

SYBASE®

Installation Guide

**Adaptive Server® Enterprise
Cluster Edition 15.5**

HP-UX

DOCUMENT ID: DC01083-01-1550-03

LAST REVISED: June 2010

Copyright © 2010 by Sybase, Inc. All rights reserved.

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

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

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

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

IBM and Tivoli are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

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

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

Contents

Adaptive Server® Enterprise Cluster Edition Overview	1
Options for Adaptive Server Cluster Edition	2
Installation Workflow	3
Style Conventions	4
Server Component Descriptions and Directory Layout	5
PC-Client Product Descriptions and Directory Layout	7
Planning the Installation	11
Obtaining a License	13
Accessing SPDC	13
Installing a New License Server	16
Enabling and Changing E-mail Notifications	19
System Requirements for Servers	19
System Requirements for Using Private Interconnect Technology	20
System Requirements for Clients	21
Preinstallation Tasks for the Server Installation	23
Adjusting the Operating System Shared Memory Parameters	24
Obtaining the SySAM Host ID	25
Installing the Server	27
Installing the Server in GUI Mode	27
Creating a Response File	29
Installing From the Command Line	29
Installing Server in Silent Mode	30
Uninstalling the Cluster Server	30
Uninstalling an Existing Adaptive Server	31
Creating and Starting the Cluster	33
Overview of the Basic Configuration Steps	33
Overview of a Shared Installation	33

Overview of a Private Installation	34
Before you Begin the Installation	34
Creating a Simulated Cluster on a Single Node	35
Setting the Environment and Starting the Unified Agents	35
Shutting Down the Unified Agent	36
About Creating a Cluster	36
Worksheet for Creating a Cluster	36
Creating a Cluster Using ASE Plug-in	42
Creating a Cluster Using sybcluster	50
Starting the Cluster When the Operating System Starts	54
Cleaning up After Failed Installation	55
Auxiliary Servers	55
Configuring Backup Server Using sybcluster	56
Configuring Backup Server Using ASE Plug-in ...	57
Installing Job Scheduler	57
Configuring XP Server Using sybcluster	58
Configuring Monitor Server Using sybcluster ...	59
Postinstallation Tasks	61
Setting the System Administrator Password	61
Installing Sample Databases	61
Default Devices for Sample Databases	62
Running the Database Scripts	62
Installing the Interpubs Database	63
Installing the Jpubs Database	63
Maintaining Sample Databases	64
PC-Client Installation	65
Installing the Client	66
Installing the Client in Silent Mode	67
Getting Started After Installing	67
Uninstalling PC-Client	71
Upgrading the Server	73
Preupgrade Tasks	73

Statistics in System Tables	75
System Catalog Changes During Upgrade	75
Checking System and Upgrade Requirements	76
Checking the runserver File Location	76
Procedure text is required For Upgrade	76
Reserved Words	76
Upgrading to a Private Installation	78
Manually Upgrade Overview	82
Manually Upgrading the Old Server	82
Preparing the Database and Devices for the Upgrade	
.....	86
Increasing Default Database Sizes	87
Upgrading the Server Using ASE Plug-in	88
Upgrading the Old Server with ASE Plug-in	88
Upgrading the Server Using sybcluster	90
Verifying the Server for Upgrade	90
Upgrading the Old Server Using an Input File	92
Upgrading the Old Server Interactively	93
Postupgrade Tasks	94
Updating .jar and .xml Files	95
Recovering from a Failed Upgrade	95
Upgrading Java in the Database	96
Migrating	97
Migrating From Version 15.x or Earlier Version	
Using BCP	97
Migrating a High Availability-Configured Server	
.....	98
Downgrading	99
Basic Downgrade Steps	99
Postdowngrade Tasks	101
Additional Steps If You Used 15.5 Cluster Edition	
Features Before Downgrading	102
Downgrading Job Scheduler	104
I/O Fencing	105
Enabling I/O Fencing	105

Setting Up I/O Fencing	106
Troubleshoot SySAM	107
Where to Look For SySAM Related Errors	107
Troubleshooting SySAM	108
Calling Sybase Technical Support	114
Troubleshoot the Server	117
Error Log Locations	118
Logs for Installation Utilities	118
Logs For Sybase Servers	119
Solutions to Common Installation Problems	119
If the Configuration Fails	122
Stopping the Server After a Failure	122
Troubleshooting Resource File Installations	123
If Adaptive Server Fails the Pre-upgrade	
Eligibility Test	123
Recovering From a Failed Installation or After	
the Files are Created	123
Recovering From a Failed Upgrade	123
If the Cause of the Failure is Known	125
Upgrading Compiled Objects with dbcc	
upgrade_object	126
Determining whether a compiled Object Has	
Been Upgraded	126
Finding Compiled Object Errors Before	
Production	127
Using dbcc upgrade_object	129
Using Database Dumps in Upgrades	132
Manually Setting Up and Managing the Cluster	133
Before You Begin	133
Setting the Environment	133
Creating the Raw Devices	133
Configuring the Cluster Manually	134
Create The Cluster Input File	134
Configuring the Interfaces File	137
Building the Master and Quorum Devices	138

Initializing and Creating sybsystemprocs	139
Installing the System Stored Procedures	139
Creating Runserver Files	140
Setting Up Local System and Temporary Databases	140
Configuring the Cluster for Automatic Takeover	141
Starting the Cluster	141
Starting the Cluster After a System Failure	142
Postconfiguration Tasks	142
Shutting Down the Cluster or the Instance	143
Shutting Down an Instance	143
Reconfiguring the Cluster	143
Enabling sybcluster and the Plug-in After a Manual Configuration	144
A sybcluster Sample Session Configuration Values	145
Sample sybcluster Session	149
Configuring the Cluster Using an Input File	152
Upgrading Servers with Replicated Databases	153
Determining If Existing Servers Contain Replicated Databases	153
Suspending Transaction Processing and Replication Activities	154
Draining the Transaction Logs for Primary Databases	154
Draining the RSSD Transaction Log	156
Disabling the Secondary Truncation Point	157
Using Sybsystemprocs	159
Increasing the Size of the Sybsystemprocs Database	159
Increasing Device and Database Capacity for System Procedures	160
Obtaining Help and Additional Information	163
Index	165

Adaptive Server® Enterprise Cluster Edition Overview

Learn about the Adaptive Server Enterprise Cluster Edition here.

The Cluster Edition allows a single installation, a private mode installation, and multiple installations 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; called a "server instance" or "instance." You can use a single configuration file to determine the configuration for all instances (a shared installation), or each instance can use a separate configuration file (a private installation).

The Cluster Edition allows you to scale workloads using multiple physical and logical clusters. If an instance in a cluster fails, one or more of the still-running instances take over the workload of the failed instance. The cluster determines the instance to which each client connects. 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 – Applications can continue to run as long as a single cluster member remains viable, even after several others have failed.
- Simple management – Data is shared by all instances and does not require data repartitioning through cluster membership changes.

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.

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 physically connected to individual instances. This logical connection allows Adaptive Server to redirect clients to various instances in the cluster and to dynamically provide clients highly available failover data.

Workload management

The Cluster Edition workload manager can customize workload management and failover for your business applications so that each performs most efficiently. The logical cluster provides individualized working environments.

Installation options

The Cluster Edition lets select one of these configurations:

- Shared installation – requires a shared file system created using a Network File System (NFS) or a cluster file system. A shared installation supports a single \$SYBASE installation directory, Adaptive Server home directory, and server configuration file.
- Private installation – does not use a shared file system. A private installation supports a separate \$SYBASE installation directory, Adaptive Server home directory, and server configuration file for each instance.

See the *Cluster Users Guide* for detailed information about the Cluster Edition.

Options for Adaptive Server Cluster Edition

Options that are available.

Table 1. Optional packages

Option	Platform availability	Description
Security and directory services	• HP-UX 64 • IBM AIX 64 • Linux 64 • Solaris 64	Provides lightweight directory services and network-based authentication and encryption using SSL and Kerberos.
Partitions	• HP-UX 64 • IBM AIX 64 • Linux 64 • Solaris 64	Enables semantic partitioning for table row data.
Encrypted Columns	• HP-UX 64 • IBM AIX 64 • Linux 64 • Solaris 64	Increases security parameters and allows for addition of datatypes.
Tivoli Storage Manager	• HP-UX 64 • IBM AIX 64 • Linux 64 • Solaris 64	Enables the database to backup and restore operations to IBM Tivoli Storage Manager.

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

Installation Workflow

Learn about the installation workflow and server installation prerequisites.

The Cluster Edition requires your network to be configured for the nodes included in the cluster.

1. Install operating system patches, and verify that the operating system meets the version-level, RAM, and network protocol requirements for your platform.
2. Before installing the software, shut down the clusters that use an earlier version of the Cluster Edition.
3. 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 using this “sybase” account.

4. Log in to the node on which you plan to install the first instance of the server 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.

5. If you are using an account that is local to only one physical box, create the sybase user account on each machine with the same user ID.
6. Decide where to install the Adaptive Server. Make sure there is sufficient available disk space. Do not use 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.
7. Ping the host to verify that your network software is configured.
8. Adjust the operating system shared memory parameter for your platform. See *Adjusting the Operating System Shared Memory Parameter* on page 24 for information on how to do this.
9. Obtain your host ID. See *Obtaining The Host ID* on page 25.
10. Decide which SySAM model to use. See *Obtaining a License* on page 13.
11. Access SPDC. See *Accessing SPDC* on page 13.
12. Generate a license. See *Generating a License* Learn how to generate a license for either a served or an unserved license model..
13. Run the installer from the command line, GUI mode or silent mode. Follow the license and install instructions in the wizard.

Style Conventions

Style conventions used.

- In a sample window display, commands to be entered exactly as shown appear 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, 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 2. SQL syntax conventions

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in a bold san-serif font.
<i>variable</i>	Variables, or words that stand for values that you fill in, are in <i>italic</i> .
{ }	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.
()	() 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, which are typed as part of the command.

Server Component Descriptions and Directory Layout

Learn about the server components and directory layouts here.

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™ 6.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/bin directory. - Job Scheduler templates and utilities – Predefined templates from which the database administrator can create and schedule useful, time-saving jobs. They are installed into the \$SYBASE/\$SYB-ASE_ASE/jobscheduler directory. - 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/\$SYB-ASE_OCS/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/SDCAADMIN - 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 Server Installed into the DataAccess directory: - ODBC (Windows, Solaris SPARC, Solaris x64, Linux Power, HP-UX Itanium, IBM AIX) – a driver used to connect to Adaptive Server from ODBC 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.
Java Client Utility	Installed into the jutils-2_0 directory and is a collection of Adaptive Server utility programs including: ribo – a tool to trace TDS traffic between client program and Adaptive Server.
Java Database Connectivity (JDBC)	Installed into the jConnect - 6_0_5 and jConnect - 7_0 directories. Provides a Java Database Connectivity (JDBC) driver for Adaptive Server.
Uninstaller	Installed into the sybuninstall/ASESuite directory.

PC-Client Product Descriptions and Directory Layout

Learn about the PC-Client product and directory layout here.

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 Server Installed into the DataAccess directory: • ODBC (Windows, Solaris SPARC, Solaris x64, Linux Power, HP-UX Itanium, IBM AIX) – 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.
Windows Cluster Server Admin utilities	Windows Cluster Administrator is a GUI tool for managing Microsoft Cluster Server (MSCS). It can be used to create, modify and displays the information of groups, resources and the cluster itself. There is also an alternative command-line tool called Cluster.exe for cluster administration.

Product	Description
PC-Client Administration Tools	The <code>ASEplugin.jar</code> file is located in <code>ASEP\lib</code>, while the <code>scjview.exe</code> executable to start up Sybase Central with the ASE plug-in is located in <code>shared\Sybase Central 6.0.0\win32</code>. - Sybase Central™ 6.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 <code>%SYBASE%\SDCAD-MIN-15_0\bin</code> directory. - Job Scheduler templates and utilities – Predefined templates from which the database administrator can create and schedule useful, time-saving jobs. They are installed into the <code>%SYBASE%\%SYB-ASE_ASE%\jobscheduler</code> 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 <code>%SYBASE%\%SYBASE_ASE%\bin</code> 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 <code>%SYBASE%\Shared\JRE-6_0_6_32BIT</code> directory. - The Administration Utility is installed to <code>%SYBASE%\SDCAD-MIN-15_0</code> - 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 <code>%SYBASE%\locales</code> directory. Provides system messages and date/time formats.
Character Sets	Installed into the charsets directory. Provides character sets available for use with Adaptive Server.

Product	Description
jConnect™	Installed into the <code>jConnect-6_0_5</code> and <code>jConnect-7_0</code> directories. Provides a Java Database Connectivity (JDBC) driver for Adaptive Server.
uninstaller	Installed into the <code>sybuninstall/PCClient</code> directory.

Separately Installable PC-Client Products

The PC-Client CD includes several additional products that install separately.

- InfoMaker® – a tool for personal data access, management, and reporting, used by both developers and end users. InfoMaker allows you to create InfoMaker presentation-quality reports and powerful queries. 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.

Planning the Installation

Learn about planning for the installation procedure.

Note: There has been a change to license quantity used by per-CPU and per-chip license types in version 15.5 Cluster Edition. Adaptive Server checks the same number of licenses as the number of cores on the machine (or chips, depending on the license type) regardless of any configuration settings. This behavior is not a change in licensing terms, but instead is a correction from earlier versions of Adaptive Server, in which if Adaptive Server was licensed per CPU or CPU chip, the license quantity requested was reduced if the **max online engines** configuration parameter was fewer than the number of CPUs on the machine.

