.

p Enter the environment shell script path.

q Enter the ASE home directory.

r Enter the interfaces directory.

s Enter path to the dataserver config file.

t Enter the name of the first instance in this cluster.

Note The upgrade process creates a cluster with a single instance.

u Enter the interface file query port number for *instance_name*.

v Enter the LST device name.

w Enter the LST device path.

x Enter LST device size (MB). [100]

y Enter the LST database name.[*cluster_name*_tdb_1]

z Enter the LST database size (MB).[100]

aa Would you like to save this configuration information in a file? [Y]

Note If the verification succeeds, you can use this file as input to upgrade the server.

ab Enter the name of the file to save the cluster creation information.For example: */data/myserver_ce.xml*.

ac Create the cluster now? [Y]

If you enter "y", sybcluster starts the verification process. It displays informational messages as the process proceeds. It does not perform any of the upgrade steps.

4 If the verification process succeeds, you can upgrade the server. See "Upgrading the server" on page 93.

5 If sybcluster displays error messages, resolve all error messages resulting from the check before upgrading the server.

Upgrading the server

After successfully completing the verification procedure (see "Using sybcluster to verify the server for upgrade" on page 91) you can upgrade the server. Perform the upgrade process:

- Using an input file. If you save the configuration information after completing a successful verification, you can use the saved file as input to the upgrade process.
- Interactively by responding to prompts displayed on the screen. These prompts are displayed in "Using sybcluster to verify the server for upgrade" on page 91. See Appendix C, "A sybcluster Sample Session," for sample input.

❖ **Upgrading the server using an input file**

1 Start sybcluster. For example, enter:

```
sybcluster -U uafadmin -P -F "mynode:8999"
```

2 Start the upgrade. For example, enter:

```
upgrade server myserver file /data/myserver_ce.xml
```

In this example, */data/myserver_ce.xml* is the configuration file created during the validation procedure.

Note The configuration file is similar to that used by sybcluster to create a cluster. See Chapter 3, "Creating and Starting the Cluster."

3 sybcluster displays these prompts:

1 Enter the name of an sa login on server *server_name* [sa]

Press Enter to accept the default, or enter the name of another user with sa priviliges on the target server.

2 Enter a password for the sa login.

sybcluster begins the upgrade procedure, displays informational messages as the upgrade proceeds, and confirms the success of the upgrade.

❖ **Upgrading the server interactively**

1 Start sybcluster. Enter:

```
sybcluster -U uafadmin -P -F "node_name:port_number"
```

2 Start the upgrade. Enter:

```
upgrade server server_name
```

3 Respond to the upgrade server prompts. Default values are displayed in square brackets. The prompts are:

1 Enter the name of the cluster. [*server_name*]

2 Enter the existing Sybase installation directory for server *server_name*.

Enter the complete path to the Sybase installation directory. For example, enter "*/myserver/sybase15*".

3 Enter the name of the subdirectory containing the ASE installation for server *server_name*.

For example, enter "*ASE-15_0*".

4 Enter the name of the subdirectory containing the OCS installation for server *server_name*.

For example, enter "*OCS-15_0*".

5 Enter the name of an sa login on server *server_name*.

6 Enter the password for the sa login.

7 Enter the maximum number of instances. [4]

8 How many nodes (hardware hosts) will participate in this cluster. [1]

9 Enter the number representing the cluster node 1. [1]

10 Enter the full path to the quorum disk.

11 Enter any traceflags.

12 Enter the full path to the existing master device.

13 Does this cluster have a private network. [Y]

14 Enter the port number from which this range will be applied. [15100]

15 Enter the SYBASE home directory:

Note sybcluster provides default locations for steps 15 through 19.

16 Enter the environment shell script path.

17 Enter the ASE home directory.

18 Enter the interfaces directory.

19 Enter path to the dataserver config file:

20 Enter the name of the first instance in this cluster:

Note Create a cluster with a single instance; add instances later on using the add instance command.

21 Enter the interface file query port number for *instance_name*.

22 Enter the LST device name.

23 Enter the LST device path.

24 Enter LST device size (MB).[100]

25 Enter the LST database name. [*cluster_name*_tdb_1]

26 Enter the LST database size (MB).[100]

27 Would you like to save this configuration information in a file? [Y]

28 Enter the name of the file to save the cluster creation information.

29 Create the cluster now? [Y]

If you enter “Y”, sybcluster upgrades the server, displaying informational messages as the process proceeds.

Post-upgrade tasks

Note After you upgrade to the Cluster Edition, you do not need to run update statistics on user tables. The upgrade process does not change the existing statistics.

To verify that your new Adaptive Server has been correctly upgraded:

1 Restore functionality in Adaptive Server.

2 Re-enable auditing.

Deploying a cluster agent for an upgraded server

After the upgrade process completes, Sybase recommends that you configure the Unified Agent to manage your cluster. This allows you to use the sybcluster utility and the Adaptive Server Plug-in to manage your cluster. To deploy a new agent for your cluster:

1 Start the Unified Agent. See “Setting the environment and starting the Unified Agents” on page 34.

2 Run the sybcluster utility. See “Creating the cluster using sybcluster” on page 44.

3 Use the sybcluster deploy plugin command to configure the Unified Agent for your cluster.

Recovering from a failed upgrade

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

  ```
  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 stops and waits 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.

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

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

Migrating to the Cluster Edition

Note See the *Migration Guide* for information about migrating from earlier versions to Adaptive Server version 15.0. See the *Installation Guide for Linux* for background and specific information about migrating Adaptive Server and upgrading your database.

Using dump and load commands

For complete instructions about using dump and load to migrate to the Cluster Edition, see the Adaptive Server 15.0 installation guide for your platform.

The basics steps are:

1 Run dbcc checks on all databases in your original server and make sure they run cleanly.

2 Create a Cluster Edition server in a new directory.

3 Create devices and databases to match those in the Cluster Edition 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 from the original server.

5 Load the databases to the Cluster Edition server.

6 If you have partitioned tables, update partition statistics.

7 Run dbcc checks on the Cluster Edition server and make sure they run cleanly.

Migrating from Adaptive Server version 15.0 or earlier using bcp

You can use the bcp utility to migrate to the Cluster Edition from any Adaptive Server. See the Adaptive Server 15.0 *Migration Guide*, the *Installation Guide* for your platform, and the *Utility Guide* for complete instructions.

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* for syntax and usage information.

The basic steps are:

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 Cluster Edition 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 Cluster Edition server and make sure they run cleanly.

Migrating to the Cluster Edition from Unsupported Editions

If you are migrating to the Cluster Edition from Adaptive Server version 15.0.2, begin by downgrading from version 15.0.2 to version 15.0.1 ESD #3. During this downgrade, you must perform one additional downgrade step that is not reflected in the instructions for downgrading from Adaptive Server 15.0.2. Do this after downgrading to Adaptive Server 15.0.1, after following the "Basic downgrade steps" section of “Downgrading from Adaptive Server 15.0.2,”, but before performing any of the "Post-downgrade tasks".

The text of system message 13152 changed between version 15.0.1 and 15.0.2. Adaptive Server incorporates this message text into certain stored procedures as it compiles them, and the 15.0.2 text causes parsing errors during 15.0.1 procedure compilation. To assure successful migration to the Cluster Edition, you must change this message text before attempting to reinstall stored procedures in version 15.0.1.

To change this message back to its 15.0.1 form, log in to Adaptive Server with 'sa_role' permissions, then do this:

```
-- permit direct updates to system catalogs
select config_admin(1, 102, 1, 1, NULL, NULL)
go

-- revise message 13152 text
update master.dbo.sysmessages
set description = "--" + description
here error = 13152
    and langid is null
    and substring(description,1,2) != "--"
go

-- turn off direct updates to system catalogs
select config_admin(1, 102, 0 1, NULL, NULL)
go
```

Migrating an HA-configured Adaptive Server

Note See *Using Sybase Failover in a High Availability System* for more information about Adaptive Server for high availability.

If you are migrating from an Adaptive Server that is configured with Sybase active-active or active-passive high availability, perform the following before upgrading to the Cluster Edition:

1. Use the appropriate cluster subsystem command to stop monitoring resources associated with Adaptive Server on each node.
2. If Adaptive server is configured for active-active high availability, drop the high-availability companionship:
   - For an asymmetric configuration – issue this command on the secondary companion:

     ```
     sp_companion primary_companion_name, "drop"
     ```

   - For a symmetric configuration – issue this command on the primary and the secondary companion:

     ```
     sp_companion companion_name, "drop"
     ```

3. If you are running an active-active configuration, issue this command on both companions to verify that both are in single-server mode:

   ```
   sp_companion
   ```

   Each companion issues this message if it is in single-server mode:

   ```
   Server 'server_name' is currently in 'Symmetric
   normal' mode.
   ```

4. Remove the resources in the cluster subsystem created for high availability such as resource groups. These are not necessary in the Cluster Edition.
5. Disable the configuration option enable HA on the companion server. Log in to the companion and issue:

   ```
   sp_configure "enable HA," 0
   ```

   In an active-active configuration, you must do this on both companions.
6. Follow the steps described in the following sections to upgrade Adaptive Server to the Cluster Edition.

CHAPTER 8

Downgrading from Adaptive Server 15.0.2

This chapter provides instructions for downgrading from Adaptive Server 15.0.2.

Downgrading from 15.0.2

This section describes how to downgrade from Adaptive Server 15.0.2 to an earlier Adaptive Server 15.0 or 15.0.1 release. Downgrading to a 12.5.x or earlier release is not supported. Instructions to downgrade to a 15.0 or 15.0.1 version include:

- Basic downgrade steps using new system procedure sp_downgrade.
- Post-downgrade tasks.
- Information regarding downgrading if you used 15.0.2 features.

- Returning to Adaptive Server version 15.0.2.

Note Adaptive Server 15.0.2 is not an upgrade release if you are starting from a 15.x version of Adaptive Server. For all pre-15.x versions Adaptive Server 15.0.2 is an upgrade release. Also, Sybase recommends that you do not use the new features offered in Adaptive Server version 15.0.2 until you are committed to staying with 15.0.2.

Note Downgrade is only available from Adaptive Server 15.0.2 to Adaptive Server versions 15.0 and 15.0.1. You cannot downgrade directly to version 12.5.x or earlier. If you must downgrade to a release earlier than 15.0, you need to restore databases from dumps made on that older server.

Basic downgrade steps

This section describes the basic steps for downgrading from Adaptive Server 15.0.2 to a 15.0 or 15.0.1 version. Additional actions are required if you used 15.0.2 features. If that is the case, follow the steps in the "Additional steps if you used 15.0.2 features before downgrading," section of this chapter.

Adaptive Server 15.0.2 provides a new procedure called sp_downgrade that is used to help with the downgrade process. This procedure requires "sybase_ts_role" to be on. The "sa_role" or "sso_role" permissions are expected when running this procedure. For more detailed information on this new system procedure and its syntax see "The New System Procedures" section in the 'System Changes" chapter of the *Adaptive Server New Feature Guide for 15.0.2*.

Note Returning to 15.0.1 ESD#3 is preferable, because this version can handle a number of error conditions that may arise as a result of downgrading for several reported errors.

If you return to a pre-15.0.1 ESD3 version of Adaptive Server, no error handling is provided. Using the new features can result in inexplicable behavior, misleading error messages and stack traces.

1 Back up all of your databases and the *$SYBASE* release area prior to the 15.0.2 downgrade.

2 Start the 15.0.2 server in single user mode using dataserver -m option to guarantee that no other user can access Adaptive Server while you are going through the downgade steps. See *Utility Guide* for more details on starting the server in single user mode.

3 Make sure Adaptive Server 15.0.2 is prepared for downgrade by executing.

```
sp_downgrade 'prepare', <'version'>
```

The value of "version" may be written as "15.0" or "150" or "15.0.1" or "1501". However, the version entered should be the Adaptive Server version you want to downgrade to.

Note If you have used the 15.0.2 features for encrypted columns and

- You are reverting back to Adaptive Server 15.0 GA or Adaptive Server 15.0 ESD #1 you must specify the version as "15.0".
- You are reverting back to Adaptive Server 15.0 EC, 15.0 ESD #2, 15.0.1 or later you must specify the version parameter as "15.0.1".

---

sp_downgrade 'prepare' validates the readiness of Adaptive Server 15.0.2 for downgrade. This means that sp_downgrade 'prepare' may print messages regarding manual changes required before the downgrade can be completed. Errors reported must be corrected and implications of warnings reported must be understood before you can proceed with further downgrade steps. Refer to Table 8-1 to get more information on correcting the issues reported by sp_downgrade 'prepare'.

4 Repeat step 3 until sp_downgrade 'prepare' reports no errors. Your system is now ready to downgrade.

5 Begin the downgrade process by executing:

```
sp_downgrade 'downgrade', <'version'>
```

The version number should be the same as the version you entered in step 3. Once the above command completes successfully there can be no more activity on the 15.0.2 server. You must immediately shutdown the 15.0.2 server.

6 Run - checkpoint and shutdown the old server.

7 Copy the *RUN_SERVER* file to the 15.0 or 15.0.1 release area you are downgrading to. Modify the *RUN_SERVER* file to use the dataserver binary from the 15.0 or 15.0.1 release.

8 Restart the old server using the modified *RUN_SERVER* file.

Note If a downgraded server is booted using the 15.0.2 configuration file, the new options will cause the message 'Unknown parameter '. The unknown option will be reported only the first time the server is booted.

9 Depending upon whether you are starting from a fresh installation or from a previously upgraded 15.0.2 installation:

- Run previously saved *restore_msgs.ebf* or *instmsgs.ebf* from 15.0.1 if none were saved, before running installmaster for the Cluster Edition.

These are the tasks needed for downgrade to Adaptive Server 15.0 or 15.0.1 if you have not used any 15.0.2 functionality. If you have used 15.0.2 functionality then follow the additional steps described in the rest of the section.

Post-downgrade tasks

After you perform the basic downgrade steps, some changes from the 15.0.2 upgrade still persist.

Component Integration Services compatibility

Sybase certifies that each version of Adaptive Server can connect to earlier releases through Component Integration Services. Component Integration Services is tested and certified to connect to earlier versions of Adaptive Server.

Sybase does not certifiy that Component Integration Services running on earlier versions of Adaptive Server can connect to later versions. You may encounter errors if an ealier version of Adaptive Server maps a proxy table to a later version and the remote table uses functionality not available in the earlier release.

For example, if you have a local and a remote server, both running Adaptive Server version 15.0, be sure to upgrade the local server first, if you are performing an upgrade on each of them. If one is to be upgraded but not the other, the local server should be upgraded first.

System stored procedures

After you run *installmaster*, *installcommit*, *installsecurity*, *installhasvss*, and *installmsgsvss* on your downgraded server, any changed system stored procedures are returned to their earlier form. Any new stored procedures introduced to Adaptive Server in releases subsequent to 15.0 or 15.0.1 are not removed. Attempts to execute such stored procedures against an older binary will have unpredictable results.

System messages

During upgrade from a previous 15.0 or 15.0.1 version to Adaptive Server 15.0.2 you should have created an *restore_msgs.ebf* file. Now you must run this script to restore your messages back to the version from which you upgraded.

```
isql -Usa -P<sa password> -S<server name> -i<restore_msgs.ebf>
```

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.

System tables

After returning to a pre-15.0.2 binary:

- If during the upgrade process you ran the 'update all statistics' on syslogins then you must delete statistics for syslogins and recreate them.
- spt_values is dropped and re-created when you run the 15.0 or 15.0.1 installmaster, so any new types are eliminated from this table.

Configuration parameters

New configuration parameters have been added in 15.0.2. If a downgraded server is booted using the 15.0.2 configuration file, the new options will cause the message 'Unknown parameter' the unknown option will be reported the first time the server is booted. These messages can be ignored because the configuration file will be rewritten without the unknown options.

Running the 15.0 or 15.0.1 version of installmaster removes configuration parameters that belong to 15.0.2, 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

Keywords decrypt_default, xmltable, path, were added in Adaptive Server 15.0.2, making it impossible to create identifiers using these names. You must change applications if you used these names.

See "Additional steps if you used 15.0.2 features before downgrading" on page 106 for more information.

Additional steps if you used 15.0.2 features before downgrading

If you are rolling back after having used any of the 15.0.2 features, additional steps can be necessary. You should take steps before you downgrade to 15.0 or 15.0.1, and others should be performed immediately after downgrading. You need to read the documentation on the features below that may require manual changes.

Table 8-1: List of topics and references for Unix

To correct errors related to	Refer to
Encrypted Columns	Chapter 2” Encrypted Columns” in Adaptive Server 15.0.2 *New Feature Guide*
User Passwords	Chapter 10 “Supported Security Features” in Adaptive Server 15.0.2 *New Feature Guide*
Instead of Triggers	Chapter 22, “Instead of Triggers” in Adaptive Server 15.0.2 *New Feature Guide*
SQL user defined functions	Chapter 17 “User Defined SQL Functions” in Adaptive Server 15.0.2 *New Feature Guide*
Archived Databases	Chapter 4 “Archived Database Access” in Adaptive Server 15.0.2 *New Feature Guide*
Hash built-ins	Chapter 19 “Changes to System Procedures, Functions and Commands” in Adaptive Server 15.0.2 *New Feature Guide*

In general, no additional steps are required when you are returning to an Adaptive Server version in which the feature was already available.

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 Users Guide*.

New sort orders

If a new nocase sortorder for Chinese or Japanese character sets is configured as Adaptive server's default sortorder. Before downgrading to a 15.0 or 15.0.1 release, switch to a sortorder that is compatible to that release. Switching a sortorder means all user indexes need to be reindexed. Please refer to *System Administration Guide* Chapter 9 “Configuring Character Sets, Sort Orders, and Languages” for details on how to change server's default sortorder. If sp_downgrade is called when the new nocase sortorder is in use, the error message “Cannot downgrade to ‘%1!’ server, which does not support server's current default sortorder” will be raised and the downgrade process will be aborted.

Native XML

The XML Service feature of Adaptive Server includes the new xmltable function. If you create views or stored procedures using the xmltable function in 15.0.2, then return to 15.0 or 15.0.1, you will get an error.

Instead of trigger

Instead of Triggers are objects stored in the system catalogs. These objects should be removed before downgrading. When the 15.0 or 15.0.1 server is booted any Instead of Triggers that were not removed will remain in the system catalogs but will not execute.

SQL user defined functions

SQL user defined functions are also objects stored in the system catalogs. If you do not remove them before downgrading, they remain in the catalogs after downgrade. Attempts to drop or execute a SQL user defined function from a 15.0 or 15.0.1 version will result in misleading error messages.

Automatic database expansion

If you installed the 15.0.2 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 15.0 or 15.0.1.

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

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

Alternatively, before downgrading, 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 15.0.2 later.

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
```

Downgrading Job Scheduler

If you are downgrading to Adaptive Server versions 15.0 or 15.0.1, you must run installjsdb script from the lower version anytime you downgrade Job Scheduler.

❖ **Downgrading Job Scheduler**

1 Disable Job Scheduler before you downgrade Adaptive Server:

```
sp_configure "enable job scheduler", 0
```

2 Downgrade Adaptive Server. See the installation guide for details.

3 Run the *installjsdb* script from the version you are downgrading to:

```
isql –Usa –Psa_password –Sservername –i
$SYBASE_ASE/scripts/installjsdb
```

Note The directory with the location of the isql executable (*$SYBASE_OCS/bin*) must be in your path.