Note: See the *Users Guide* for hardware requirements for using Infiniband, Interconnect on a production system. Sybase does not support file system devices when running on multiple nodes.

Note: If you intend to run the cluster under Symantec Storage Foundation for Sybase Cluster Edition, refer to Chapter 11, “Using the Cluster Edition with the Veritas Cluster Server,” in the *Clusters Users Guide* for more information on how to do this.

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 PGRs (such a configuration is supported for test and development environments that can tolerate the possibility of data corruption).

1. Create a \$HOME directory on the node on which you will run the installer.
2. Ensure that all nodes are running on the same operating system version.
The number of processors and the amount of memory can vary from node to node but the operating system version cannot.
3. Ensure that the quorum resides on its own device.
4. Create the local system temporary databases on a shared device using the Adaptive Server plug-in or **sybcluster**. Do this for each instance during the initial startup of the cluster and later on whenever you add an instance to the cluster.
You can create or drop a local system temporary database from any instance, but you can access it only from the owning instance.
5. Ensure that all database devices, including quorum devices, are located on raw partitions. Do not use the Network File System (NFS).

Warning! Do not use file system devices for clusters – The Cluster Edition is not designed to run on a file system; mounting a nonclustered file system on multiple nodes will immediately cause a corruption, leading to a total loss of the cluster and all of its databases.

For this reason, Sybase does not support file system devices when running on multiple nodes.

6. Ensure that the raw partitions are 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, use a single node or machine to run multiple instances of the Cluster Edition in a cluster configuration. When you do this, you must use the local file system (not NFS) or SAN Storage for the database devices.

7. Ensure that all hardware nodes use Network Time Protocol (NTP) or a similar mechanism to ensure that clocks are synchronized.
8. If you are using a shared installation, ensure that all Adaptive Server Enterprise software and configuration files (including the \$SYBASE directory, the interfaces file) are installed on a Network File System (NFS) or a clustered file system (CFS or GFS) that is accessible from each node in the cluster using the same access path. Supported versions of clustered file system are detailed in the next section.

If you are using a private installation, each node must have its own installation on a cluster file system.

9. Ensure that you have a high-speed network interconnection (for example, a gigabit Ethernet) providing a local network connecting all hardware nodes participating in the cluster.
10. Sybase recommends that each node in the cluster have two physically separate network interfaces:
 - A primary network – for cluster interconnect traffic.
 - A secondary network – for cluster interconnect traffic.

The primary and secondary networks should be physically separated from each other, and are needed for security, fault-tolerance, and performance reasons. For fault-tolerance, the two network cards should be on different fabrics so that a cluster survives network failure.

11. 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).

Obtaining a License

For simple installations, follow these instructions to obtain and install a SySAM license.

Prerequisites

- Before you install your product, obtain your license files from the *Sybase Product Download Center (SPDC)* before you install.
- Decide which SySAM license model to use:
 - The unserved license model – obtains licenses directly from the license file. To use an unserved license, save the license to the same machine on which you are installing the Sybase product.
 - The served license model – uses a license server to manage the license allocation to multiple machines.

To use a served license, you must have a license server. You can run the license server on the same machine or a different machine with a different operating system and architecture.

To use an existing license server, you must know the server host name and port number.

Note: There can be only one instance of a SySAM license server running on a given machine. To setup a SySAM 2.0 license server on a machine that is already running a SySAM 1.0 license server, you must migrate the old license server to SySAM 2. A migrated license server can serve licenses for both SySAM 1.0 and SySAM 2-enabled products.

Refer to the *Sybase Software Asset Management Users Guide* for more information about:

- Served and unserved licenses, and to decide which license model is best suited for your environment.
- Choosing a license server host.
- License server administration.
- Migrating a license server.
- If your installation is complex, for example it uses three-server redundancy, high availability, or virtualization, Sybase recommends that you see the *Sybase Software Asset Management Users Guide* about correctly setting up your license model.

Accessing SPDC

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

1. After you order a Sybase product, you receive an email message 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 message.

Planning the Installation

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.

Next

Sybase Product Download Center

Generating a Served License

Before you install your Sybase product set up the license server and required licenses.

Prerequisites

Before generating the licenses, determine the host ID. See *Obtaining the SySAM Host ID* on page 25.

- If you have problems, see the Troubleshooting SySAM topic.

Task

1. From the welcome screen on the SPDC site, select the product family that contains the product for the license you want to generate.
2. From the list of Sybase products select the appropriate product edition, and version. Make sure you select the product for the operating system you are using. The license agreement displays.
3. If the license agreement screen displays, select the country or region to display the applicable license text, read the agreement, then click **I Agree**.

Note: The license agreement appears only the first time you attempt to download a product. Once agree to the license, you do not see the license agreement for all subsequent downloads of that product.

4. On the product download page, click **License Keys**. The License Information page displays.
5. Select the license to generate.

Verify that you are generating the correct license by comparing the product name, edition, and license type with your company's purchase order.

Note: If your product requires both a server and a client license, Sybase recommends that you generate the server license first.

6. Click **Select to Generate**.
7. Select **Served License**, then click **Next**.
8. Enter the quantity of licenses to generate, then click **Next**. You can generate licenses for a maximum of ten machines.
9. Enter the License Server Host ID and, optionally, the host name and port number.

- a) A port number is not required unless you are using a three-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 you do not specify a TCP/IP port number, one of the default ports in the range of 27000 and 27009 is used.
- b) To generate a license for a three-node redundant cluster, specify the host ID, host name, and port number for each node of the cluster. In this configuration the port number is required and should be outside of the 27000 to 27009 range.
- c) Specify redundant server information to configure your servers for three-server redundancy.

10. Click **Generate.**

11. If you have just generated a server license your product also requires a client license, repeat steps 5-10 above for the client license.

12. Once you have generated all the licenses your product requires, you can:

- Download the license file by clicking the **Download All Licenses for Host** button.
- Print a paper copy of the license by clicking the **Print Friendly** button.
- Return to the license information page to generate additional licenses.

13. After generating the licenses, save them with a `.lic` extension to the `licenses` directory in your license server installation.

Generating an Unserved License

Learn how to generate an unserved license.

Prerequisites

Before generating licenses, determine the host ID. See *Obtaining the SySAM Host ID* on page 25.

- If you have problems, see the Troubleshooting topic.

Task

1. From the welcome screen on the SPDC site, select the product family that contains the product for the license you want to generate.
2. From the list of Sybase products select the appropriate product edition, and version. Make sure you select the product for the operating system you are using. The license agreement displays.
3. If the license agreement screen displays, select the country or region to display the applicable license text, read the agreement, then click **I Agree**.

Note: The license agreement appears only the first time you attempt to download a product. Once agree to the license, you do not see the license agreement for all subsequent downloads of that product.

Planning the Installation

4. On the product download page, click **License Keys**.

The License Information page displays.

5. Select the license to generate.

Verify that you are generating the correct license by comparing the product name, edition, and license type with your company's purchase order.

Note: If your product requires both a server and a client license, Sybase recommends that you generate the server license first.

6. Click **Select to Generate**.

7. Select **Unreserved License** and click **Next**.

8. Select the number of machines to license and click **Next**. You can generate licenses for a maximum of ten machines.

9. Enter the host ID, and optionally, the host name for each machine for which you want to generate a license.

For some license types, you must enter the number of CPUs or licenses for the machine where this license is used.

10. Click **Generate**.

11. If you have just generated a server license your product also requires a client license, repeat steps 5-10 above for the client license.

12. Once you have generated all the licenses your product requires, you can:

- Download the license file by clicking the **Download All Licenses for Host** button.
- Print a paper copy of the license by clicking the **Print Friendly** button.
- Return to the license information page to generate additional licenses.

13. After generating the licenses, save them with a `.lic` extension to the `licenses` directory in your license server installation.

Installing a New License Server

Before installing a new license server ensure that your system is updated with the correct operating system patches for your platform.

1. Insert DVD into the appropriate drive. Mount the DVD by running the following command:

```
mount -F cdfs -o ro,rr <CDROM device> /cdrom
```

2. To install the SySAM Suite using the GUI tool, launch the executable file and follow these steps.

3. To install the SySAM utility and license server separately follow these substeps:

- a) Add a Java Runtime Environment 1.6 to `$PATH`.

For C Shell enter:


```
setenv PATH <JRE 1.6>/bin:$PATH
```

For Bourne Shell enter:

```
set PATH=<JRE 1.6>/bin:$PATH; export PATH
```

- b) Launch the SySAM Utility installer program from command prompt in the CD drive. Enter:

```
./archives/sysam_util/setup.bin
```

- c) Launch the SySAM License Server installer program from command prompt in the CD drive. Enter:

```
./archives/sysam_server/setup.bin
```

4. The Welcome Window displays. Click **Next**.
5. Select the most appropriate country from the drop-down list.
6. Accept the license agreement.
7. Enter or select the destination directory.
8. Select **Typical Installation Set**.
9. Click **Install** to proceed with the installation.
10. View the panel that reports installation success or failure.
11. Start the license server.

Note: The license server cannot be started until there is at least one valid served license installed in the `licenses` directory located at `$SYBASE/SYSAM-2_0/licenses`.

12. If you plan to use sub-capacity licensing see the *SySAM Users Guide* for information on how to configure the **sysamcap** utility.
13. Obtain the host ID for the machine on which you run the license server. For information on how to generate the host ID, see Getting your host ID.
14. Go to the *Sybase Product Download Center (SPDC)*.
15. Generate the licenses for the products you want to install. For instructions on generating your licenses, see the Generating licenses at SPDC topic.
- 16.

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

- C Shell

```
source SYBASE.csh
```

- Bourne Shell

```
./SYBASE.sh
```

17. Copy the license file in the `$SYBASE/SYSAM-2_0/licenses` directory on the network license server machine.
18. Refresh or restart the license server:

1. Change to:

```
$SYBASE/SYSAM-2_0/bin
```

2. 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
```

19. Refresh or restart the license server:

20. 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:

```
> sysam start
```

```
Starting the SYSAM service
The SYSAM service is starting.
The SYSAM service was started successfully.
```

21. 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:

```
Starting the SYSAM service
The SYSAM service is starting.
The SYSAM service was started successfully.
```

```
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_603020315.lic
```

22. The SySAM License Server installer installs both IPv4 and IPv4/IPv6 dual stack License Server binaries during installation and configures use of IPv4 only binaries. You can use the SySAM script to configure the appropriate version of the license server:

```
sysam configure [IPv6|IPv4]
```

For example, this command configures use of the IPv4/IPv6 dual stack binaries:

```
sysam configure IPv6
```

23. The IPv4 version of the license server cannot be used properly if you have enabled IPv6 on the license server host. If this is the case, you can:

- Disable IPv6 TCP/IP protocol on the license server host.
- Use another Windows host that does not have IPv6 enabled.

- Use a UNIX host for the license server. License Servers on UNIX are able to serve licenses to both IPv4 and IPv6 client hosts.
- If you must use a Windows host for the License Server and have both IPv4 and IPv6 clients on the license server, setup two License Servers, one serving the IPv4 network and the other serving the IPv6 network.

If you are using the IPv6 version of the license server on a Windows host, then this license server can be accessed only through IPv6 protocol. Any host with an IPv4 only network stack will not be able to obtain licenses from this IPv6 license server.

If you start the SySAM server from Windows services in Windows 7.0, Vista, or Windows 2008, the server starts successfully without any extra steps.

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.

System Requirements for Servers

Before installing Adaptive Server, ensure that your system is updated with the latest patches and system requirements. Do not use a patch that is earlier than the version listed below for your operating system. However you should use the patch recommended by the operating system vendor, even if it supersedes the patch listed here.

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

- For HP-UX:
swlist

Note: The Java Virtual Machine (JVM) and associated Adaptive Server support requires a minimum of 250MB of virtual memory space to start. You may need more memory depending on the requirements of the individual java executions. You may need to adjust the memory parameters when running java to ensure that the java has enough virtual memory space to start, and that both Adaptive Server and java tasks successfully coexist.

In particular, systems that have Adaptive Server total memory set to a value higher than 1.5GB (or 2.5GB on some Enterprise servers) may experience problems and may have to reduce Adaptive Server total memory.

Table 3. Adaptive Server specifications for HP-UX

Hardware and Memory	
Processor 64-bit	Intel Architecture-64 Itanium
Minimum RAM required for Adaptive Server	128MB
Default user stack size	148KB
Minimum RAM per additional user	Approximately 430KB

Table 4. Operating system requirements for HP-UX

Hardware	Operating system	Updates	Suggested minimum RAM
Intel Itanium	HP-UX B.11.31	PHSS_36352, PHKL_37802, PHKL_38102	1G

Table 5. Minimum disk space requirements for HP-UX

Platform	Typical installation	Default data-bases	Management features	Total disk space requirements
HP-UX Itanium	1,372MB	150MB	25MB	1,547MB

Table 6. Clustered file system for HP-UX

Operating system	Product
HP-UX B.11.31	Veritas Cluster File System 5.0

System Requirements for Using Private Interconnect Technology

The Cluster Edition supports only the UDP network protocol for private interconnects; Do not use the TCP network protocol.

A private interconnect is a physical connection that allows internode communication and is an essential component of a shared disk cluster installation. 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 you configure more than two nodes a switch that enables high-speed communication between the nodes in the cluster is required.

To handle the amount of traffic created by contention, use scalable interconnect technology to connect nodes. 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.

The Cluster Edition supports the current standards for interconnects. Sybase recommends that you research the available interconnects to find the one that works best for your site.

System Requirements for Clients

Learn about the system requirements for PC-Client.

Table 7. Operating system requirements for the PC-Client

Product	Hardware	Operating system	Updates	Suggested minimum RAM
PC-Client	P4 1.0GHz	WinXP Pro SP2, Win2003 Server SP1, WinVista Business Edition		512MB

Note: If you are using ODBC, OLE DB or ADO.NET Drivers, then you must have Microsoft .NET Framework 2.0 Service Pack 1 installed on your system. The installer produces errors during installation and the drivers do not function without this service pack installed.

To check whether you have this service pack installed:

- Open Add or Remove Programs item from the Control Panel. If .NET Framework is listed under the Currently installed programs then you do not need to update your system.
-

Preinstallation Tasks for the Server Installation

Execute these tasks before installing the server.

1. Ensure that there is at least 1 gigabyte of space available in the temp directory before beginning the installation.
2. Install the appropriate operating system patches on each node.
3. Shut down all clusters that use the previous version of the Cluster Edition before beginning the software installation.
4. If you are having trouble starting the installer, make sure you have the required operating system patches for the Java Runtime Environment (JRE) version 6. Information about the required system patches is available from the *HP Java Web site*.
5. Create a “sybase” account on your system to perform all installation tasks.
 - a) This account should own all devices and files, and must have permission to read and write all devices the cluster uses.
 - b) Create all devices with the “sybase” account.
 - c) Ensure that all disk devices are accessible from all the nodes in the cluster.
 - d) Ensure that the account used to start the cluster has permissions to read and write to all of the disk devices.
 - e) Ensure that the cluster and the SCSI generic driver have write permission on the /dev/sg* files that correspond to configured database devices.

The SCSI driver expects write access to /dev/sg* files for SCSI-3 PGR commands used in IO fencing.

6. Ensure that the current shell has the appropriate inheritable privileges.