4 Enable Job Scheduler:

```
sp_configure "enable job scheduler", 1
```

5 To start Job Scheduler, either restart the server, or enter:

```
use sybmgmtdb
go
sp_js_wakeup "start_js",1
go
```

Returning to Adaptive Server version 15.0.2

If you return to Adaptive Server version 15.0.2, follow the instructions in Chapter 5 of the Adaptive Server Enterprise 15.0.2 Installation Guide.

CHAPTER 9

Downgrading Adaptive Server 12.5.4

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. For the Cluster Edition 12.5.3 ESD #6 is preferred.

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 Direct upgrade to the Cluster Edition is only supported from versions 12.5 to 12.5.3 ESD #6.

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.

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 114 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 118.

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 Users 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 10

Troubleshooting SySAM

This section includes information about troubleshooting the most common SySAM 2.0-related errors.

Calling Sybase Technical Support

Note If SySAM errors are not addressed in a timely fashion, Adaptive Server may cease to function.

There may also be more recent information than published in this guide at SySAM FAQ at http://www.sybase.com/sysam. For the most up-to-date troubleshooting information in this Installation Guide, go to Product Manuals site at http://www.sybase.com/support/manuals/.

If you call SybaseTechnical Support for SySAM-related issues, have this information available:

- The Adaptive Server error log.

  On UNIX:

  $SYBASE/ASE-15_0/install/<servername>.log

  On Windows:

  %SYBASE%\ASE-15_0\install\<servername>.log

- Whether or not Adaptive Server was started grace mode – available from the sp_lmconfig command output.

- If Adaptive Server does not start, the server's properties file:

  On UNIX:

  $SYBASE/ASE-15_0/sysam/<servername>.properties

  On Windows:

  %SYBASE%\ASE-15_0\sysam\<servername>.properties

  servername must correspond to the name of the Adaptive Server that does not start.

- The license files saved in the SYSAM-2_0 *licenses* directory on the machine running Adaptive Server.
- The value of SYBASE_LICENSE_FILE and LM_LICENSE_FILE environment variables.
- Output from lmutil lmpath -status command. *lmutil* is located in the *bin* directory under the *SYSAM-2_0* folder.
- If you are using a served license model:
  - The licenses saved in the *$SYBASE/SYSAM-2_0/licenses* directory of the license server.
  - The license server log file in the *$SYBASE/SYSAM-2_0/log* directory.

Where to look for SySAM-related errors

All SySAM-related errors are printed in the Adaptive Server error log and are prefixed with `"kernel SySAM."` For example:

```
<timestamp> kernel  SySAM: Failed to obtain 1 license(s)
for ASE_CORE feature from license file(s) or server(s).
```

If e-mail alerts are configured, recipients receive an e-mail whenever a SySAM event occurs.

You can always check the current state of SySAM licenses in Adaptive Server using the sp_lmconfig command. The output shows the current SySAM configuration and the status of each license.

If you are using a served license model, the license server's debug log, located in the *log* directory, provides information about any license checkout issues.

Troubleshooting SySAM

Table 10-1 lists methods for troubleshooting SySAM.

Table 10-1: Troubleshooting SySAM

Description	What to do
Cannot find license type during the Adaptive Server installation	The Adaptive Server installation program displays only the most commonly used license types. If you are using a legacy license or a new license type, your license may not be listed by the installation program. If you do not find the license type you need, select "Unknown" or "Other" from the dropdown list and complete the installation. After the installation is complete, use sp_lmconfig to change the license type. See the *Reference Manual:Procedures* for sp_lmconfig usage information. **Note** If you set the license type value to "Unknown" or "Other," Adaptive Server uses the first available license. If you have licenses for more than one type, Adaptive Server may pick the incorrect license. Set the correct license type when you have multiple license types on the license server.

Description	What to do
The installation program produces a warning about not finding a suitable license	If you select a served license model, the install program checks for a license with the selected edition and license type. A warning message is displayed if no suitable license is available. If you are performing a new Adaptive Server installation or upgrading an existing Adaptive Server version 12.5.x or earlier, proceed with the installation. Adaptive Server provides a 30 day grace period to resolve licensing issues. Once the installation is complete, debug the license check-out failure. For more information, see "Adaptive Server cannot check out a license and starts with graced license instead" in this table. If you are upgrading an existing Adaptive Server version 15.0 or later, Adaptive Server may not start after the upgrade if you do not have a suitable license. Exit the installation program and see if the existing Adaptive Server is able to check-out licenses. If it is not, troubleshoot this issue and then proceed with the upgrade. If the existing Adaptive Server is able to check-out licenses, verify the license you are using authorizes use of the update you are applying. This can be determined using the date based version of your license file and the date this update was released. You must have a license with a date based version that is greater than or equal to the date the update was released. Adaptive Server provides a grace period for updating the licenses after support is renewed. The date based version of the license can be determined from the Version column of the output from sp_lmconfig, by looking at the license checkout message in the Adaptive Server error log, or by examining the license itself. For more information, see Appendix A, "Anatomy of a License," of the *Sybase Software Asset Management Users Guide*. The release date of the Adaptive Server can be determined by the date displayed in the version string, cover letter, or SPDC download link. This data is also shown in the error message posted. You can download an updated license from SPDC, if your support was paid for the date the update was released. If you support was not paid, you are not authorized to install an update. See Appendix B of *Sybase Software Asset Management Users Guide* for explanation of how SySAM uses date-based versioning. See "Updating your Sybase licenses" in Chapter 3, "Getting and Using your Licenses," in the *Sybase Software Asset Management Users Guide* for how to update your license.
The license server executables and scripts are not installed	The *sysam* wrapper script and the executables for the license server (lmgrd and *SYBASE*) are not installed or found in the SYSAM-2_0 *bin* directory. The license server component is not selected for installation by default. To install the license server, choose the Custom install option and select license server component. Alternatively, you can download a standalone SySAM license server install program from SPDC.

Description	What to do
The license server does not start	The most common cause for the license server failing to start is that there are no served licenses in the server. In order for license server to start, there needs to be at least one served license in the *licenses* directory. When you install a new license server, there are no served licenses in the licenses directory and the license server does not start until you activate a license from SPDC and copy it into the licenses directory. Other causes for license server not to start include: • The license activated from SPDC was an unserved license. Examine the license files. Served licenses always start with a `SERVER` header. If you do not see a line starting with `SERVER` at the top of the license, you are using an unserved license which does not work with the license server. • If you are using a specific port number for the license, that port number may already be in use. Use `"netstat -a"` to verify that the port number is free. If not, free the port or use a different port number for the license server. • The host name recorded in the `SERVER` header does not match the actual host name. The host name is recorded next to the `SERVER` keyword. If it does not match, correct the host name in the license file or set it to `"this_host"`, which is a keyword that works with any host name. • The licenses copied may be activated for a different machine. Examine the Host ID recorded in the line. It is recorded next to the host name in the `SERVER` header. It must match the Host ID for the machine where license server will be running.
The license server does not recognize the license files	If the license server refuses to serve the licenses, this may be because: • The license was activated for a different machine or activated using incorrect host ID. Examine the host ID recorded in the license with the host ID for the machine. If they are mismatched, check-in and regenerate the license with the correct host ID. • The license has been altered. Nothing in the license except some designated information can be altered. The license is rejected if you change other information. If the license is being created by typing from a paper copy, verify if there are any errors entering the license. You can also download a new copy of the activated license from SPDC. If the Host ID for your platform is based on the Network adapter identification, verify that you have used an ID associated with a valid NIC. IDs associated with loopback adapters are not valid. If the ID associated with a removable network adapter was used, ensure that the adapter is attached to the computer.
Adaptive Server does not start, and generates a license check-out failure error	If Adaptive Server cannot check out the required license, it determines whether the license can be issued in grace mode. If a grace period cannot be given, the license is not granted. If the base license for Adaptive Server (ASE_CORE) was in grace and the issue is not fixed by the end of the grace period, Adaptive Server fails to start. For more information, see “Adaptive Server cannot check out a license and starts with graced license instead” in this table. The troubleshooting tips for resolving licenses issued under grace apply to solve this issue as well.

Description	What to do

Adaptive Server cannot check out a license and starts with graced license instead	Execute the sysam diag *feature_name* command from the SYSAM-2_0 *bin* directory, where *feature_name* is the SySAM feature name for Adaptive Server or the optional feature that is in grace mode. The feature name is printed in the Adaptive Server error log and in the optional e-mail notifications.
	If the diag command shows no licenses available for check-out, this may be due to one of the following reasons:
	• If you are using a **served** model:
	  • Verify that the license server is running and reachable from the machine you are using. Use `"sysam status"` to verify that you can access the license server. If you cannot access the license server, see if the license server is running.
	  • If the license server is running, use `"sysam status -f feature_name"` to determine whether the license server is serving a license for the given feature. If not, obtain the correct licenses from SPDC.
	  • All licenses on the license server may be in use. If `"sysam status -f feature_name"` indicates that no licenses are available, obtain additional licenses or shutdown existing instances of Adaptive Server.
	• If you are using an **unserved** model:
	  • The license for the given feature may not be copied in the local *licenses* directory. Obtain the correct licenses from SPDC and copy them to the local *licenses* directory.
	  • The license may have been activated for a different machine or activated with incorrect host ID. Check-in and reactivate the license with the correct host ID.
	• The available licenses are for a different operating system or architecture. Obtain license for the correct platform from SPDC.
	• If you are running in a terminal server environment, unserved licenses cannot be used. Set up served licenses for a terminal server environment.
	• The available license may not authorize use of this version of Adaptive Server. For information about date based versioning, see Appendix B, “Installing Product Updates, EBFs, and Support Renewal” in the *Sybase Software Asset Management Users Guide*. For information about how to update your license, see the “Updating your Sybase licenses” section in Chapter 3, “Getting and Using your License,” in the *Sybase Software Asset Management Users Guide*.

Description	What to do
Adaptive Server cannot checkout a license and starts with graced license instead (continued)	If the diag command shows the license for the given feature is available for check-out, the reason that Adaptive Server cannot check-out the license could be: • The product edition or license type does not match the requirement. If the edition and license type configuration parameters are set, Adaptive Server uses only the license that matches these settings. 1 Run sp_lmconfig to see the configured values. 2 Check the VENDOR_STRING attribute of the licenses to make sure a matching license exists. If a matching license does not exist, obtain a suitable license from SPDC or adjust the Adaptive Server configuration. If Adaptive Server cannot start because the grace period expired, examine the *servername.properties* file located in the *ASE-15_0/sysam* directory under your *SYBASE* directory for values. These values are stored in two lines that start with PE= and LT=. These lines are present only if the configuration is set. • Also see "Adaptive Server cannot find license for optional feature, even though the license exists" in this table for more information.
Adaptive Server shows graced licenses even after the issue has been fixed	Adaptive Server performs license checks periodically, and the license status is not immediately updated. It is updated only after the next heartbeat cycle is completed. This may take a few hours.
Adaptive Server cannot find license for an optional feature, even though the license exists	Execute "sysam diag *feature_name*" to ensure that license for the optional feature exists and can be checked out from the machine where Adaptive Server is running. If the feature exists, but cannot be checked out from Adaptive Server, this could be because: • The optional feature is for a different edition. • The Active or Standby flags do not match for base Adaptive Server and the optional feature. Execute sp_lmconfig to determine the edition and active and standby configuration value. The edition and active and standby values are displayed in the Property Name / Property Value table. The edition value is displayed in the PE row and active and standby value is displayed in the AS row. Compare the values of PE and AS for the Adaptive Server with the same values recorded in the VENDOR_STRING and ISSUER fields of the optional feature license. They must match. Some features like ASE_HA are not offered with standby pricing, so there is no "AS=A|S" flag in the license for this feature. This field is not considered when searching for a license without active and standby capability.

Description	What to do
Adaptive Server does not start with the expected edition or license type	If the edition and license type configuration parameters are not set, Adaptive Server uses the first available `ASE_CORE` license. If you have multiple `ASE_CORE` licenses with different editions and license types, which license is the first available depends on many factors such as license file directory sort order, available licenses on the license server, and so on. Because what edition and license type is granted cannot be guaranteed, Sybase recommends that you set the edition and license type configuration parameters. When these parameters are set, you are assured that Adaptive Server starts with that configuration. If a matching license were to be unavailable, Adaptive server starts under grace (if available) to allow you to resolve the licensing issue.
Adaptive Server does not pick the correct license	When Adaptive Server looks for a license, it does so in the following locations: • Any saved values in the *registry* on Windows platforms, or the *.flexlmrc* file on UNIX platforms. Use `"lmpath -status"` to view the saved values. • The values set for SYBASE_LICENSE_FILE and LM_LICENSE_FILE environment variables. • All license files (any file with a *.lic* extension) in the licenses directory under the *SYSAM-2_0* directory. All of the above locations are searched in the order specified until a suitable license is found. When a directory is specified, license files in that directory are loaded in the directory sort order. The first license that matches the feature name, version, edition, and license type filter is used. This license may not be what you were expecting to use. Adjust the above locations to control this.

SySAM grace mode

If Adaptive Server cannot find a required license, it continues to operate in grace mode, if a grace period is available. Determine the license status of Adaptive Server using the output from sp_lmconfig. The `Status` column shows `graced` for any license that is operating within a grace period.

The Adaptive Server error log includes information about licenses awarded under grace mode. If e-mail notifications are set up for a severity of “warning,” e-mail messages are sent for this event. See “Enabling and changing e-mail notifications” on page 131 for more information.

The warnings in the Adaptive Server error log and e-mail messages are repeated while the license is under grace.

SySAM grace period

Typically, you have a 30-day grace period to resolve any SySAM-related issues. When a license is awarded under grace, the Adaptive Server error log entry and optional e-mail message include the date and time when the grace period expires. In addition, you can execute sp_lmconfig to determine the expiration date. Adaptive Server disables the functionality corresponding to the license if the issue is not fixed by the expiration date. Adaptive Server shuts down an expired license is ASE_CORE.

Enabling and changing e-mail notifications

Use sp_lmconfig to configure e-mail notification. To view the current notification status and configuration, execute sp_lmconfig without any parameters.

To enable e-mail notifications, set the smtp host, smtp port, email sender, email recipients, and email severity properties.

To change the severity that triggers e-mails, set the email severity property to: `"ERROR"`, `"WARNING"`, `"INFORMATIONAL"` or `"NONE"`.

To change the list of e-mail recipients, set the email recipients property to a comma-separated list of e-mail addresses. E-mail addresses can include anything that is capable of receiving SMTP messages.

CHAPTER 11

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*.

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 134.

Table 11-1 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 srvbuild unexpectedly quits, or if you cannot correct the problem, see the *Error Messages and Troubleshooting Guide*.

Table 11-1: Troubleshooting guidelines for UNIX platforms

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. **Note** Adaptive Server may not start if it cannot find a SySAM license and if the grace period has expired. Check the Adaptive Server error log for the cause of the license checkout failure and fix the issues.
The installation program detects reserved word conflicts.	See "Stopping Adaptive Server after a failure" on page 138.
The upgrade fails.	See "Recovering from a failed upgrade" on page 140.

Error log locations

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

Logs for installation utilities

Table 11-2 lists the default error log locations for the installation, upgrade, and setup utilities.

Table 11-2: Error log locations for installation utilities

Utility	Error log location
InstallShield	*$SYBASE/log.txt*
srvbuildres	*$SYBASE/$SYBASE_ASE/init/logs/srvbuildMMDD.NNN* where: • *MM* is the month. • *DD* is the date. • *NNN* is a three-digit number identifying the srvbuild session.

Utility	Error log location
sqlupgraderes	• *$SYBASE/$SYBASE_ASE/init/logs/sqlupgradeMMDD.NNN* • *$SYBASE/$SYBASE_ASE/upgrade/upgrade.NNN* (a temporary file created by the upgrade process)

Logs for Sybase servers

Table 11-3 lists the default error log locations for each Sybase server.

Table 11-3: Error log locations for Sybase servers for UNIX

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*.

Cannot 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-15_0
% 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 the installer 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, where *host_name* is the name of the machine on which you want the installer 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 the installer 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:

1 Enter the following command at the UNIX prompt of your *local* machine, where *remote_machine* is the machine on which you are running the installer:

```
xhost +remote_machine
```

2 Restart the installer.

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 the Software” for instructions on adjusting the shared memory value. Restart the installation or upgrade process.

The Installer fails to start

If the installer does not start, restart it using:

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

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 the Adaptive Server 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. Verify that the XP server exists in the interfaces file or LDAP server.

Use sp_addserver to add an entry to the sysservers table.

Configuration failed

Sun Solaris

Check the location of the system shared libraries.

The shared libraries *libXt.so* and *libX11.so* are normally stored in */usr/openwin/lib*. The shared library *libsocket.so* is normally located in */usr/lib*.

If the shared libraries on your operating system are located in directories other than those mentioned in the preceding paragraph, set the environment variable LD_LIBRARY_PATH to indicate the new locations.

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 cluster 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 11-3 on page 135.

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 11-3 on page 135.

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.

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

```
% setenv SYBSHUTWAIT 120
```

If Adaptive Server fails the pre-upgrade eligibility test

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 failed the pre-upgrade eligibility test.

After you resolve any problems, shut down Adaptive Server and use sybcluster or manual upgrade 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 11-2 on page 134.

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

Failures while upgrading a server installation fall into one of two categories: failure to upgrade an individual database, or failure to complete configuration changes after all databases have been upgraded.

1 If an individual database fails to upgrade, you can retry that upgrade manually. Begin by correcting the problem that caused the failure. Output from the upgrade utility should identify the problem. The most common cause of failure is running out of some resource: space (either data or log), locks, auxiliary scan descriptors. You can add space to the database using the alter database command. Other resource failures may be correctable by changing the server's configuration via the sp_configure stored procedure.

   If an upgrade failure leaves a database offline, and the failure can only be corrected by making data changes in the database, you can gain access to the failed database by using isql or a similar program to connect to the affected server as user "sa" and issuing this command:

```
dbcc traceon(990)
```

Note This trace flag grants access only to user “sa”. It is not sufficient to use an account having “sa_role”. If you have disabled the “sa” login, you must re-enable it to get access using this method.

Having set this trace flag, user “sa” can now use the offline database and make the necessary changes to correct the upgrade failure.

To restart a failed upgrade, use the online database command:

```
online database <failed_db_name>
```

The server restarts that database’s upgrade from the point of failure.

2 If the failure occurs after all databases have been upgraded, or if a failure somehow causes the upgrade utility to crash, you can re-run the utility manually. First diagnose and correct the failure, then run the upgrade utility:

```
$SYBASE/$SYBASE_ASE/upgrade/upgrad
```

When restarted in this way, the upgrade process says it is “verifying” the upgrade rather than “starting” it, but it makes all the same checks as for the original upgrade.

To verify that a database has upgraded successfully, you can check any database’s upgrade status using the online database command. If any upgrade is required for a database, this command performs it. You may also use a procedure such as this to check all databases at a given installation:

```
declare @dbname varchar(255)
select @dbname = min(name)
from master..sysdatabases
while @dbname is not null
begin
online database @dbname
select @dbname = min(name)
from master..sysdatabases
where name > @dbname
```

```
end
```

Note There are certain upgrade failures from which the server cannot recover. For example, attempts to upgrade system tables to version 15.0 format are quite sensitive to failures at certain points during the required changes. If you encounter such a failure, restore the failed database from backup. To prevent the upgrade from failing again, correct the problem that caused the original failure **before** issuing the online database command for that database. These catastrophic failures are nearly always caused by running out of resources, as described above, which then causes a failure to undo the aborted transaction.