```
sudo lssecattr -p $$
    487528 eprivs= mprivs= iprivs=PV_KER_RAS lprivs=PV_ROOT
uprivs=
> $SYBASE/ASE-15_0/bin/iofenceutil /dev/rhdisk2
/dev/rhdisk2
```

- a) If it is not fence capable, grant the current shell the appropriate inheritable privileges.

```
sudo setsecattr -p iprivs=+PV_KER_RAS $$
Then restart UAF
```

7. Use the **ls -l** commands to verify paths and file permissions.
8. Use the **dd** utility to verify that the sybase account can read and write to the devices.
9. If you need to change the permissions, use **chmod** or **chown** to correct the write permissions for /dev/sg* files. You can change the access permissions for these files to root only after rebooting the machine. Write permissions should be corrected prior to starting the server.

10. 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.
11. If you do not have open administrative privileges you must create a destination directory prior to running InstallAnywhere.
12. 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.
13. Decide where the Adaptive Server software will be installed.
 - a) Make sure there is sufficient disk space and temporary directory space available.
 - b) Ensure that there are no spaces in the path name of the directory.
 - c) If this is a shared installation, the \$SYBASE location must be on a shared file system, accessible from all cluster nodes using the same path.
 - d) If this is a private installation, install the Adaptive Server software on each node of the cluster.
 - e) Each instance in the cluster must have its own \$SYBASE directory.

The private installation mode does not use a Network File System (NFS) or cluster file system.
14. Verify that the operating system meets the version-level, RAM, and network protocol requirements for your platform.
15. 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.
16. Adjust the operating system shared memory parameter for your platform. See *Adjusting the Operating System Shared Memory Parameters* on page 24 for information on how to do this.

Adjusting the Operating System Shared Memory Parameters

Adjust shared memory segments if Adaptive Server cannot obtain large memory as a single segment or if Backup Server stripes fail due to insufficient segments.

Depending on the number and types of devices used 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. There are six default shared memory segments available for process attachments.

If any reconfiguration through **sp_configure** requires additional memory, Adaptive Server allocates shared memory segments after start-up. To account for these additional segments, allocate the maximum memory available to Adaptive Server by using the **allocate max shared memory** configuration parameter. See the *System Administration Guide*.

1. Adaptive Server 15.5 requires an additional 32MB of shared memory than earlier versions of Adaptive Server. Increase the value of the configuration parameter '**max memory**' to accommodate the additional memory requirement.

To query the amount of shared memory:

```
# kctune shmmax
```

To increase the amount of shared memory. For example:

```
# kctune shmmax+=33554432
```

The default total logical memory parameter for HP is 45,056 2K pages (88MB) on the 64-bit operating system.

2. To adjust the shared memory value of the operating system, use the System Administration Manager (SAM). After you install Adaptive Server, change any configuration parameter, procedure cache, or data cache size. This may require that you increase the value of the **max memory** configuration parameter.

Obtaining the SySAM Host ID

Learn how to obtain your SySAM host ID.

1. Change to \$SYBASE/SYSAM-2_0/bin, Where *\$SYBASE* is the Sybase installation directory.
2. To obtain a host ID for a served license 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 that you use 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 an appropriate native operating system command to get additional details. See the SPDC FAQ topic titled, "What's my Host ID."

3. Make note of the host ID so you can use it later at SPDC.
 - Your host ID is platform-specific, but may be similar to:
 - 00400516E525 on Linux platforms.
 - 170a3472 on Solaris platforms.
 - fb491d9 on AIX platforms.

Preinstallation Tasks for the Server Installation

- ID_STRING= 972179e8-7b2b-11d7-bd55-4a651849415a on HP Itanium

Installing the Server

If you have a working Adaptive Server on your system, do not install the Cluster Edition in the same \$SYBASE directory. If you must install other products into the same directory, install the Cluster Edition last.

Warning! Adaptive Server 15.5 uses InstallAnywhere for installation while past versions of Adaptive Server as well as other Sybase products use InstallShield Multiplatform for installation. These two types of installers conflict and necessary files are silently overwritten. Therefore, do not install products with these different installers in the same directory.

Warning! If you install the Cluster Edition into the same directory as earlier versions of the same components, the Cluster Edition overwrites the older versions. Installing other products on top of the Cluster Edition will likely result in one or more of the products not working correctly.

If you perform a private installation Sybase recommends that you install Adaptive Server on each node in the cluster in its own directory.

InstallAnywhere supports 508 functionality with some limitation. Please note these limitations:

- The **Choose Install Set** screen — installer does not support the use of **hot key** to select install type. You must use the **Tab** key to change the focus and the **space** key to choose.
- The **Choose Product Features** screen you must use a mouse to choose features. The keyboard does not work.

Installing the Server in GUI Mode

Learn how to install the server components in GUI mode.

If you are installing the Cluster Edition in private installation mode, use the installer for each instance in the cluster. This ensures that environment variables, soft links, and so on are set correctly for each instance.

Sybase recommends installing products as a system administrator; however, root permissions are not required to run the installer. The Installer creates the target directory (if necessary) and installs the selected components into that directory. At the end of the installation, you can verify the product installation. You may need to perform additional configuration procedures before you can use some products.

You must source the environment variables needed for Adaptive Server products by running the SYBASE . CSH script file after exiting the Installer.

1. Shut down all programs before running the installer.
2. Insert the Adaptive Server CD, or download and extract the Adaptive Server installation image from the Sybase Product Download Center (SPDC).
3. For a CD drive enter:

```
mount -F cd9660 -o ro,rr <CDROM device> /cdrom
```
4. Launch the Installer. For a CD drive enter:

```
/cdrom/setup.bin
```
5. Select a directory for the installation. Click **Next** to accept the default of:

```
/opt/sybase
```
6. If you are updating your pre-existing server, you will not see this step. Choose the type of installation to be performed:
 - **Typical**
 - **Full**
 - **Custom**
7. If you select **Custom**, the **Products and Features Selection** window displays. Select the products to install by checking the box next to the product name. Some features are dependent on other features; therefore, the installer prevents you from unselecting some items without first unselecting others.
8. Select the most appropriate country from the drop-down list, read the license terms, and then click **I agree**. Click **Next**.
9. Enter the license key here. If you do not have a license key choose to continue without and click **Next** to view the summary screen. However, if you chose to install a served license, click **Next**, to set up the license server.
10. Select the license type for the Cluster Edition:
 - **CP** – CPU license
 - **DT** – development and test license
 - **AC** – OEM Application Deployment CPU license
 - **LP** – application specific CPU license
 - **Unknown** – without a license
11. Configuring your server for e-mail notification enables designated users to 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

12. The Product Selection Summary window displays the selections that you have made. Verify that you have selected the correct type of installation, and that you have enough disk space to complete the process. Click **Next**.
13. The Install Progress window shows the progress of the installation. The Install Status window displays the result of the installation process.
14. You are given the option to have ASE plug-in remember passwords. Select **Enable** or **Disable** . Click **Next**.
15. The Installation Complete window is the last window. If you encounter any errors, see the *Troubleshooting Guide*. At the end of a successful installation click **Done**.

Creating a Response File

To perform a silent, or "unattended" installation, run InstallAnywhere and provide a response file that contains your preferred installation configuration.

To create a response file when installing in GUI or console mode, specify the **-r** command line argument. The **-r** argument records your responses to the installation wizard's prompts and creates a response file when the Install anywhere Wizard exits. The response file is a text file that you can edit to change any responses prior to using it in any subsequent installations.

To generate the response file during GUI installation, run:

```
setup.bin -r responseFileName
```

-r responseFile is optional. If specified, the installation session is recorded to response file. Where *responseFileName* is the absolute path of the file name you choose for the response file, for example:

```
/tmp/responsefile.txt
```

Note: The directory path you specify must already exist.

Installing From the Command Line

The steps for installing components in console mode are the same as those described in the Installing in GUI Mode topic, except that you run the installer from the command line using the setup command, and you enter text to select the installation options. Look here to learn how to install from a command line.

1. At the command line, enter:

```
setup.bin -i console
```

The installation program starts.

2. In cases where the installation program 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.

Installing Server in Silent Mode

After running the installer in GUI mode and recording the responses into a response file, edit the file to customize the responses. Then use that response file to answer the installer when you run the installer in silent mode.

Prerequisites

Use

```
setup.bin -r responseFileName
```

to generate a installation response file during a console or GUI installation.

To install in silent mode, run:

```
setup.bin -f responseFileName -i silent  
-DAGREE_TO_SYBASE_LICENSE=true -DRUN_SILENT=true
```

where *responseFileName* is the absolute path of the file name containing the installation options you chose.

Note: You must agree to the Sybase License Agreement when installing in silent mode. You can either:

- Include the option
-DAGREE_TO_SYBASE_LICENSE=true -DRUN_SILENT=true

in the command line argument.

- Edit the response file to include the property

```
AGREE_TO_SYBASE_LICENSE=true  
RUN_SILENT=true
```

Except for the absence of the GUI screens, all actions of InstallAnywhere 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.

Uninstalling the Cluster Server

Actions performed by the Installer are removed during an uninstallation process. Files, or registry entries created after installation are not removed and must be manually removed when the installation process has finished.

To uninstall the Cluster Edition, run:

```
$SYBASE/sybuninstall/ASESuite/uninstall
```

Uninstalling an Existing Adaptive Server

To remove an existing server, follow these instructions.

Prerequisites

Shut down the server before proceeding with these instructions.

Task

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. Remove any operating system files for database devices that exist.

Creating and Starting the Cluster

Learn how to configure and start the cluster server. Follow these instructions after 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. However, you can also configure and manage the cluster manually. See *Manually Setting Up and Managing the Cluster topic* on page 133 for information on how to do this.

Overview of the Basic Configuration Steps

Learn the basic steps for configuring the cluster server.

1. Read *Before you Begin the Installation* on page 34.
2. Set the \$SYBASE environment and start the Unified Agent on each node of the cluster.
3. Configure the cluster using one of the following:
 - The Adaptive Server plug-in.
 - The **sybcluster** utility.
4. Optional – configure the auxiliary servers: Monitor Server, XP Server, Backup Server, and Job Scheduler.
5. If the installation and startup fails you must follow the steps at the end of this section before re-installing.

Overview of a Shared Installation

In a shared installation, 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 input file.
- The *interfaces* file, unless LDAP is used for server lookup.
- A quorum device used to coordinate the instances with the cluster.

Overview of a Private Installation

In a private installation, each instance of Adaptive Server in the cluster shares the directories and files outlined here.

- All databases and database devices (for example, all instances share the same *master* database)
- A quorum device used to coordinate the instances with the cluster

In a private configuration, each instance maintains its own:

- \$SYBASE directory
- Interfaces file, unless LDAP is used for server lookup
- Server configuration file

Note: Although each instance has its own server configuration file, the quorum device maintains the master configuration file. Adaptive Server uses the master configuration file to ensure that each instance has the latest version of this file.

Before you Begin the Installation

Before you begin, determine these conditions:

- Whether the cluster installation will support a shared (NFS) or a private installation.
- Whether the cluster will use LDAP (recommended for private installations) or interfaces files for server lookup information.
- Will the cluster support Java in the database via a third-party JVM?
- Whether the cluster will support a Veritas Cluster Server (VCS) for managing application servers in the clustered environment.
- Whether the cluster will use a single or multiple Backup Servers.
- The location of the \$SYBASE installation directory. If this is to be a private installation, you must provide a separate \$SYBASE installation directory for each instance.
- 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 agents in the cluster. 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. Local user temporary databases do not require shared storage and can use local file systems created as private devices. However, local system temporary databases you create during the cluster configuration can use only shared disks. See Chapter 8, “Using

Temporary Databases” of the *Cluster User Guide* for additional information on using local disks for temporary data.

- 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 or network names of the private primary and secondary network cards on each node. Only UDP network protocol is supported at this time.

Creating a Simulated Cluster on a Single Node

Test the server configuration and installation.

Sybase recommends that you configure each instance on a different node, that is, 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.

1. For best performance make sure the number of engines in all instances running on a single node do not exceed the number of CPUs on that node.
2. The default setting for the **runnable process search count** parameter is '3'. Sybase recommends using this default setting. 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.

The default port for the Unified Agent is 9999. If for some reason you prefer to start the agent on a different port, see step 3.

From the \$SYBASE directory of each instance:

1. Source either the SYBASE.csh or SYBASE.sh environment scripts.
2. Start the Unified Agent:

```
$SYBASE/UAF-2_5/bin/uafstartup.sh &
```

Note: The Unified Agent creates \$SYBASE/UAF-2_5/nodes/<node_name>/log/agent.log, and sends agent output to that file.

3. Optionally, to specify the UAF port number, where *port_number* is any available network port, use:

```
$SYBASE/UAF-2_5/bin/uafstartup.sh -port port_number
```

Shutting Down the Unified Agent

Learn how to shut down the Unified Agent.

To shutdown a Unified Agent running on a specific port number, enter:

```
$SYBASE/UAF-2_5/bin/uafshutdown.sh -port port_number
```

For example, to shutdown a UAF agent that is started on port number 12345:

```
$SYBASE/UAF-2_5/bin/uafshutdown.sh -port 12345
```

Note: Do not cancel your Unified Agent process by using Ctrl-C to terminate a parent Java process on a running server; this also kills any Adaptive Server and other auxiliary server instances the agent started.

About Creating a Cluster

Learn how to create a 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.

Creating a cluster with Storage Foundation for Sybase Cluster Edition

Before you create a cluster with Storage Foundation for Sybase Cluster Edition, make sure you review Chapter 11, “Using the Cluster Edition with the Veritas Cluster Server,” in the Clusters *User Guide*. This chapter includes steps you must perform for the Veritas Storage Foundation software and for your operating system.

Worksheet for Creating a Cluster

The information you need to create a cluster using the Adaptive Server plug-in or **sybcluster** is listed in this worksheet.

Table 8. Configuration values for creating a cluster

Parameter	Default value	Minimum Value
Cluster name		
Number of instances	4	
Number of agents	4	

Parameter	Default value	Minimum Value
Configuration type		
Is the cluster to be configured using a private \$SYBASE installation?	N	
Page Size		
The page size in kilobytes:	2kB	
Master Device		
Full path and name for the master device		
Size of master device	30MB	
Size of master database	13MB	
Sybase System Procedure Device		
Full path and name for sybsystemprocs device		
Size of sybsystemprocs device (MB)	152	
Size of sybsystemprocs database (MB)	152	140
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 secondary network	Y	
Quorum Device		
Full path and name for the quorum device		
Trace flags		
PCI device (optional)		
Full path to the PCI database device		
Size of the PCI database device	24MB	
Size of the PCI database	24MB	
Verify file locations (for shared installation)		
The location of the home directory	\$SYBASE	

Creating and Starting the Cluster

Parameter	Default value	Minimum Value
Full path to environment variable script	\$SYBASE/SYB-ASE.sh	
Path to \$SYBASE_ASE	ASE-15_0	
Path to interfaces file directory	\$SYBASE	
Path to the dataserver configuration file	\$SYBASE/cluster_name.cfg	
Instance Information (for shared installation)		
Node name		
Instance name 1		
Query port number for instance1		
Primary protocol address for instance1		
Secondary protocol address for instance1		
Instance Information (for private installation)		
Node name		
Instance name		
Full path for \$SYBASE installation directory for instance 1		
Full path to environmental shell script for instance 1		
The Adaptive Server home directory for instance 1		
Full path to the server configuration file for instance 1		
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.) Device for a local system temporary database must be a shared disk.		
Path to local system temporary database device		

Parameter	Default value	Minimum Value
Local system temporary database device size (MB)		
Local system temporary database name		
Local system temporary database size (MB)		
Instance Information (for shared installation)		
Node name		
Instance name		
Query port number for instance2		
Primary protocol address for instance2		
Secondary protocol address for instance2		
Instance Information (for private installation)		
Node name		
Instance name 2		
Full path for \$SYBASE installation directory for instance 1		
Full path to environmental shell script for instance2		
The Adaptive Server home directory for instance2		
Full path to the server configuration file 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.) Device for a local system temporary database must be a shared disk.		
Path to local system temporary database device		

Creating and Starting the Cluster

Parameter	Default value	Minimum Value
Local system temporary database device size (MB)		
Local system temporary database name		
Local system temporary database size (MB)		
Instance Information (for shared installation)		
Node name		
Instance name		
Query port number for instance3		
Primary protocol address for instance3		
Secondary protocol address for instance3		
Instance Information (for private installation)		
Node name		
Instance name 3		
Full path for \$SYBASE installation directory for instance3		
Full path to environmental shell script for instance3		
The Adaptive Server home directory for instance3		
Full path to the server configuration file for instance3		
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.) Device for a local system temporary database must be a shared disk		
Path to local system temporary database device		

Parameter	Default value	Minimum Value
Local system temporary database device size (MB)		
Local system temporary database name		
Local system temporary database size (MB)		
Instance Information (for shared installation)		
Node name		
Instance name		
Query port number for instance4		
Primary protocol address for instance4		
Secondary protocol address for instance4		
Instance Information (for private installation)		
Node name		
Instance name 4		
Full path for \$SYBASE installation directory for instance4		
Full path to environmental shell script for instance4		
The Adaptive Server home directory for instance4		
Full path to the server configuration file for instance4		
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.) Device for a local system temporary database must be a shared disk		
Path to local system temporary database device		

Creating and Starting the Cluster

Parameter	Default value	Minimum Value
Local system temporary database device size (MB)		
Local system temporary database name		
Local system temporary database size (MB)		
Save configuration information in a file	Y	
Veritas Cluster Server support (optional - appears only if VCS is supported on your system)		
Integrate the Cluster Edition server with the VCS?	Y	
The path to the interfaces file on the cluster. Note: This prompt does not appear if LDAP is supported.		
Check if this device (master device, System Procedure Device, and System Database Device) is managed by the Veritas Cluster File System or Veritas Volume Manager?	Y	
Check each cluster device (master device, System Procedure Device, and System Database Device) for IO fencing capability?	Y	

Creating a Cluster Using ASE Plug-in

The Adaptive Server plug-in includes a wizard that steps you through the process of creating, and starting 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 *Cluster Users Guide*. Adaptive Server plug-in and **sybcluster** provide upgrade support.

1. Register the Adaptive Server plug-in.
2. Start Sybase Central.
3. Select the Adaptive Server Enterprise icon in the left pane.
4. Select the Utilities tab in the right pane.

5. Double click or right click on Create Cluster and select open to open the Create Cluster wizard. Or you can select **Tools > Adaptive Server Enterprise > Create Cluster** to start the Create Cluster wizard.
6. Enter the name of the cluster.
7. Select the platform from the drop-down list.
8. Select whether you are creating a Shared or Private installation.
9. Select the maximum number of instances you plan to use for this cluster. Click **Next**.
10. Discovery settings – you can change the settings the Adaptive Server plug-in uses to find nodes to use for the cluster. For more information see “Administering Clusters” in the *Users Guide to Clusters*.
11. Select the hosts that participate in the cluster. You can also click on Specify Node to specify a node that does not appear in the list of hosts.
12. Enter the unified agent user name and password. Select **Next**.
13. Quorum device path – enter the full path to the quorum device. Click **Next**.
14. If you are installing a private installation this step is skipped until you specify the instances in the cluster. If this is a shared installation you are prompted at this time to enter:
 - Sybase home – the Adaptive Server release directory (also the value of \$SYBASE).
 - Adaptive Server Enterprise home – the full path to the ASE-15_0 directory (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**
15. Enter the master device information:
 - Master device path – full path to the master device.
 - 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**.
16. If the device you specify does not include SCSI-3 PGR capability, Adaptive Server plug-in asks if you wish to continue. If you select **No**, the plug-in does not move to the next screen.
17. Enter the system procedures (*sybsystemprocs*) information:
 - System procedures device – 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**.

18. If the device you specify does not include SCSI-3 PGR capability, Adaptive Server plug-in asks if you wish to continue. If you select No, the plug-in does not move to the next screen.

19. Enter the system database (*sybssystemdb*) information:

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

If the device you specify does not include SCSI-3 PGR capability, Adaptive Server plug-in asks if you wish to continue. If you select No, the plug-in does not move to the next screen.

Click **Next**.

20. Select the checkbox to enter a PCI device. Enter the full path to the PCI device and enter the device and database size. Click **Next**. See *Java in Adaptive Server Enterprise* for more information about PCI devices.

21. If the device you specify does not include SCSI-3 PGR capability, Adaptive Server plug-in asks if you wish to continue. If you select **No**, the plug-in does not move to the next screen.

22. Verify the following or change to appropriate values:

- Network Protocol – select the UDP network protocol.
- Starting Port – select a starting port number for the cluster's network interconnects. Make certain this port number is not already in use.
- Check the box if you have a secondary network available.

23. Click **Validate Port** to make sure the ports are not currently in use, then click **Next**.

24. Specify the instances for the cluster. Click **Add** to add instances to the list.

25. In the Define Server Instance dialog, enter the following under the General tab 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.
- Log file – full path to the error log for this instance.

a) Enter the following in the Advanced tab:

- Primary – Select address from the dropdown.
- Secondary – enabled if you specified that a secondary address is available. Select host name or protocol address for the secondary connect.
- Start parameters – list any startup parameters for the instance.
- Traceflags – list any trace flags you want used for the instance at startup.

b) If it is available, enter the following in the Private configuration tab:

- Sybase home – the Adaptive Server release directory (also the value of \$SYBASE).

- Adaptive Server Enterprise 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.
 - Sybase environment script – full path to the script that you use to set the environment variables.
 - Cluster configuration file – full path to the cluster configuration file.
- c) Click **Next** to accept the cluster instance properties.
26. If there is more than one device on the node that supports IO fencing, the plug-in asks you to confirm the selection.
27. Specify the local system temporary databases and devices. You must create local system temporary database on a shared disk. The plug-in lists the default values for the local temporary databases. Accept the defaults or select each table cell to change its value.
28. 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** then click **Next**.
29. Review the information at the Summary page to be sure it is correct. 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.
30. Click **Next** to create this cluster.
- Adaptive Server plug-in displays another window showing the server messages as it creates the cluster.

Starting and Stopping Sybase Central

Learn how to start and stop Sybase Central.

To start Sybase Central:

- On UNIX platforms, run \$SYBASE/shared/sybcentral600/scjview.sh.
- On Windows platforms, 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.

1. On UNIX, run \$SYBASE/ASEP/bin/registerASEP.
2. On Windows, run %SYBASE%\ASEP\bin\registerASEP.bat.

3. To register the Adaptive Server plug-in manually:
 - a) From **Tools > plug-ins**, select **Register**. A registration wizard appears.
 - b) Select Register a Plug-in by specifying a plug-in registration file.
 - c) Click **Browse**. 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.

Adding an Instance to a Cluster

To add an instance, the **max instances** parameter must be greater than the current number of instances, and a cluster-supported agent must run on the node on which you are creating the instance. You must know the host name and port number of the unified agent (UA).

Note: I/O fencing requires that each instance be on a separate node, so you cannot add an instance to a node that already has one. A pop up dialog allows you to continue if you select Yes.

1. Open the Server Instances folder in the left pane to display the server instances and options in the right pane.
2. Select **Add Cluster Server Instance** to open the Add Cluster Server Instance wizard.
3. Follow the steps in the wizard to add an instance to the cluster.
4. Click **Finish**. The new instance is listed under the Server Instances view.

Adding Logical Clusters

The logical cluster is an abstract representation of one or more instances in a physical shared-disk cluster 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 failover. 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.

1. Select **Logical Cluster** from the toolbar or right click on the Logical Clusters folder and select **New Logical Cluster**. The Adaptive Server plug-in starts the Logical Cluster wizard.
2. Name the logical cluster. Select a name that is representative of the job this logical cluster will perform. For example, SalesLC.
Click **Next**.
3. 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**.

4. 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** then click **Next**.

5. 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 *Cluster Users Guide* 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**.

6. Define the options for the logical cluster (for more information about these options, see the “Managing the Workload” chapter of the *Cluster Users Guide*).

- 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.

Creating and Starting the Cluster

- 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.

a) Click **Next**.

7. Select **Finish** to build the logical cluster.

Connecting to a Cluster

When starting Sybase Central, the main window opens, displaying the ASE plug-in, with the list of icons of the clusters and instances to which you have previously connected. If the cluster is running, a green triangle appears inside the icon next to the cluster name.

Note: If you have difficulty connecting to a server or viewing it in the tree view in the left pane, check to make sure the Connection profile found in the toolbar under **Tools > Connect** contains the port number associated with the server in the `interfaces` file. If you are using PC-Client on Windows check the port number associated with the server in the `%SYBASE%\ini\sql.ini` file.

The fastest way to connect to a running cluster in the list is to right-click the cluster name and select **Connect**. The ASE plug-in uses the previous connection data to make the connection. If a cluster does not appear in the tree view, you can find it with Server Discovery or provide the cluster's host and port, login name, and password information. Either method starts by clicking the **Connect** icon in the toolbar near the top of the Sybase Central window.

If you know the required connection information, enter it into the appropriate fields in the Connect window. If you do not have the host and port number for the cluster or a cluster node, enter the login name and password and click **Find**. The Unified Agent searches for clusters and provide a list of those available. If the list does not include the cluster you are searching for, you need to change your discovery settings.

1. Select **Tools > Connect**.

If you are running multiple registered Sybase Central plug-ins, select the ASE plug-in.

2. Enter the login name you want to use to connect to the instance.

3. Enter the password for the login.
4. Select the cluster name from the Server Name list, which is populated with entries from the `interfaces` file, or type in the host and port of a cluster node. In Windows, with PC Client, it is populated from `sql.ini` file.
5. (Optional) Specify the host and port of an instance within the cluster.
6. Click **OK**.

Changing Server Discovery Settings

Learn how to change the discovery settings.

1. Navigate to **Tools > Adaptive Server Enterprise > Preferences > Server Discovery**.
2. Check **Enable Server Discovery**.
3. Select **Add**.
4. Select the discovery method from the scroll menu.
 - JINI – an open architecture that enables developers to create network-centric services that are highly adaptive to change. JINI offers a standard lookup service for discovery.
 - User Datagram Protocol (UDP) – a network protocol that provides a procedure for application programs to send messages to other programs with a minimal protocol mechanism.

Note: If UDP is used, only servers on the same subnet as the one on which Sybase Central is running are discovered.

5. If you selected JINI in the previous step:
 - a) Select the host of the JINI server
 - b) Select either the default host and port or enter new ones
6. Click **OK**.
7. To add or edit a discovery filter, click **Filters**. Server Discovery uses only the selected filters for its search. To specify a filter:
 - a) Click **Add**.
 - b) Select **Enable this Filter**.
 - c) Select the target you want filtered, Host, Name, OS, Platform, Port, Release type, Status, Version.
 - d) Select the condition: contains, does not contain, is, is not, starts with, ends with.
 - e) Enter the condition string you want Server Discovery to filter on.
 - f) Click **OK**.
8. Configure the ASE plug-in to discover clusters currently running on the system. Use
 - **Remove** – to remove a discovery service from the list.
 - **Edit** – to edit the settings for a current discovery service.
 -
 - **Up** – to move the selected discovery service up the list.

- **Down** – to move the selected discovery service down the list.

Creating a Cluster Using sybcluster

Follow the instructions in the Before You Begin the Installation topic and in the Setting the Environment and Starting the Unified Agents topic before configuring your cluster with **sybcluster**.

view a complete **sybcluster** session that includes answers to all the prompts in *Sample sybcluster Session* on page 149.

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 *Cluster Users Guide*.

Configuring the Cluster

Learn how to configure the cluster here.

1. Start **sybcluster**. For example:

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

See the *Clusters Users Guide* for complete syntax and usage information for **sybcluster**.

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 value, press **Enter**. Otherwise, enter the correct value, then press **Enter**.

3. **sybcluster** prompts for the following. See *Worksheet for Creating a Cluster* on page 36 for more information.

- 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 agents 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 . Select agents 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, the default setting for the **runnable process search count** parameter is '3'. Sybase recommends using this default setting in this environment. This can be set using the **sp_configure** stored procedure.

-
- The number representing the cluster agent. **sybcluster** numbers the remaining agents in the cluster starting with this number.
 - Is the cluster to be configured using private installation mode? The default value is no [N]; the cluster is to be configured for a shared installation.
 - The full path for the quorum device. For example, `/dev/raw/raw11`.
 - Trace flags.
 - The page size of the *master* database, in kilobytes.
 - The full path for the *master* device. For example, `/dev/raw/raw12`.
 - The size of the *master* device.
 - The size of the *master* database.
 - The full path to the system procedure database device, *sysprocsdev*. 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, *systemdbdev*. For example, `/dev/raw/raw14`.
 - The size of the system database device.
 - The size of the system database.
 - Enable the Pluggable Component Interface (PCI) to support Java in the database? [N]
If you enter "Y", enter:
 - The path to the PCI database device path. For example, `/dev/raw/raw20`.
 - The PCI database device size (24MB)
 - The PCI database size (24MB)
 - Does this cluster have a secondary network? (Y).
 - If you enter "Y" to a secondary network, **sybcluster** uses the default port number of 15100 as the starting port number for the range of port numbers required so that the instances can exchange messages, and reserves that number of ports, starting with the default value.

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.

- If you enter "N" to a secondary network, **sybcluster** asks for the beginning port number, and calculates the number of additional ports required and reserves that number of ports. The default value is 15100.
- If the cluster uses a private installation, go to step 4. If the cluster uses a shared installation, **sybcluster** prompts for:
 - 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 already contain the cluster or instance information. **sybcluster** adds this information 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 dataserer configuration file. For example, `/remote/var/sybase/mycluster.cfg`. **sybcluster** must be able to find this file during configuration.

4. **sybcluster** displays the node name and prompts for values for each instance, one node at a time.

- Instance name.
- If the cluster supports a shared configuration, **sybcluster** prompts for:
 - 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. For example, 10.0.1.1.
 - The secondary protocol address for the instance (if you answered Y to a secondary network). For example, 10.0.1.2.
- If the cluster supports a private configuration, **sybcluster** prompts for:
 - The \$SYBASE home directory
 - The environmental shell script path
 - The Adaptive Server home directory
 - The full path to the server configuration file

Note: The path for the server configuration file can be the same or different for each instance.

If you are using interfaces files for server lookup, **sybcluster** prompts for:

- The interface file query port number
- The primary protocol address
- The secondary protocol address
- Local system temporary database device. The name of the Adaptive Server database device for the local system temporary (LST) database. You must create local system temporary database on a shared disk.
- 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`
 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.
 8. While creating the cluster, **sybcluster**:
 - a) Checks to see if VCS is supported on your system. If yes, **sybcluster** asks:
 - Whether to integrate the cluster database with VCS [Y]. If yes, **sybcluster** prompts for:
 - The path to the interfaces file for each instance, unless LDAP is supported.
 - Whether to check if the master database, system procedure device, and system database devices are managed by the VCS Volume Manager or Veritas Cluster File System [Y]. If yes, **sybcluster** reports status for each device and asks if you want to continue.
 - b) If you answer no, **sybcluster** prompts for the path to the interfaces file for each instance, unless LDAP is supported.
 - c) Asks if you want to check if each cluster device supports IO fencing. If you answer yes, **sybcluster** checks all devices and reports the IO fencing capability of each one.

Verifying the Cluster Configuration

Verify that the cluster configuration 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 Unified 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 *Clusters Users Guide* 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:

Creating and Starting the Cluster