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 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.

2 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.

3 Run sybluster or restart the manual upgrade.

4 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*.

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 15.0. 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 144.

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 147.

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 15.0 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*.

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' ] ] ) ]
```

Parameters

- *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

Example 1 Upgrades all compiled objects in the active database:

```
dbcc upgrade_object
```

Example 2 Upgrades all stored procedures in the listdb database. Single quotes are used around procedure because set quoted identifiers is on:

```
dbcc upgrade_object(listdb, 'procedure')
```

Example 3 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, "rule")
```

Example 4 Upgrades all stored procedures named list_proc in the listdb database.

```
dbcc upgrade_object(listdb, list_proc)
```

Example 5 Upgrades the stored procedure list_proc, which is owned by the login “jkarrik”.

```
dbcc upgrade_object(listdb,
  "listdb.jkarrik.list_proc")
```

Example 6 Returns an error because the value of *dbname* is master and the value of *database* is listdb. These values must match.

```
dbcc upgrade_object(master,
  "listdb.jkarrik.list_proc")
```

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 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 contains 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.

CHAPTER 12

Removing Adaptive Server

Only actions performed by the Installer are removed during an uninstallation process. This means that files or registry entries created after installation are not removed and must be manually removed by the customer when the installation process has finished.

Uninstalling the Cluster Edition

Run:

```
$SYBASE/uninstall/ASESuite/uninstall
```

Follow the instructions to uninstall some or all of the components.

To uninstall PC-Client from a Windows machine, run:

```
%SYBASE%\uninstall\PCClient\uninstall.exe
```

You can also remove PC-Client using the Add/Remove Programs from the Control Panel.

Removing an existing Adaptive Server

1 To remove an existing Adaptive Server, from *$SYBASE*, 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.

APPENDIX A

Alternative Installation Methods

Installation on consoles (non-GUI mode)

You can launch the installer in console or non-GUI mode.

1 Change the default directory to the CD.

2 Run the setup program with the -console command line argument:

```
./setup -is:javaconsole -console
```

3 In cases where the installer launches automatically, select Cancel to stop the GUI installation, then launch the setup program from a terminal or 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

Run the installer to perform a silent installation . You need to provide a response file that contains answers to all of the installer's questions. There are two methods of generating a response file for the installer:

The first is to run the installer in GUI mode and record all of the responses into a response file. Then edit the file to customize the responses if needed.

1 Change the default directory to the CD.

2 Run the setup program:

```
./setup -is:javaconsole -options-record <full_path response file>
```

3 Run through the installation, saving all of the responses.

4 Edit the response file to customize it for your needs.

Run the setup program with the -options *filename* and -silent command line arguments. You must also include the -W argument, indicating that you accept the license agreement.

```
./setup -is:javaconsole -silent -options <full path response file>
-W SybaseLicense.agreeToLicense=true
```

The second method is to run the installer and have it write out a template file. The template file contains all of the questions, but no responses. You can then edit the file to provide responses for each of the questions. The installer is then invoked from the response file, and all questions are answered by reading from the response file.

To generate a template response file, skip step 3 and alter step 2 above to do the following:

```
setup -is:javaconsole -options-template <full path response file>
```

Except for the absence of the GUI screens, all actions of the installer 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.

Installation in silent mode may fail if a response is required from the user. Such a response is required when files already exist on disk and appear to be newer than the files the installer is attempting to install, or when files have been modified by the user.

Set the following options at the command line to specify actions to take when installing in silent mode:

- replaceExistingResponse – stores your response to whether you want to replace a file that currently exists on your system with the one being installed.

- replaceNewerResponse – stores your response to whether you want to replace a file that currently exists on your system with the one being installed if the existing file is newer than the file being installed.
- removeExistingResponse – stores your response to whether you want to remove a file that currently exists on your system.
- removeModifiedResponse – stores your response to whether you want to remove a file that has been modified since installation.

You can specify any of these options on the command line or within the options file as follows:

```
-G <option>="<value>"
```

Valid values are:

- yesToAll
- yes
- noToAll
- no

For example:

```
./setup -is:javaconsole -silent -options <responseFileName>
   -W SybaseLicense.agreeToLicense=true
   -G replaceNewerResponse="yesToAll"
```

APPENDIX B

Manually Setting Up and Managing the Cluster

This appendix describes how to manually configure and manage a shared-disk cluster.

Note If you set up the cluster manually, you cannot use the Adaptive Server plug-in or sybcluster to manage your cluster until you create and deploy the Unified Agent. See "Enabling sybcluster and Adaptive Server plug-in after a manual configuration" on page 172.

Before you begin

Before you configure the cluster,

- Set the Sybase environment
- Create the raw devices

Setting the environment

Set your Sybase environment. From the Sybase release directory, source the *SYBASE.sh* or *SYBASE.csh* file:

```
. ./SYBASE.sh
```

or,

```
SYBASE.csh
```

Creating the raw devices

Each raw device must be accessible from each node using the same path. See your operating system storage administrator or documentation for help in setting up raw devices.

The local system temporary database device and the quorum disk device are new for Cluster Edition. The other required devices are the same as those required for all Adaptive Servers.

- Master database device
- sybstemprocs database device
- System database device
- Local system temporary database device (you can create multiple devices to setup one local system temporary database per device)
- Quorum disk device (at least 20 MB)
- All other database devices

Configuring the cluster

To configure the cluster manually,

1 Create the cluster input file.

2 Configure the interfaces file.

3 Build the master and quorum devices.

4 Create a runserver file for each instance.

5 Initialize a device for sybsystemprocs and create the sybsystemprocs database.

6 Install the system stored procedures.

7 Set up a local system temporary database for each instance.

8 Start the cluster.

Note Sybase strongly recommends that the total number of engines in all instances does not exceed the number of CPUs.

Creating the cluster input file

Before configuring the cluster, create a cluster input file that specifies the name of the cluster, the number of instances in the cluster, the path to the directories containing the interfaces file, log files, and quorum disk devices, and other required configuration information. You can choose any name for the cluster configuration file (for example, *mycluster.inp*).

When you configure the cluster, Adaptive Server reads the information from the cluster input file and stores it securely in the quorum device. Adaptive Server subsequently retrieves the cluster configuration information from the quorum device.

See “Reconfiguring the cluster” on page 171 for information about changing configuration information after the cluster has been initialized.

Note You can configure one cluster with each cluster input file.

The cluster input file is distinct from the server configuration file, which stores Adaptive Server configuration values associated with sp_configure.

This is the syntax for the cluster input file:

```
# all input files must begin with a comment

[cluster]
name = cluster name
max instances = number
master device = path to the master device
primary protocol = udp | tcp | other
secondary protocol = udp | tcp | other
config file = Adaptive Server configuration file name
interfaces path = interfaces file path
```

traceflags = *trace flag number*, *trace flag number*, . . .
additional run parameters = *any additional run parameters*

[management nodes]
hostname = *node_name*
hostname = *node_name*
hostname = *node_name*
hostname = *node_name*

[instance]
id = *instance ID*
name = instance name
node = *name of node on which this instance runs*
primary address = *primary interconnect address*
primary port start = *port number*
secondary address = *secondary interconnect address*
secondary port start = *port number*
errorlog = *file name*
config file = *Adaptive Server configuration file name*
interfaces path = *interfaces file path*
traceflags = *trace flag number*, *trace flag number*, . . .
additional run parameters = *any additional run parameters*

[instance]
id = *instance ID*
name = *instance name*
node = *name of node on which this instance runs*
primary address = *primary interconnect address*
primary port start = *port number*
secondary address = *secondary interconnect address*
secondary port start = *port number*
errorlog = *file name*
config file = *Adaptive Server configuration file name*
interfaces path = *interfaces file path*
traceflags = *trace flag number*, *trace flag number*, . . .
additional run parameters = *any additional run parameters*

Where:

- name – name of the cluster.
- max instances – The maximum number of instances in the cluster that your version of Adaptive Server supports. See the Release Bulletin for the maximum number of instances your version of Adaptive Server supports.
- master device – path to the master device.
- primary protocol – protocol used for the primary interconnect.
- secondary protocol – protocol used for the secondary interconnect.

- config file – path to the Adaptive Server configuration file. This file is used by all instances in the cluster unless an individual instance overrides this setting.
- interfaces path – path to the interfaces file. Omit this parameter if you are using an LDAP environments. This interfaces file is used by all instances unless an individual instance overrides this setting.
- traceflags – comma-separated list of traceflags used at instance startup.
- additional run parameters – additional parameters passed to instances during startup.
- hostname – name of the node. This name should be the same as returned by the hostname command when run on this node. There is one hostname field for each node that must be registered. That node only needs to be specified once in the management node section.
- name – name of the instance.
- ID – ID of the instance.
- node – name of the node on which this instance runs.
- primary address – address of this instance on the primary interconnect.
- primary port start – starting port number on the primary interconnect.
- secondary address – address of this instance on the secondary interconnect. Required if a secondary is defined in the interconnect section or if a secondary protocol is specified. Ignored if a secondary is not defined.
- secondary port start – starting port number on the secondary interconnect. Required if secondary address or a secondary protocol is specified.
- error log – full path to the error log for this instance.
- config file – path to an Adaptive Server configuration file. This file overrides the configuration file field specified in the cluster section of the cluster input file.
- interfaces path – path to the server-side *interfaces* file. This file overrides the interfaces file field in the cluster section of the cluster input file. Do not include the *interfaces* file name in the path.
- traceflags – comma-separated list of traceflags used at instance startup. These are used in addition to, not in place of, the trace flags specified in the cluster section of the cluster input file.