```
show cluster status
```

5. Verify the cluster configuration. Enter:

```
show cluster config
```

Starting and Stopping a Cluster Using sybcluster

Learn how to start and stop a cluster using **sybcluster**.

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]]..."
```

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. Enter:

```
connect
```

3. Start the cluster. Enter:

```
start cluster
```

4. To stop a cluster enter. 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. Do this by starting the Unified Agent on the host system and then executing the **sybcluster** commands to start the cluster using a shell script.

1. Start UAF
2. Verify that UAF is started successfully.
3. Execute **sybcluster** and pass a command file to **sybcluster** containing the command to start the instances on the system.
4. The command file is passed using the **sybcluster -i** command line parameter and looks something like this:

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

5. The **sybcluster** command line looks something like the following, where **asece1_startup** is the command file shown above:

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

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.

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. Remove the `< cluster name >.cfg` file if it exists after the last attempt.
6. Restart the UAF agents.

Auxiliary Servers

Learn how to configure the auxiliary servers with **sybcluster**. You can also configure the auxiliary servers Backup Server, XP Server and Monitor Server using Sybase Central. See the Help documentation for more information. You can access Help by clicking on **Help** in the Sybase Central toolbar.

Cluster Edition version 15.5 and later allows a cluster to use multiple Backup Servers with one of these methods:

- Dedicated method – each instance is assigned a specific Backup Server.
- Round-robin method – at the time of the **dump** or **load** command, the Cluster Edition assigns the instance a Backup Server from a group according to availability.
- A single Backup Server named SYB_BACKUP

For more information, see Chapter 5, “Using Backup Server in a Clustered Environment,” in the *Cluster Users Guide*

Backup Servers in a Shared Disk Cluster environment are available across all nodes of the cluster as a single Backup Server or to operate on only one instance in the cluster. 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.

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 the `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

Creating and Starting the Cluster

configured to run on this node in the interfaces file. If multiple Backup Servers are configured, the Backup Server is assigned based on whether the configuration is in dedicated or round-robin mode.

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.

The Backup Server binary (\$SYBASE/\$SYBASE_ASE/bin/backupserver) is installed when you install Adaptive Server version 15.5 Cluster Edition.

Configuring Backup Server Using sybcluster

Learn how to Configure Backup Server with **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 one or multiple Backup Servers for the cluster, enter:

```
create backupserver
```

5. sybcluster prompts you:

```
Do you want to create multiple Backup Servers?
```

a) If you answer “Y” for yes, **sybcluster** prompts for:

- A routing policy for the multiple Backup Servers. The choices are:
 - 1 – Dedicated.
 - 2 – Round robin
- Each Backup Server name. The default value is “*cluster_name_BS*”. For example, “mycluster_BS”.
- The path to each Backup Server log file.
- The listening port for each Backup Server.

b) If you answer “N” for no, **sybcluster** prompts you to configure Backup Server on the hosts, one by one.

- 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.

Sybase recommends that you configure Backup Server on all nodes so that the Backup Server can be started on any node. In some cases where Backup Server is not configured on all nodes, Adaptive Server cannot start Backup Server when Backup Server is not running. This happens when the **dump** command is initiated on a node that does not have Backup Server configured.

You also have the option to add or remove Backup Server from nodes at a later stage.

Configuring Backup Server Using ASE Plug-in

Learn how to configure Backup Server using the plug-in.

1. Select **Auxiliary Servers**. The right pane displays the Backup Server, XP Server, and Monitor Server tabs
2. Select the tab for the auxiliary server you want to configure.
3. Right-click the Auxiliary Server folder in the left panel.
4. Select the Configure Backup Server, Configure XP Server, or Configure Monitor Server heading provided.
5. Follow the instructions provided by the wizard.

Installing Job Scheduler

All instances in the cluster share a single Job Scheduler. 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.

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 JSAGENT in the interfaces file using either **dscp**, **dsedit**, or a text editor as appropriate. Sybase suggests that you name the entry "*clustername_JSAGENT*".

Creating and Starting the Cluster

To enable High Availability failover, the JSAGENT 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 “myxml1,” **uname -n** returns the value “myxml1.sybase.com,” and on host “myxml2,” **uname -n** returns the value “myxml2.sybase.com”. The correct entry for JSAGENT is:

```
mycluster_JSAGENT
  master tcp /dev/tcp myxml1.sybase.com 17780
  query tcp /dev/tcp myxml1.sybase.com 17780
  master tcp /dev/tcp myxml2.sybase.com 16780
  query tcp /dev/tcp myxml2.sybase.com 16780
```

Note: You must specify a port not currently in use.

See “{Volume 1} of the *System Administration Guide* for more information on directory services.

4. Using **sp_addserver**, create an entry in the *sys.servers* 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 sybmgmtadb
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 Using sybcluster

Configure an XP Server for each instance in the cluster using the **sybcluster create xpserver** command, which performs the configuration steps for you.

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

Configuring Monitor Server Using sybcluster

Configure a Monitor Server for each instance in the cluster using the **sybcluster create monitorserver** command.

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”:

```
connect to mycluster
```

3. Start the cluster:

```
start cluster
```

4. To configure Monitor Server enter:

```
create monitorserver
```

5. For each instance, Adaptive Server prompts you for:

- The Monitor server log file location for each instance
- The Monitor Server port number
- A user name
- A password

Postinstallation Tasks

Review the following topics to learn what to do after installing the server here.

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 Adaptive Server as “sa” and set a password using **sp_password**:

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

Where:

- **null** – is the default password
- **new_password** – is the password that you are assigning to the “sa” account.

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

Installing Sample Databases

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.

- **installpubs2** – installs the *pubs2* sample database, which 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, an updated version of *pubs2* that 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

Learn how to run the database scripts here.

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 the *Configuration Guide*.

Installing the Interpubs Database

interpubs is a database similar to the *pubs2* database which contains French and German data.

interpubs data contains 8-bit characters and is available for use at Adaptive Server installations using the ISO 8859-1 (`iso_1`), ISO 8859-15 (`iso_15`), 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.