- `additional run parameters` – additional parameters passed to instances during startup.

  The formula for finding the socket port range is:

  start_port_number + (*max_instances* * 5) – 1

Note If you do not use an ADO.NET, ensure that the port numbers you choose are not in use by any other process.

In this example, the cluster input file defines a cluster named “mycluster” with two instances, “ase1” on node “blade1” and “ase2” on node “blade2.” The addresses on the private interconnects are 192.169.0.1 and 192.169.0.2. The name of the server configuration file is *mycluster.cfg*. The maximum instances is 2. “ase1” has a starting port range of 15015, and “ase2” has a starting port range of 16015.This adds additional information to the mycluster cluster:

```
#input for a 2 node / 2 instance cluster
[cluster]
name = mycluster
max instances = 2
master device = /opt/sybase/rawdevices/mycluster.master
config file = /opt/sybase/ASE-15_0/mycluster.config
interfaces path = /opt/sybase
primary protocol = udp
seconday protocol = udp

[management nodes]
hostname = blade1.sybase.com
hostname = blade2.sybase.com


[instance]
id = 1
name = ase1
node = blade1.sybase.com
primary address = 192.169.0.1
primary port start = 15015
secodary address = 192.169.1.1
secondary port start = 15015
errorlog = /opt/sybase/ASE-15_0/install/ase1.log
additional run parameter = -M/opt/sybase/ASE-15_0

[instance]
id = 2
name = ase2
```

```
node = blade2.sybase.com
primary address = 192.169.0.2
primary port start = 16015
secondary address = 192.169.1.2
secondary port start = 16015
errorlog = /opt/sybase/ASE-15_0/install/ase2.log
additional run parameter = -M/opt/sybase/ASE-15_0
```

For an example of cluster input file where all instances are located on a single node, see the *Users Guide*.

Configuring the interfaces file

The interfaces file for clusters must include entries for the cluster server and all instances.

The syntax for the interfaces file is:

instance_name
  query *network_protocol* ether *machine_name* *port_number*
. . .
cluster_server_name
  query *network_protocol* ether *machine_name* *port_number*

Where:

- *instance_name* – instance for which you are making this interfaces file entry.
- *network_protocol* – network protocol used for the instance.
- *machine_name* – name of the machine on which the instance runs.
- *port_number* – port number used to connect to this instance.
- *cluster_server_name* – name of the cluster server.

This example includes the instances “ase1,” “ase2,” and “ase3,” on machines “blade1,” “blade2,” and “blade3,” running in the cluster “mycluster”:

```
ase1
   master tcp ether blade1 19786
   query tcp ether blade1 19786
ase2
   master tcp ether blade2 19786
   query tcp ether blade2 19786
ase3
   master tcp ether blade3 19786
   query tcp ether blade3 19786
```

```
mycluster
   query tcp ether blade1 19786
   query tcp ether blade2 19786
   query tcp ether blade3 19786
```

Building the master and quorum devices

Create the quorum and master devices on raw devices on the shared-disk storage.

When creating the quorum device, use the same device name and major and minor numbers on all machines. Each raw device must be accessible from each node using the same path. The example described in "Creating the cluster input file" on page 159 specifies raw device *raw11* as the quorum disk:

```
dataserver
...
--quorum_dev /dev/raw/raw11
```

The syntax for creating Cluster Edition master and quorum devices is:

```
dataserver
--cluster_input= cluster_input_filename_and_path
--quorum_dev= quorum_device_and_path
--master_device_size= master_device_size
--logical_page_size= page_size
--instance= instance_name
--buildquorum
```

Where:

- --master_device_size=<*size spec*> – specifies the size of the master device.
- --cluster_input=<*cluster input file*> – loads the cluster configuration specified in the input file onto the quorum device.
- --quorum_dev= *path_to_quorum_device* – specifies the full path to the quorum device.
- --instance= *instance_name* – specifies the name of an instance.
- --logical_page_size= – specifies the page size.

- --buildquorum – specifies that a new quorum device should be built.

Note See dataserver under "Changes to commands" section in the "System Changes" chapter of the *User Guide* for more information about the dataserver parameters.

Create the master and quorum devices using dataserver. This example creates an instance named "ase1" that is configured for 4K pages, a master device of 500MB, and a quorum device,

```
/opt/sybase/ASE-15_0/bin/ dataserver\
--quorum_dev=/dev/raw/raw11\
--instance=ase1
--cluster_input=/opt/sybase/mycluster.inp
--master_device_size=500M
--logical_page_size=4k
--buildquorum
```

Creating the runserver files

After creating the master and quorum devices, create a runserver file for each instance in the cluster. Later, you will use these files to start the instances.

❖ **Creating runserver files for each instance**

1 Create a runserver file.

This example creates *RUN_ase1* for instance ase1. Enter the runserver file on a single line. The "\" indicates that the line continues without a space.

```
$SYBASE/ASE-15_0/bin/dataserver
--quorum_dev=/dev/raw/raw11
--instance=ase1
```

Note Create all devices, database devices, and quorum devices as raw devices on shared disks.

2 Make a copy of the runserver file for each of the Adaptive Servers in the cluster. For example, the three instances for the cluster named "mycluster" have runserver files named *RUN_ase1*, *RUN_ase2*, and *RUN_ase3*. Make sure you include the same quorum device for all instances. --instance should change in each file to point to the appropriate instance name.

Initializing and creating *sybsystemprocs*

Use disk init to initialize a device for sybsystemprocs and then create the sybsystemprocs database.

1 Initialize a raw device of at least 150MB for sybsystemprocs:

```
disk init name = "sysprocsdev",
physname = "/dev/raw/raw13",
size = "150"
```

2 Create the sybsystemprocs database:

```
create database sybsystemprocs on sysprocsdev = 150
```

Installing the system stored procedures

Run installmaster from the *$SYBASE/$SYBASE_ASE/scripts* directory to install the system stored procedures. You can run installmaster from any instance.

```
isql -U sa -P sa_password -S server_name -n
-i $SYBASE/$SYBASE_ASE/scripts/installmaster
-o output_file
```

Setting up local system temporary databases

In a shared-disk cluster, each instance must have a local system temporary database.

❖ **Creating a local system temporary database**

In the Cluster Edition, if an instance does not have a local system temporary database, it will boot only if it is the first instance to boot in the cluster.

1 Start Adaptive Server. For example:

```
startserver -f $SYBASE/$SYBASE_ASE/install/RUN_ase1_coord
```

2 Log in to Adaptive Server.

3 If you are not using the master device for the temporary databases, create the devices for the local system temporary databases. Sybase recommends separate devices for log and data for these databases. For example:

```
disk init name="tempdbdev1",
physname="/dev/raw/raw14",
size="400M"
```

And for the log device:

```
disk init name="temp_log_dev1",
physname="/dev/raw/raw15",
size="200M"
```

4 Create a local system temporary database for each instance in the cluster. For example, to create three local system temporary databases "ase1_tdb1," "ase2_tdb1," and "ase3_tdb1" for instances "ase1," "ase2," and "ase3," respectively, enter:

```
create system temporary database ase1_tdb1 for instance ase1
on tempdbdev1 = 100
log on temp_log_dev1 = 50
create system temporary database ase2_tdb1 for instance ase2
on tempdbdev1 = 100
log on temp_log_dev2 = 50
create system temporary database ase3_tdb1 for instance ase3
on tempdbdev1 = 100
log on temp_log_dev3 = 50
```

5 Shut down the cluster using the shutdown cluster command.

Dropping Local System Temporary Databases

Because a local system temporary database is always in use if its instance is running, you cannot drop the local system temporary database while the instance is running. To drop a local system temporary database, shutdown the instance and drop the local system temporary database from another instance.

To drop the local system temporary database of the last instance:

- Use sp_tempdb_markdrop to first mark the database to be dropped.
- Shutdown this instance.
- Drop the temporary database from another instance when you restart the cluster.

For example, if mycluster_tbdb_1 is the last local system temporary database, mark it with this command:

```
sp_tempdb_markdrop mycluster_tdb_1
```

Configuring the cluster for automatic cluster takeover

You can configure the cluster to automatically recover from an abrupt total cluster failure. The automatic cluster takeover is triggered if an instance attempts to connect to the cluster and:

- You have enabled automatic cluster takeover.
- The quorum device indicates the cluster is running, but the instance detects no heartbeat.

The automatic cluster takeover configuration parameter enables the instance attempting to connect to the cluster to form a new cluster, starting the cluster coordinator and recovering the databases. The syntax for automatic cluster takeover is:

```
sp_configure "automatic cluster takeover", [1 | 0]
```

Setting automatic cluster takeover to 1 enables the automatic cluster takeover (the default). Setting it to 0 disables automatic cluster takeover.

See “Starting the cluster” on page 168 for information about manually restarting the cluster.

In environments that have i/o fencing enabled, automatic cluster takeover is guaranteed to be a safe operation. In environments that don't have i/o fencing, a malfunction of the algorithm could introduce data corruption. Therefore this configuration parameter is present to disable the algorithm. However there is always a risk of data corruption in environments without i/o fencing, and disabling automatic cluster takeover does not mitigate all of those risks.

Starting the cluster

Start the cluster by starting all the instances associated with the cluster on each node in the cluster.

You can start a cluster:

- Normally, after a graceful shutdown. See “Starting the cluster after a normal shutdown” on page 169.

- After a system failure. See "Starting the cluster after a system failure" on page 169.

Note Sybase recommends that you do not alter the runserver files used for normal cluster startup.