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, 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
```

Installing the Jpubs Database

Learn how to install the Jpubs database here.

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

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 Adaptive Server 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**.
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
```

```
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
```

```
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 Sample Databases

Sybase recommends that you remove the "guest" user option from user databases in production systems.

The sample databases contain a guest user option that allows access to the database by any authorized 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*.

1. 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.
2. If space is a problem, instruct the user to issue the **begin transaction** command before updating a sample database.
3. After the user has finished updating one of the sample databases, he or she can issue the **rollback transaction** command to undo the changes.

PC-Client Installation

The PC-Client installer contains client applications and utilities you can use to access the Adaptive Server to execute queries or manage the server. It also contains the Sybase Open Client/ Open Server™ Software Developers Kit, which you can use to develop applications that access the server, 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 the Getting Started After Installing topic for information on configuring network connections.

The Sybase PC-Client CD includes the Software Developer's Kit (SDK) for Windows platforms, consisting of:

- Embedded SQL/C (ESQL/C)
- Embedded SQL/Cobol (ESQL/Cobol)
- Monitor Client Library
- XA Interface Library for ASE Distributed Transaction Manager
- Additional Connectivity Language Modules
- Open Client (CT-Library, DB-Library)
- Agent Management Console
- Microsoft Cluster Server Resource Type for ASE
- Interactive SQL
- Sybase Central
- SySAM Plug-in
- ASE ADO.NET Data Provider
- ASE OLE DB Provider by Sybase
- ASE ODBC Driver by Sybase
- ASE plug-in
- QPTune
- jConnect™ for JDBC
- jConnect 6.0.5 for JDBC
- jConnect 7.0 for JDBC
- SySAM License Utilities
- DBISQL
- SDC Administration Utility

Installing the Client

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 9. Minimum System requirements for PC-Client on Windows

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

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 15.5.
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**.
 - a) If you specify a destination that contains a previous version of the server, the **Choose Update Installation** panel displays. In this panel, you choose the features to update.
9. If you are updating your pre-existing server, you will not see this step. Choose 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 *Adaptive Server Enterprise Configuration Guide for Windows*.

Installing the Client in Silent Mode

After running the installer in GUI mode and recording the responses into a response file, edit the file to customize the responses.

To install in silent mode, run:

```
setupConsole.exe -f responseFileName -i silent
-DAGREE_TO_SYBASE_LICENSE=true
```

where *responseFileName* is the absolute path of the file name containing the installation options you chose.

Note: You must agree to the Sybase License Agreement when installing in silent mode. You can either:

- Include the option

```
-DAGREE_TO_SYBASE_LICENSE=true
```

 in the command line argument.
- Edit the response file to include the property

```
AGREE_TO_SYBASE_LICENSE=true
```

Except for the absence of the GUI screens, all actions of InstallAnywhere 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.

Warning! Sybase recommends that you use the `setupConsole.exe` executable, which runs in the foreground when you are running a silent installation. The normal `setup.exe` executable runs in the background, giving you the impression that the installation has terminated, and results in additional installation attempts using the silent installation. Multiple installations at the same time may corrupt the Windows Registry and lead to a failure to restart the operating system.

Getting Started After Installing

After installing Adaptive Server, Sybase Central, and the Java Runtime Environment, test the installation and network connections. 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.

1. Select **Start > Programs > Sybase > Sybase Central 6.0.0**.
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.

Creating a Testing Environment

To create a testing environment, create a simulated cluster on a single node with all instances running on that node.

Sybase recommends that you configure each instance on a different node i.e. 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.

The default setting for the **runnable process search count** parameter is '3'. Sybase recommends using this default setting. This can be set using the **sp_configure** stored procedure.

Configuring libtcl.cfg for LDAP

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.

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.

Using a Directory Service

For more information, see the configuration guide for your platform.

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
```

Note: On Windows x64, the `.dll` file is called `libsybdldap64.dll`.

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**.
 1. Select the transport type
 2. Enter the host name
 3. Enter the port number
 - f) Click **OK** twice to exit the **dsedit** utility.

Adding a Server to the Interfaces File

To access a Sybase server through the `sql.ini` file, each server must be included in the `sql.ini` file on the client computer.

Note: You can use Windows Directory Replication to replicate `sql.ini` 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*.

1. From Windows, select **Start > Programs > Sybase Connectivity > Open Client Directory Service Editor**.
2. Click **OK** on the first window to open the Interfaces Driver window.
3. Select **Server Object** 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 Protocol Network Address, click **Add**.
7. 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.

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

Uninstalling PC-Client

Actions performed by the Installer are removed during an uninstallation process. Files, or registry entries created after installation are not removed and must be manually removed when the installation process has finished.

Follow the instructions to uninstall some or all of the components.

To uninstall PC-Client from a Windows machine, run:

```
%SYBASE%\sybuninstall\PCClient\uninstall.exe
```

You can also remove PC-Client using the **Add/Remove** Programs from the **Control Panel**.

Upgrading the Server

To upgrade from Adaptive Server 15.5 Noncluster Edition to the 15.5 Cluster Edition, use the upgrade procedures described in this manual. Look here to learn about upgrade compatibilities.

You can upgrade to the Cluster Edition from:

- 12.5 through 12.5.4 ESD #8
- 15.0 through 15.5

You can upgrade to this version of the Cluster Edition from:

- 15.0.1 Cluster Edition to 15.0.1 Cluster Edition ESD #4
- 15.0.3 Cluster edition
- 12.5.4 including ESDs and 15.0.x versions including 15.5 nonclustered server

Note: Once you have upgraded to the 15.5 Cluster Edition you cannot downgrade back to an nonclustered 15.x or earlier server. You can downgrade back to 15.0.1 Cluster Edition through 15.0.1 Cluster Edition ESD #4, Cluster Edition 15.0.3 and Cluster Edition 15.5.

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

Use Adaptive Server plug-in or the **sybcluster** utility to perform many of the pre-upgrade and upgrade steps for you. You can also upgrade the server manually.

- To use Adaptive Server plug-in – See the Upgrading the Server With ASE Plug-in topic.
- To use **sybcluster** to upgrade the server – See the Upgrading the Server with **sybcluster** topic .
- To upgrade the server manually – follow the instructions in this and succeeding sections.

See the Migrating topic for information about migration.

Note: To upgrade from a shared installation mode to a private installation mode in Adaptive Server 15.5 Cluster Edition refer to the Release Bulletin for special upgrade instructions.

Preupgrade Tasks

You must have SA privileges to perform an upgrade.

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

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.

1. Install the Cluster Edition onto your system in a new location.
2. Check system and upgrade requirements. See the Checking System and Upgrade Requirements topic.
3. Check the location of the `RUN_server` file. See the Checking Runserver File Location topic.
4. 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.
5. Stored procedure text in the *syscomments* table is required for upgrade. See the Procedure Text is Required for Upgrade topic.
6. Resolve reserved words using quoted identifiers. See the Reserved Words topic.
7. Verify users are logged off.
8. Check for database integrity.
9. Back up the databases.
10. Dump the transaction log.
11. Ensure that master is the default database for the “sa” user.
12. Prepare the database and devices for upgrade. Follow these steps.
 - a) Disable auditing
 - b) Disable Job Scheduler by ensuring the “enable Job Scheduler” is off.
 - c) Archive auditing data and truncate auditing tables.
 - d) Disable disk mirroring.

Note: Cluster Edition version 15.5 does not support disk mirroring.

- e) Verify that your `$SYBASE` environment variable points to the location of the new Adaptive Server software files you just unloaded.
13. If you are upgrading to a private installation see *Upgrading to a Private Installation* on page 78.
14. If the upgrade is from 12.5.4 or 15.0.2 or 15.5 non-clustered server, and archive database access has been applied in the lower version server, disable the related functionality before upgrade.
15. If you are upgrading from a 12.5.3 installation with cache bindings on **sybssystemdb** drop the cache bindings for *sybssystemdb* that are bound to user-defined cache before running **preupgrade**.

If you do not do this prior to running **preupgrade** you will receive this error:

```
Current process... infected with 11
```

. If you do receive this error, rerun the **preupgrade** utility after you have dropped the cache bindings.

16. 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`

17. If the Java in database feature is enabled, create **sybpcidb** database. The **sybpcidb** database stores configuration information for the Pluggable Component Interface (PCI) and all Pluggable Component Adapter (PCA) components. This database is used by **installpci** script. For example:

```
1> disk init
2> name = "sybpcidb_dev",
3> physname = "/dev/raw/raw20",
4> size = '24MB'
5> go
1> create database sybpcidb on sybpcidb_dev = 24
2> go
```

Note: The **installpci** script must only be run on the first node and not on any other nodes in the cluster. In addition only create the PCI device on the first node. A single instance of the PCI configuration is shared between the nodes of a Cluster.

Statistics in System Tables

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

Execute **update index statistics** on the following system tables:

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

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 *Clusters users Guide*, for a complete list of catalogs that are affected.

Checking System and Upgrade Requirements

Learn how to check system and upgrade requirements here.

1. Verify that the computer on which you plan to upgrade the Sybase products meets the requirements outlined in the System Requirements topic.
2. Determine whether the version of your server is ungradable to the Cluster Edition.
3. Verify that you unloaded Adaptive Server in a **different** directory than your previous Adaptive Server installation. However, if you overwrite the previous 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 text.

Reserved Words

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

Transact-SQL disallows 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.

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.

1. If other identifiers are reserved words, use:

- **sp_rename** to change the name of the object, before or after the upgrade.
- Quoted identifiers.
- Brackets around the identifier. For example:

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

2. 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

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.

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

Running a Reserved Word Check

Run the reserved word check on the old Adaptive Server.

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 old Adaptive Server and execute **sp_checkreswords** on all the databases. For example:

```
use sybsystemprocs
go
sp_checkreswords
go
```

4. Correct any errors from the reserved word check.

Upgrading to a Private Installation

Upgrade from a shared installation to a private installation.

In Adaptive Server Cluster Edition version 15.5, you can configure your cluster as a "shared" or "private" installation. See Chapter 1, "An Overview of the Cluster Edition," in the *Clusters Users Guide*.

Upgrading a symmetric multiprocessing (SMP) version of Adaptive Server to a private installation of the Cluster Edition must be manually performed. First, upgrade your Adaptive Server to a Cluster Edition shared installation, then switch to a private installation using the steps below. Since private installations were introduced in version 15.0.3, cluster instances created with earlier versions of Adaptive Server Cluster Edition automatically continue as a shared installation. See the installation guide for your platform for instructions on how to upgrade your SMP Adaptive Server to a shared-disk cluster.

Note: When deciding on the installation location for Adaptive Server Cluster Edition 15.5, choose the location where you will install the private installation for this node. This location need not be accessible from other nodes participating in the cluster.

Changing from shared installation mode to private installation mode

1. Make sure each participating node in the cluster has its own \$SYBASE environment variable. Typically, the private installation is performed on a local file system, as there is no longer a need for other nodes participating in the cluster to have access to this installation.
2. Install the Cluster Edition on each node participating in the cluster. You can set up one of the nodes to use the existing installation if it satisfies the needs, otherwise you can discard it at the end of this process. You may need to discard an existing installation if, for example, it is on an NFS file system being used by nodes and you want to install on a local file system. See the installation guide for your platform for instructions on how to install a Cluster Edition for every node.
3. On each node, shut down the Cluster and UAF Agent.
4. On one of the nodes in the cluster, set up your environment by sourcing SYBASE.csh or SYBASE.sh, depending on the shell you are using. If the SYBASE installation location differs from the shared installation and the private installation, set up the environment from the shared installation area.
5. Extract the current cluster quorum configuration from the quorum device. For example:

```
% $SYBASE/$SYBASE_ASE/bin/qrmutil
--extract-config=mycluster_shared.cfg
--quorum-dev=/dev/raw/raw50m41
```

Executing command 'extract cluster configuration', argument

```
'mycluster_shared.cfg'...
```

```
Extracted input file 'mycluster_shared.cfg'
```

```
Command 'extract cluster configuration', argument  
'mycluster_shared.cfg' succeeded.
```

```
qrmutil execution completed.
```

6. Create a new cluster configuration file and update the required information:
 - a) Make a copy of the extracted configuration file, then edit the new file to change required configurations, such as: `cp mycluster_shared.cfg mycluster_private.cfg`
 - b) Edit the new configuration file; in the **[cluster]** section.

Change:

```
installation mode = shared
```

To:

```
installation mode = private
```

- c) In the **[instance]** section:
 1. Move the configuration file and interfaces entry from the **[cluster]** to the **[instance]** section
 2. If the SYBASE installation location has changed from shared to private, adjust the paths in the error log, config file, and interfaces path locations.
 3. If you have more than one instance in the configuration file, perform these actions for each instance. For example:

```
% cat mycluster_private.cfg
```

```
# All input files must begin with a comment
```

```
[cluster]
```

```
name = mycluster
```

```
max instances = 4
```

```
primary protocol = udp
```

```
secondary protocol = udp
```

```
master device = /dev/raw/raw1g2
```

```
traceflags =
```

```
additional run parameters =
```

```
installation mode = private
```

```
membership mode =
```

```
[management nodes]
```

```
hostname = nuno1
```

```
hostname = nuno2
```

```
[instance]
```

```
name = mycluster_instance1
```

```
id = 1
```

```
node = nuno1
```

```
primary address = nuno1
```

```
primary port start = 15100
```

```
secondary address = nuno1
```

```
secondary port start = 15181
errorlog = /mysybase1/mycluster_inst1.log
config file = /mysybase1/mycluster.cfg
interfaces path = /mysybase1
traceflags =
additional run parameters =

[instance]
name = mycluster_instance2
id = 2
node = nuno2
primary address = nuno2
primary port start = 15100
secondary address = nuno2
secondary port start = 15181
errorlog = /mysybase2/mycluster_inst2.log
config file = /mysybase2/mycluster.cfg
interfaces path = /mysybase2
traceflags =
additional run parameters =
```

7. Load the updated cluster configuration file into the cluster quorum device. For example:

```
% $SYBASE/$SYBASE_ASE/bin/qrmutil
--quorum-dev=/dev/raw/raw50m41
--cluster-input=mycluster_private.cfg
Loaded a new quorum configuration.
qrmutil execution completed.
```

8. If you have:

- More than one node in the cluster or have changed the SYBASE installation location – copy the Adaptive Server configuration file—typically named **servername.cfg**—and interfaces file from the original shared installation cluster into the corresponding interfaces path and configuration file locations for each instance in the private installation cluster. You can find the locations in the **[instance]** section of the updated cluster configuration file.
- Only one node/instance in the cluster and are not changing the SYBASE installation directory – update the UAF agent configuration information. The agent plug-in XML file is located at \$SYBASE/UAF-2_5/nodes/[*machine_name*]/plugins/[*cluster_name*]/agent-plugin.xml.

In it, replace:

```
<set-property property="ase.installation.mode"
value="shared" />
```

With:

```
<set-property property="ase.installation.mode"
value="private" />
```

9. Restart UAF Agent on each node in the cluster using the private installation directories. From the \$SYBASE directory, enter **UAF-2_5/bin/uafstartup.sh**
10. If you have more than one node in the cluster or have changed the SYBASE installation location, deploy UAF Agent plug-in for each node:

- a) Start **sybcluster**. For example, enter:

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

- b) Deploy the plug-in on each node individually. For example, enter:

```
deploy plugin agent "blade1"
deploy plugin agent "blade2"
deploy plugin agent "blade3"
```

See “The **sybcluster** Utility” in the *Cluster Users Guide* for complete syntax and usage information for **sybcluster** and the Adaptive Server plug-in.

11. You have now upgraded your shared installation to a private installation. You can start the cluster using **start cluster** command, or the individual nodes using **start instance <instance name>** command.

While issuing either command, you may see **sybcluster** display the following, including an error message that the cluster ID on the quorum device does not match the master device:

```
INFO - Starting the cluster mycluster instance
mycluster_instance1 using the operating system command:
/mysybase1/ASE-15_0/bin/dataserver --quorum_dev= /dev/raw/
raw50m41 --instance_name= mycluster_instance1
INFO - 01:00:00000:00000:2009/06/07 23:09:35.46 kernel Quorum
UUID: 00000000-0000-0000-0000-000000000000
INFO - 01:00:00000:00000:2009/06/07 23:09:35.46 kernel Master
UUID: 91f058aa-bc57-408d-854d-4c240883a6c9
INFO - 01:00:00000:00000:2009/06/07 23:09:35.46 kernel Unique
cluster id on quorum device does not match master device. You may
be using the wrong master device. If this is the correct master,
pass 'create-cluster-id' on the command line to pair the devices.
```

When this occurs, reissue the same command, but add **create-cluster-id** as suggested in the message to pair the master device, and start the node manually. For example, issue:

```
/mysybase1/ASE-15_0/bin/dataserver --quorum_dev= /dev/raw/raw50m41
--instance_name= mycluster_instance1--create-cluster-id
```

The command should now run without an error message.

12. To add new nodes to this cluster, you can now use either the Sybase Central Adaptive Server plug-in or the **sybcluster** tool to do so. See *Clusters Users Guide*.

You have now upgraded your shared installation to a private installation. To add new nodes to this cluster, you can now use either the Sybase Central Adaptive Server plug-in or the **sybcluster** tool to do so. See *Clusters Users Guide*.

Manually Upgrade Overview

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. Pick up **preupgrade** utility from the Cluster Edition release location. Run it against the old servers to update the underlying schema so that their structures are correct for the new server.

Note: Each new version of Adaptive Server contains different features that introduce new parameters, commands, reserved words, and so on. For this reason, use **preupgrade** to prepare the old server for an upgrade. However, if you are upgrading from 15.0.1 Cluster Edition or 15.0.3 Cluster Edition to the 15.5 Cluster Edition, you can skip this step.

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

5. Perform upgrade tasks.
6. Perform post-upgrade tasks.

Manually Upgrading the Old Server

Start the Cluster Edition with only one instance until the upgrade is complete. Follow these steps to manually upgrade your old server.

1. Back up all old databases.
2. Start the earlier version of Adaptive Server:
 - a) Move to the old \$SYBASE directory

```
cd $SYBASE
```
 - b) Source SYBASE.sh (Bourne shell) or SYBASE.csh (C shell).

```
source SYBASE.csh
```
 - c) 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 pre-upgrade test on the old server using the **preupgrade** utility, located at `$SYBASE/$SYBASE_ASE/upgrade`, where `$SYBASE` and `$SYBASE_ASE` are the values for the Cluster Edition. **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:

```
$SYBASE/$SYBASE_ASE/upgrade/preupgrade -Sserver_name -
Username -Ppassword-I $OLD_SYBASE/interfaces
```

Where:

- `$SYBASE_ASE` – is the cluster edition of Adaptive Server
- If -U option is ignored, then -P option is for the system administrator's password.

- b) Correct all errors from the output of the pre-upgrade 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 old Adaptive Server:

- 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:

```
1> use sybsystemprocs
2> go
1> sp_checkreswords
2> go
```

- d) Correct any errors the reserved word check reveals.

7. Shut down the old Adaptive Server.

8. Copy the old Adaptive Server configuration file `mycluster.cfg` from the old `$SYBASE` directory to the new `$SYBASE` directory.

9. If you are upgrading from a 15.0.1 Cluster Edition or a 15.0.3 Cluster Edition to a 15.5 Cluster Edition server, skip this step. Complete this step if you are upgrading from a nonclustered server. Create the cluster input file. For example `mycluster.inp`:

#all input files must begin with a comment

```
[cluster]
name = mycluster
max instances = 2
master device = /dev/raw/raw101
config file = /sybase/server_name.cfg
interfaces path = /sybase/
```

```
traceflags =
primary protocol = udp
secondary protocol = udp

[management nodes]
hostname = blade1
hostname = blade2

[instance]
id = 1
name = server_name
node = blade1
primary address = blade1
primary port start = 38456
secondary address = blade1
secondary port start = 38466
errorlog = /sybase/install/server_name.log
config file = /sybase/server_name.cfg
interfaces path = /sybase/
traceflags =
additional run parameters =

[instance]
id = 2
name = server_name_ns2
node = blade2
primary address = blade2
primary port start = 38556
secondary address = blade2
secondary port start = 38566
errorlog = /sybase/install/server_name_ns2.log
config file = /sybase/server_name.cfg
interfaces path = /sybase/
traceflags =
additional run parameters =
```

For an example of what this input file must contain, see the [Creating a Cluster Input File](#) topic.

Note: The first instance's *server_name* should be the name of the old server from which you are upgrading.

10. If you are upgrading from a 15.0.1 Cluster Edition or a or 15.0.3 Cluster Edition to a 15.5 Cluster Edition server, skip this step. Complete this step if you are upgrading from a nonclustered server. Add an additional entry to the interfaces file for each of the instances in your cluster input file (described in Step 9). See the [Configuring the Interfaces File](#) topic for more information.
11. Complete this step if you are upgrading from a nonclustered server. If you are upgrading from a 15.0.1 Cluster Edition or a or 15.0.3 Cluster Edition to a 15.5 Cluster Edition server, use the original quorum device and cluster input file, and specify **--buildquorum=force** to rebuild the quorum and to override the existing one. 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=server_name \
--cluster-input=mycluster.inp \
--quorum-dev=/dev/raw/raw102 \
--buildquorum \
-M$SYBASE
```

Note: The *server_name* you indicate with the **--instance** parameter must be the name of the server from which you are upgrading, 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 *Clusters Users Guide*.

13. Run the **upgrade** utility, where **instance_name** is the first instance in your cluster that has the same name as the server from which you are upgrading:

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

14. If you are upgrading from a 15.0.1 Cluster Edition or 15.0.3 Cluster Edition to a 15.5 Cluster Edition server, skip this step. 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 the Setting Up Local System Temporary Databases topic for more 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=server_name \
--quorum-dev=/dev/raw/raw102 \
-M$SYBASE
```

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

```
1> use sybsystemprocs
2> go
1> sp_checkreswords
2> 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:

- a) Add this argument to the `run_server` file: **--quorum-dev=<path to the quorum device>**
- b) Remove these options, as the information is now stored in the quorum device.
 - **-c**
 - **-i**
 - **-e**

See the [Creating the Runserver Files](#) topic for more information.

20. Start each of instance in the cluster:

```
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
```

Preparing the Database and Devices for the Upgrade

If you ran the **preupgrade**, it scanned the system catalogs and calculates how much additional space you need to perform the upgrade.

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

1. 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).

2. If the database has insufficient space for upgrade, increase the size with **alter database**. Use **preupgrade** to determine the database size.

If you have a *sybsystemdb* and the database is large enough, you do not need to make any changes.

3. 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.

4. Verify that the *sybsystemprocs* database is large enough.

For an upgrade, the recommended minimum size for *sybsystemprocs* is the larger of 140MB, 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. 140MB accommodates additional internal data structures, but does not account for the possibility of a larger number of user-defined system procedures.

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

5. To enable Java in database feature you must enable PCI and configure **sybpcidb** database.

For **sqlupgrade[res]**, the **enable java** setting in the old server is detected. If it is set at 1, then **preupgrade** asks you to create a **sybpcidb** device and database before you run **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 increase the *master*, *tempdb*, and *model* databases follow these steps.

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
```

Upgrading the Server Using ASE Plug-in

To login to the ASE plug-in see instructions starting with the Setting the Environment and Starting Unified Agents topic.

To upgrade:

- The **dataserver** for the pre-existing server must be on the same system as the cluster UAF agent is running.
- You must be able to reach the install files for the Cluster Edition and the pre-existing server from the system on which the UAF agent is running.

This means that the old server must run on the same system as the one that the UAF agent for the Cluster Edition runs on, and on which the newly upgraded cluster will reside.

Note: If you cannot connect to a server or view it in the tree view in the left pane, confirm that the Connection profile—in the toolbar under Tools | Connect—contains the port number associated with the server in the `interfaces` file (`sql.ini` for PC-Client).

After you install the Cluster Edition, start the Adaptive Server plug-in and make sure you can view the old server icon in the left pane. Connect to the old server the same way you would connect to a Cluster Server. See the Connecting to the Cluster topic for more information.

Upgrading the Old Server with ASE Plug-in

Learn how to upgrade your server using ASE plug-in.

Set the environment variables and start the Unified Agent. See the Setting the Environment and Starting the Unified Agents topic for how to do this.

1. Back up your old server before performing an upgrade.
2. Right-click on the old-server icon in the tree view in the left pane and select Upgrade. A wizard opens.
3. Click **Next** to proceed. To change the server name, enter the new name in the field for Cluster name. The default name is the same name as that of the old server.
4. Choose the maximum number of instances the cluster server will have. Click **Next**. The Upgrade wizard validates the cluster name and checks for available hosts.
5. Select the correct host from the list of available hosts. If no agents for that host are shown, check the protocol settings using the Discovery Settings button.
6. Enter the administrator's user name and password for the selected UAF agent. Click **Next**.
7. Enter the full release directory path for the old server, verifying that the Adaptive Server and OCS directories for the old server are correct. Click **Next**.
8. Enter the full quorum device path and name, such as `/dev/raw/raw21`. Click **Next**.

Note: Sybase recommends that you use raw devices for the quorum; although it is possible to run the quorum on a file device, this is not supported.

9. Check to make sure the paths to the new cluster release area are correct.
You have the option to enter parameters and traceflags. See the *Cluster Users Guide* for more information on parameters and traceflags. Click **Next**.
10. Accept the default port number of the range of interconnect ports; do not confuse these with instance query ports. See Chapter 19, “Setting Up Communications Across Networks” in the *Users Guide to Clusters* for more information on the latter.
If you choose a different port number, starting with the port number you enter, the wizard validates the range of ports. If none are available, the wizard provides a list of all unavailable ports in the range.
 - a) Check the box if a secondary network is available.
 - b) Click **Next**.
11. Enter a name for the instance in the cluster created in the upgrade process. After the cluster is created, you can add instances up to the number of instances you specified as the maximum for the cluster. You can specify start parameters and traceflags for each instance at the instance configuration page.

To specify start parameters and traceflags for the cluster, enter those at the cluster configuration page. Accept the default path for the instance **log** file or enter a new one. Click **Next**.

Note: See the Adding an Instance to a Cluster topic for how to add more instances after upgrade is complete.

12. Specify the local system temporary database name and size, or accept the default. If you plan to create all local system temporary databases including those for instances you plan to add later, the device must be large enough to accommodate all local system temporary databases. Click **Next**.

At the local system *tempdb* page, you can create a device on which to put the local system temporary databases by clicking the Create Device button and entering the path, name and size in the Create Device dialog.

You can create local system temporary database only on shared disks. See Chapter 6, “Using temporary databases,” in *Users Guide to Clusters* for more information about using local disks for temporary data.

13. Check the box on the summary page if you wish to start the server once it is successfully upgraded. Click **Next**.

The **preupgrade** check begins; this may take a while. If the server is not ready for upgrade, **preupgrade** returns errors, which you must fix according to the parameters specified by the wizard’s **preupgrade** check before you can perform a successful upgrade.

Note: The wizard lets you change things such as your configuration options and database size, and correct most errors that require fixes without closing it.

See the *Users Guide to Clusters* for information on cluster-specific configuration parameters and stored procedures. Also see the Adaptive Server version of the installation guide for your pre-existing Adaptive Server for information on fixing reported errors. These manuals are available at *The Product Manual Download site*

14. After you fix the reported errors, click Retry to run **preupgrade** again. If you see no errors, the Upgrade wizard upgrades the old server to a Cluster Edition server, and when finished, you see the icon for the new server instead of the icon for the old server in the tree view in the left pane.