❖ **Starting the cluster after a normal shutdown**

1 Start each cluster instance from the node on which it runs:

For example, start the "ase1" instance:

```
startserver -f $SYBASE/$SYBASE_ASE/install/RUN_ase1
```

2 Log in to each node on which another instance will run, and execute startserver.

For example, to start the "ase2" instance on "blade2":

1 On "blade2," move to the *$SYBASE* directory.

2 Issue the startserver command:

```
startserver -f $SYBASE/$SYBASE_ASE/install/RUN_ase2
```

❖ **Starting the cluster after a system failure**

If the cluster goes down because of a system failure and you have enabled the automatic cluster takeover configuration parameter follow step 1. If you have not enabled the automatic cluster takeover configuration parameter follow step 2.

1 Set it to the default, which is 1. The instance attempting to connect to the cluster restarts the cluster, acting as the cluster coordinator and recovering any databases. See "Configuring the cluster for automatic cluster takeover" on page 168 for information about configuring the cluster to automatically restart.

2 If you do not enabled automatic cluster takeover, you must reboot the cluster with the dataserver . . . --cluster_takeover parameter. For example, this example adds the -cluster_takeover parameter to the runserver file that originally started the cluster:

```
$SYBASE/ASE-15_0/bin/dataserver \
--quorum_dev=/dev/raw/raw11\
--instance=ase1\
--cluster_takeover
```

3 Once the cluster is running, restart all the instances as you regularly would.

Post-configuration tasks

To determine whether the instances you configured are running, log in to the cluster and enter:

```
sp_cluster show
```

❖ **Verifying that you can connect to the instances**

1 Source *SYBASE.sh*.

2 Use isql to connect to servers. At the command prompt, enter:

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

where *server_name* is the instance name.

The command prompt displays if the login is successful.

3 To display the Adaptive Server version number, enter:

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

If you encounter errors, see the *Error Messages and Troubleshooting Guide*.

Shutting down the cluster or the instance

❖ **Shutting down the cluster**

Shutting down the cluster shuts down all the instances associated with the cluster.

1 Log in to any instance. For example:

```
isql  –Usa  –P –Sase2 –I$SYBASE/interfaces
```

2 Issue the shutdown command:

```
shutdown cluster
```

Shutting down an instance

Use the shutdown command to shut down an individual instance in the cluster. You can shut down an instance from another instance in the cluster.

❖ **Shutting down an instance**

1 Log in to any instance. For example:

```
isql  -Usa  -P -Smycluster -I$SYBASE/interfaces
```

2 Issue the shutdown command:

```
shutdown ase2
```

Reconfiguring the cluster

You can reconfigure a cluster to add instances, change trace flags, and so on. Edit the cluster input file and then create a new run file for the instance you intend to boot first that includes dataserver with the cluster_input option. This option tells Adaptive Server to write the new configuration information to the quorum device.

❖ **Reconfiguring the cluster**

Log in to any instance, and shut down the cluster.

Note If you use sybcluster or the Adaptive Server plug-in, you do not need to shut down the cluster before reconfiguring it.

---

1 Extract the current configuration to a file using the -extract_config command to the qrmutil utility. See the *Users Guide* for more details:

2 Edit the cluster input file.

3 Copy the run file for the instance you intend to boot and add the --cluster_input option to the dataserver statement. For example, change *RUN_ase1* in this way:

```
$SYBASE/$SYBASE_ASE/bin/dataserver\
--cluster_input=/<input_file>\
--quorum_dev=/dev/raw/raw11\
--instance=ase1\
```

4 Start the cluster.

Enabling *sybcluster* and Adaptive Server plug-in after a manual configuration

To use sybcluster or the Adaptive Server plug-in to manage the cluster after a manual configuration, you must start a Unified Agent on each node used by the cluster, and then deploy the connection agent to each node.

Follow these steps:

1 Start a Unified Agent on each node used by the cluster. From the *$SYBASE* directory, enter:

```
UAF-2_5/bin/uafstartup.sh
```

2 Start sybcluster. For example, enter:

```
sybcluster -U uafadmin -P -C mycluster -F "blade1,
blade2,blade2"
```

3 Execute deploy plugin. For example:

```
deploy plugin agent "blade1,blade2,blade3"
```

See the chapter titled The "sybcluster Utility" in the *Users Guide* for complete syntax and usage information for sybcluster and ASE plug-in.

APPENDIX C

A *sybcluster* Sample Session

This appendix provides an example of a sybcluster session and instructions for configuring a cluster using a sybcluster input file.

Configuration values

This appendix provides an example of how to configure a typical shared-disk cluster using sybcluster. See Table C-1for the configuration values used in the example session.

Note This example assumes a primary and secondary private network. If you do not specify a primary private network, sybcluster asks for a beginning port number and calculates the number of port numbers needed for communication among the instances. Ensure that the number you provide, and the following required port numbers, are not in use by another application. For information on about how sybcluster calculates the number of required ports, see "Creating the cluster using sybcluster" on page 44.

Table C-1: sybcluster prompts and configuration values entered for sample session

Parameter	Value
Cluster name	mycluster
Number of instances	
Maximum number of nodes	
Cluster node ID	1
-----Quorum Device-----	

Parameter	Value
Full path and name for the quorum device	/dev/raw/raw11
-----Master device-----	
Full path and name for the master device	/dev/raw/raw12
Size of master device [80MB]	80MB
Size of master database [60MB]	60MB
Instance page size in kilobytes [4]	2KB
-----Sybase system procedure device-----	
Full path and name for sybsystemprocs device	/dev/raw/raw13
Size of sybsystemprocs device (MB)	150MB
Size of sybsystemprocs database (135)	135MB
-----System database device-----	
Full path and name of system database device	/dev/raw/raw14
Size of system database device (MB)	12MB
Size of system database (MB)	12MB

Does this cluster have a secondary network [Y]	Yes
-----Verify file locations-----	
The location of the SYBASE home directory	/remote/var/sybase/
Full path to environment variable script	/remote/var/sybase/SYBASE.sh
Path to *$SYBASE_ASE*	/remote/var/sybase/ASE-15_0
Path to interfaces file directory	/remote/var/sybase
Path to the dataserver configuration file	/remote/var/sybase/mycluster.cfg
-----Instance information-----	
Node name	blade1
sybcluster displays: `Cluster: mycluster - Node: blade1 - Agent blade1:9999`	
Instance name	ase1
Query port number for ase1	19786
Primary protocol address for ase1	000.000.001.001
Secondary protocol address for ase1	000.000.001.002
-----Local system temporary database-----	
Local system temporary database device name (Enter the name of the Adaptive Server database device for the local system temporary database.)	mycluster1_tempdb
Path to local system temporary database device	/dev/raw/raw15
Local system temporary database device size (MB)	40MB
Local system temporary database name	mycluster_tdb_1
Local system temporary database size (MB)	40MB

Parameter	Value

Do you want to add another instance (Y or N)	Yes
Node name	blade 2
sybcluster displays: `Cluster: mycluster - Node: blade2 - Agent blade2:9999`	
Instance name	ase2
Query port number for ase2	19786
Primary protocol address for ase2	000.000.002.001
Secondary protocol address for ase2	000.000.002.002
-----Local system temporary database-----	
Local system temporary database device name (Enter the name of the Adaptive Server database device for the local system temporary database.)	mycluster2_tempdb
Path to local system temporary database device	/dev/raw/raw16
Local system temporary database device size (MB)	40MB
Local system temporary database name	mycluster_tdb_2
Local system temporary database size (MB)	40MB

Do you want to add another instance (Y or N)	Yes
Node name	blade3
sybcluster displays: `Cluster: mycluster - Node: blade3 - Agent blade3:9999`	
Instance name	ase3
Query port number	19786
Primary protocol address for ase3	000.000.003.001
Secondary protocol address for ase3	000.000.003.002
-----Local system temporary database-----	
Local system temporary database device name (Enter the name of the Adaptive Server database device for the local system temporary database.)	mycluster3_tempdb
Path to local system temporary database device	/dev/raw/raw17
Local system temporary database device size (MB)	40MB
Local system temporary database name	mycluster_tdb_3
Local system temporary database size (MB)	40MB

Do you want to add another instance (Y or N)	N
Save configuration information in a file [Y]	Yes
Full path for configuration file	/remote/var/sybase/mycluster.xml
Create the cluster now [Y}	Yes

Sample session

```
sybcluster -Uuafadmin -P -F "blade1:9999, blade2:9999, blade3:9999"
> create cluster
Please enter the name of the cluster: mycluster
Cluster mycluster - Please enter the maximum number of instances:  [ 4 ]
How many nodes (hardware hosts) will participate in this cluster:  [ 3 ]
Verifying the supplied agent specifications...
       1) blade1 9999 2.5.0 Linux
       2) blade2 9999 2.5.0 Linux
       3) blade3 9999 2.5.0 Linux
Enter the number representing the cluster node 1: [ 1 ]

------------------ Quorum Device  ---------------------
The quorum device is used to manage a cluster.  It contains information shared
between instances and nodes.
Enter the full path to the quorum disk: /dev/raw/raw11
Please enter any traceflags:

--------------- Master Database Device  ----------------
The master database device controls the operation of the Adaptive Server and
stores information about all user databases and their associated database
devices.
Enter the full path the master device: /dev/raw/raw12
Enter the size the Master Device (MB):  [ 80 ]
Enter the size the Master Database (MB):  [ 60 ]
Please enter the page size in kilobytes:  [ 2 ]

------------ Sybase System Procedure Device --------
Sybase system procedures (sybsystemprocs) are stored on a device.
Enter the System Procedure Device path: /dev/raw/raw13
Enter System Procedure Device size (MB):  [ 150 ]
Enter the System Procedure Database size (MB):  [ 135 ]

-------------- System Database Device ------------------
The system database (sybsystemdb) stores information about distributed
transactions.
Enter the System Database Device path: /dev/raw/raw14
Enter the System Database Device size (MB):  [ 12 ]
Enter the System Database size (MB):  [ 12 ]
-----------------------------------------------------
Does this cluster have a secondary network: [ Y ]
-----------------------------------------------------
Please enter the SYBASE home directory: /remote/var/sybase
Please enter the environment shell script path: [/remote/var/sybase/SYBASE.sh ]
Please enter the ASE home directory:  [ /remote/var/sybase/ASE-15_0 ]
```

```
Please enter the interfaces directory:  [ /remote/var/sybase ]
Enter path to the dataserver config file:  [ /remote/var/sybase/mycluster.cfg ]
-----------------------------------------------------

You will now be asked for the instance information on a node by node basis.
Enter the name of the node: blade1
-- Cluster: mycluster - Node: blade1 - Agent: blade1:9999 --

Enter the name of the cluster instance: ase1
Enter the interface file query port number for instance ase1: 19786
Enter the primary protocol address for ase1: 10.0.1.1
Enter the secondary protocol address for ase1: 10.0.1.2

--------------- Local System Temporary Database ---------
The Local System Temporary Database Device contains a database for each instance
in the cluster.
Enter the LST device name: mycluster1_tempdb
Enter the LST device path: /dev/raw/raw15
Enter LST device size (MB): 40
Enter the LST database name:  [ mycluster_tdb_1 ]
Enter the LST database size (MB): 40
-----------------------------------------------------
Do you want to add another instance? (Y or N):  [ N ] Y
Enter the name of the node: blade2
-- Cluster: mycluster - Node: blade2 - Agent: blade2:9999 --

Enter the name of the cluster instance: ase2
Enter the interface file query port number for instance ase2: 19786
Enter the primary protocol address for ase2: 10.0.2.1
Enter the secondary protocol address for ase2: 10.0.2.2

--------------- Local System Temporary Database ---------
The Local System Temporary Database Device contains a database for each instance
in the cluster.
Enter the LST device name:  [ mycluster2_tempdb ]
Enter the LST device path:  [ /dev/raw/raw16]
Enter LST device size (MB): 40
EEnter the LST database name: [mycluster_tdb_2
Enter the LST database size (MB): 40
-----------------------------------------------------------
Do you want to add another instance? (Y or N):  [ N ] Y
Enter the name of the node: blade3

-- Cluster: mycluster - Node: blade3 - Agent: blade3:9999 --

Enter the name of the cluster instance: ase3
```

```
Enter the interface file query port number for instance ase2: 19786
Enter the primary protocol address for ase3: 10.0.3.1
Enter the secondary protocol address for ase3: 10.0.3.2

--------------- Local System Temporary Database ---------
The Local System Temporary Database Device contains a database for each instance
in the cluster.
Enter the LST device name:  [ mycluster3_tempdb ]
Enter the LST device path:  [ /dev/raw/raw17]
Enter LST device size (MB): 40
EEnter the LST database name: [mycluster_tdb_3
Enter the LST database size (MB): 40
Do you want to add another instance? (Y or N):  [ N ] N

Would you like to save this configuration information in a file?  [ Y ] y
Enter the name of the file to save the cluster creation information:
[ /remote/var/sybase/mycluster.xml ]
-----------------------------------------------------------
Create the cluster now?  [ Y ] y
```

Configuring a cluster using an input file

At the end of a session, sybcluster asks if you want to save the values from the current session to an external file. You can use this file to re-create the same cluster, or edit the values in the file to create a different cluster.

The syntax is:

create cluster *cluster_name* file *xml_input_file*

This example creates “mycluster2,” using an input file named *mycluster.xml*:

```
create cluster mycluster2 file ./mycluster.xml
```

APPENDIX D

Creating Raw Partitions in Linux

Overview

Linux

Adaptive Server allows you to create and mount 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 can be reused with the disk resize command.
- 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:

```
ls -l <device path>
```

❖ **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.

I/O fencing provided by the SCSI-3 PGR feature operates on devices only, not on partitions. For example, */dev/sda1* and */dev/sda2* are partitions of the device */dev/sda*. A fencing operating targeted to a raw device bound to */dev/sda1* will affect all partitions of */dev/sda*. Therefore any file systems or other applications (including another Adaptive Server) utilizing partitions on that device will also be affected. For this reason the device must be used exclusively by the Adaptive Server cluster instance.

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 must 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 You can create partitions with the Linux default fdisk(8) utility. You must have "root" privileges to use the command fdisk. See 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 layout. The output is:

  ```
  Disk /dev/sdd: 64 heads, 32 sectors, 8683 cylinders
  Units = cylinders of 2048 * 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. You can assign the remaining partitions 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 when you see:

```
First cylinder (8013-8683, default 8013):
```

3 Accept the default by pressing Enter again, when you see:

```
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
```

The output is:

```
Changed system type of partition 5 to 60 (Unknown)
```

6 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.

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 administer partitions using the */etc/sysconfig/rawdevices* file.

This is a plain text file that contains comments and examples for possible configurations:

```
# 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 you have created them, bind the raw devices 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

Administer raw disk partitions in the */etc/raw* file, which 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 you have created them, bind the raw devices 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 Server

Once you have created the partitions and bound the devices for raw disk I/O, Adaptive Server can 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*.

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
  ```

- You should have root privileges to execute this command. If you do not, you may see the following output for the raw -qa command:

  ```
  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 size automatically. Enter the absolute path to the raw device when creating the master, sybsystemprocs, sybtempdb, or any other device.

APPENDIX E

Upgrading Servers with Replicated Databases

Introduction

After you have upgraded, you cannot scan any part of the transaction log that existed before the upgrade, so you must follow the procedure outlined below if your server contains replicated primary databases (this includes replicated RSSDs). This procedure helps to ensure that data from a replicated database has been successfully replicated to the replicate database.

Warning! Be sure to have a valid dump database and a dump transaction before executing the procedures in the following sections.

The procedures described here do not upgrade Replication Server itself. For information on upgrading Replication Server, see your Replication Server documentation.

These steps precede the database upgrade procedure:

- 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.

To determine whether your existing server contains replicated databases:

1 Use isql to connect to the Server you are upgrading.

2 If your server uses Replication Agent to replicate data, connect to the server and execute:

```
1> sp_config_rep_agent
2> go
```

The above command will display the list of databases enabled for replication agent.

If you are using other application to replicate data, LTM or Replicator run the following command in each database including the master database:

```
1> dbcc gettrunc
2> go
```

3 If the command returns "1" for "secondary 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 Before you begin the upgrade verify that subscriptions created by the create subscription command, with primary data in the databases being upgraded, have reached the "valid" state at the primary Replication Server. 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 attempt 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 ensure the transaction log has been drained off:

1 Wait for all remaining transactions to be replicated.

2 Check that a manual update on a replicate table is forwarded to Replication Server:

a If the database is enabled for Replication Agent:

```
1> use <dbname>
2> go
1> sp_config_rep_agent <dbname>, "traceon",
"9201"
2> go
```

If you cannot access to the console where Adaptive Server was started, you must also define a trace log file to check the information Replication Agent will send to it:

```
1> sp_config_rep_agent <dbname>,
"trace_log_file", "path for the log file"
2> go
```

b If the database is using a different method to replicate data, you must ensure that data is replicated by consulting Replication Server queues. Run the following Replication Server command:

```
1> admin who, sqm
2> go
```

1 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.

2 Open a queue dump file by executing the following Replication Server command, where *file_name* is the name of the file in which you will dump the queue

```
1>sysadmin dump_file, "file_name"
2>go
```

3 Use isql to update one row in a single replicated tale 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 Sybase recommends you use a dummy table with a primary key which you define. It is recommended that you define a replication definition with all replicateable columns

---

4 In the primary Replication Server, execute the admin who, sqm command until the last segment: block entry for the inbound queue changes.

5 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
```

6 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 instep 4.

7 Repeat steps 3-5 until the transaction that corresponds to the update is in the dump file.

8 Stop the application or process reading the transaction log from the primary database:

If you are using Rep Agent, log into the Adaptive Server, and stop the Rep Agent:

```
1> sp_stop_rep_agent database
2> go
```

If you are using LTM or Replicator stop them.

After draining the transaction logs, disallow all other activity in the databases. If activity does occur, you must redrain the logs.

9 Instruct Replication Server to reject incoming connections from Replication Agent, LTM or Replicator:

Log in to the Replication Server and suspend the Log Transfer connection from that database:

```
1> suspend log transfer from server.database
2> go
```

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 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 need not 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 , the Log Transfer Agent or Replicator must not be running, and you should turn off the secondary truncation point 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.

3 Truncate the primary transaction log running the command:

```
1>dump tran primary_dbname with truncate_only
2>go
```

4 Zero out the Replication truncation point as soon as you can stop Replication Server. In the RSSD for the Replication Server, run:

```
1>rs_zeroltm primary_servername, primary_dbname
2>go
```

APPENDIX F

Using sybsystemprocs

Overview

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 135MB, 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. 135MBs 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, 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, expand sybsystemprocs to a second device, or initialize another device that is large enough. See "Increasing device and database capacity for system procedures" on page 197.

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*.

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 must 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.
- *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.

Make a note of the names of the devices; you will need them in the folowing steps.

2 Drop sybsystemprocs:

```
use master
go
drop database sybsystemprocs
go
```

Note In versions of Adaptive Server Enterprise earlier than 15.0, use sysdevices to determine which device has a low through high virtual page range that includes the *vstart* from step 2.

In version 15.0, select the vdevno from *sysusages* matching, the dbid retrieved in step 1.

---

3 Remove the device or devices using sp_configure:

```
sp_configure "allow updates", 1
go
delete sysdevices
    where name in ("devname1", "devname2", ...)
go
sp_configure "allow updates", 0
go
```

The where clause contains the list of device names returned by the query in step 1.

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 you use 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 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*.

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
```

Index

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

W