Upgrading the Server Using sybcluster

Learn about the upgrade process using the **sybcluster** utility, and preupgrade tasks to complete before upgrading your old server. You can also perform a manual upgrade to the Cluster Edition. See the Manually Upgrading topic for how to do this.

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 to Clusters* for detailed information about **sybcluster** and the UAF.

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 15.5 installation directories are located on the same machine. This is the upgrade machine.
- Review the Creating and Starting the Cluster topic and the Worksheet for creating a cluster for a list of the information you need for the upgrade.

Complete the tasks before you upgrade your server.

1. See the Verifying the Server for Upgrade topic to make sure that your server is ready for upgrade.
2. See the Upgrading Your Old Server With **sybcluster** topic for how to upgrade the server to Adaptive Server Cluster Edition version 15.5.
3. Perform the postupgrade tasks after the upgrade, whether you upgrade the server manually or use **sybcluster**.

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.

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*]:** Enter the existing Sybase installation directory for server *server_name*.
 - b) **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 secondary 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 configuration 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*.**

Note: Ensure that the instance query port number is same as the query port number for the old nonclustered server, otherwise the upgrade may fail.

- v) **Enter the local system temporary database device name.** You can create local system temporary database on shared disks only. See Chapter 6, “Using temporary databases” in *Cluster Edition User Guide* for more information about using local disks for temporary data.
- w) **Enter the local system temporary database device path.**
- x) **Enter local system temporary database device size (MB).** [100]
- y) **Enter the local system temporary database name.** For example:
[*cluster_name_tdb_1*]
- z) **Enter the local system temporary 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. You also can use it as input to validate the server for numerous subsequent upgrades, if needed. For example,

```
upgrade server server_name file /data/myserver_ce.xml checkonly
```

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

Upgrading the Old Server Using an Input File

After successfully completing the verification procedure you can upgrade 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 the **Creating and Starting the Cluster** topic.

3. **sybcluster** displays these prompts:

- a) 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 privileges on the target server.
- b) 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 Old Server Interactively

Learn how to upgrade the old 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:

- a) Enter the name of the cluster. [*server_name*]
- b) Enter the existing Sybase installation directory for server *server_name* using 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*.
- f) 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 secondary network. [Y]
- n) Enter the port number from which this range will be applied. [15100]
- o) Enter the SYBASE home directory. [Your default location.]
- p) Enter the environment shell script path. [Your default location.]
- q) Enter the ASE home directory. [Your default location.]

- r) Enter the interfaces directory. [Your default location.]
- s) Enter path to the dataserver configuration file. [Your default location.]
- t) 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.

- u) Enter the interface file query port number for *instance_name*.
- v) Enter the local system temporary database device name.
- w) Enter the local system temporary database device path.
- x) Enter local system temporary database device size (MB).[100]
- y) Enter the local system temporary database name, for example, [*cluster_name_tdb_1*]
- z) Enter the local system temporary database size (MB).[100]
- aa) Would you like to save this configuration information in a file? [Y]
- ab) Enter the name of the file to save the cluster creation information.
- ac) Create the cluster now? [Y]

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

Postupgrade Tasks

Complete these steps after upgrading to the Cluster Edition.

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. If you are upgrading from 15.x version, restart the server to make the statistics available.

Warning! Running **installpcidb** will drop and re-create the **sybpcidb** database and all of the tables and data. The configuration returns to a default state. If changes have been made previously to the Pluggable Component Interface (PCI) configuration, the same changes will need to be made again after running **installpcidb**.

1. After upgrading from Adaptive Server Cluster Edition version 15.0.1 to 15.5 Cluster Edition, you must a one-time procedure for each of your user-created nodes to update the jar and xml files. See the Updating .jar and .xml Files topic for instructions on how to do this:

Updating .jar and .xml Files on page 95

2. Restore functionality in Adaptive Server.
3. Re-enable auditing.

Updating .jar and .xml Files

Learn about updating .jar and .xml files before running preupgrade.

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 the cluster.

1. Start the Unified Agent. See the Setting the environment and starting the Unified Agents topic.
2. Run the **sybcluster** utility. See the Creating the cluster server using sybcluster topic.
3. Use the **sybcluster deploy plugin** command to configure the Unified Agent for your cluster.

If your .xml files are from the, \$LOWVERSION_SYBASE, then refer to lower version install location. If they are from the higher version number, \$HIGHVERSION_SYBASE refer to higher version install location.

4. Compare the versions, then add the lower version agent-plugin.xml unique properties to the higher version agent-plugin.xml properties.

```
$LOWVERSION_SYBASE/UAF-2_5/nodes/<Host Name>/plugins/<Cluster OR
InstanceName>/agent-plugin.xml with $HIGHVERSION_SYBASE/UAF-2_5/
nodes/<Host Name>/plugins/<Cluster OR InstanceName>/agent-
plugin.xml,
```

For example, copy following property from lower version xml file to higher version xml file.

```
<set-property property="ase.database.devices"
value="[1, sysprocsdev, /work/river/ase1501ce/data/
sybsystemprocs.dat, 150, 135; 0, master, /work/river/ase1501ce/data/
master.dat, 80, 60; 2, systemdbdev, /work/river/ase1501ce/data/
sybsystemdb.dat, 12, 12; 3, lstdb1, /work/river/ase1501ce/data/
lstdb1.dat, 50, 50]" />
```

5. Repeat this procedure for all of the nodes in the cluster.
6. Shut down the UAF agent.

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:

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 are insufficient for the data copy phase of the upgrade process. In this case, an error is generated letting you know there is insufficient space in the *system* segment for the upgrade. The upgrade process stops and waits for space to be provided. 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'.

Upgrading Java in the Database

The Pluggable Component Interface (PCI) configuration values are used by all of the nodes in a cluster. To share files between nodes, using working directories, you need shared directory locations in the file system that are visible to all nodes using the same path.

Each node has its own separate and local temporary directory usually /tmp on Unix, tmp on Windows.

1. The **installpci** script must only be run on the first node, not on any other nodes in the cluster. A single instance of the PCI configuration is shared between the nodes of a Cluster.
2. The **sybpcidb** database must be created before running the **installpcidb** script. You can choose the location and size of this database. As long as the database is named **sybpcidb** the **installpcidb** script will work correctly. For example:

```
1> disk init
2> name = "sybpcidb_dev",
3> physname = "/dev/raw/raw20",
4> size = '24MB'
5> go
1> create database sybpcidb on sybpcidb_dev = 24
2> go
```

3. Once the **sybpcidb** database has been created, run the **installpcidb** script to create and populate the tables that hold the (PCI) and Pluggable Component Adapter (PCA) configuration properties. Stored procedures and tables are created in the **sybpcidb** database. Using **isql**, run the **installpcidb** script that is included with the release. Save the output to an operating system file. For example:

```
isql -Usa -P<sa_password> -S<server_name>
-i$SYBASE/$SYBASE_ASE/scripts/installpcidb -o<output_file>
```

4. Enable the following two configuration parameters in order to enable the Java in database functionality.

```
1> sp_configure 'enable pci', 1
2> go
1> sp_configure 'enable java', 1
2> go
```

5. You may need to increase the '**max memory**' in order to enable these parameters. After they are enabled, restart the server for changes to take affect. You can configure the

maximum size of the PCI Bridge memory pool through the '**pci memory size**' configuration parameter. For more information see the *Java in Adaptive Server Enterprise for 15.5* guide.

Migrating

For complete instructions about using **dump** and **load** to migrate to the Cluster Edition, see the Adaptive Server 15.5 installation guide for your platform.

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.

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 Version 15.x or Earlier Version Using BCP

You can use the **bcp** utility to migrate to the Cluster Edition from any Adaptive Server version.

See the Adaptive Server 15.5 *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.

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 a High Availability-Configured Server

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

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

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:

a) For an asymmetric configuration, issue this command on the secondary companion:

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

b) 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.

Downgrading

Learn how to downgrade from Adaptive Server 15.5 Cluster Edition to an earlier Adaptive Server Cluster Edition release. Downgrading to a 15.x, 12.5.x or earlier release is not supported. Instructions to downgrade to a 15.0.1 Cluster Edition ESD version or a 15.0.3 Cluster Edition version include:

- Basic downgrade steps using new system procedure **sp_downgrade**.
- Post-downgrade tasks.
- Information regarding downgrading if you used 15.5 Cluster Edition features.
- Returning to Adaptive Server version 15.5 Cluster Edition.

Basic Downgrade Steps

Learn the basic steps for downgrading from Adaptive Server 15.5 Cluster Edition to a 15.0.1 Cluster Edition, 15.0.1 Cluster Edition ESDs, and 15.0.3 Cluster Edition. Additional actions are required if you used 15.5 Cluster Edition features. Follow the steps in the Additional Steps If You Used 15.5 Cluster Edition Features Before Downgrading topic.

Adaptive Server 15.5 Cluster Edition provides a 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 required to run this procedure. For detailed information on this 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*.

1. If you have used Pluggable Component Interface (PCI) functionality, and are downgrading to a version of the 15.0.1 Cluster Edition or a 15.0.1 Cluster Edition ESD, drop the **sybpcidb** and restart the old server binary.
2. If you are downgrading to version 15.0.3 Cluster Edition, rerun **installpcidb** and restart the old server binary.
3. Start the 15.5 Cluster Edition server in single user mode, with only one instance running, using **dataserver -m** option to guarantee that no other user can access Adaptive Server while you are performing the downgrade. See *Utility Guide* for more details on starting the server in single user mode.
4. Make sure Adaptive Server 15.5 Cluster Edition is prepared for downgrade by executing **sp_downgrade** from the master database:

```
sp_downgrade 'prepare', @toversion=<'version'>
, @override = 1
```

The value of “version” may be written as “15.0.1.5”, “15015”, “15.0.3.5” or “15035”. However, the version entered should be the version you want to downgrade to.

sp_downgrade 'prepare' validates the readiness of Adaptive Server 15.5 Cluster Edition 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 the Additional Steps If You Used 15.0.3 Cluster Edition Features Before Downgrading topic for more information on correcting the issues reported by **sp_downgrade 'prepare'**.

5. Repeat step 3 until **sp_downgrade 'prepare'** reports no errors. Your system is now ready to downgrade.
6. Begin the downgrade process by executing **sp_downgrade** from the master database:

```
sp_downgrade 'downgrade', @toversion = '15015', @override = 1
```

The version number should be "15.0.1.5", "15015", "15.0.3.5" or "15035". Once the above command completes successfully there can be no more activity on the 15.5 Cluster Edition server. You must immediately shutdown the 15.5 Cluster Edition server. **sp_downgrade 'downgrade'** may report that the password for sa and other login accounts has been reset and that the account is locked. In this case, the new password shown by **sp_downgrade** is used as the password for sa after the downgrade. Any other affected login accounts should be unlocked after the downgrade is complete.

7. Run - checkpoint and shutdown the old server.
8. Copy the RUN_SERVER file to the release area of the version to which you are downgrading. Modify the RUN_SERVER file to use the dataserver binary from the release area of the version to which you are downgrading. Remove the **-m** option when you modify the RUN_SERVER file or ensure the server is **not** running in **single user** mode.
9. Extract the quorum device using:

```
qrmutil --quorum-dev=<dev> --extract-config=quorum.out
```

 - a) If you are downgrading to a Cluster Edition version 15.0.1 or a Cluster Edition 15.0.1 ESD, comment these two lines in **quorum.out**:

```
#installation mode=shared  
#membership mode=
```
10. Rebuild the new quorum device with the old quorum device backup you created by adding this to the **dataserver** parameters:

```
--buildquorum=force --cluster-input=quorum.out
```
11. Restart the old server using the modified RUN_SERVER file.

Note: If a downgraded server is booted using the 15.5 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. When downgrading to a 15.0.1 Cluster Edition server, user passwords may be reset and printed on the console. Save the output of **sp_downgrade 'downgrade', <version>** into a file so that the passwords are not lost. The old passwords do not work. Otherwise, you must restart the downgraded server with **-psa** to regenerate the sa password.

12. Depending upon whether you are starting from a fresh installation or from a previously upgraded 15.x installation, run previously saved `restore_msgs.ebf` or `instmsgs.ebf` from 15.0.1 Cluster Edition or 15.0.3 Cluster Edition if none were saved, before running **installmaster** for the Cluster Edition.
13. Restart all instances of the cluster.

Postdowngrade Tasks

After you perform the basic downgrade steps, some changes from the 15.5 Cluster Edition upgrade still persist.

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 certify that Component Integration Services running on earlier versions of Adaptive Server can connect to later versions. You may encounter errors if an earlier 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 15.0.1 Cluster Edition, 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.

1. 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 are not removed. Attempts to execute such stored procedures against an older binary will have unpredictable results.
2. During upgrade from a previous 15.0.1 Cluster Edition or 15.0.3 Cluster Edition version to Adaptive Server 15.5 Cluster Edition you should have created a `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>
```
3. 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.
4. After returning to a pre-15.5 Cluster Edition binary:
 - a) If during the upgrade process you ran the **'update all statistics'** on *syslogins* then you must **delete statistics** for *syslogins* and re-create them.

- b) **spt_values** are dropped and re-created when you run the 15.0.1 Cluster Edition or 15.0.3 Cluster Edition **installmaster**, so any new types are eliminated from this table.
- 5. New configuration parameters have been added in 15.5 Cluster Edition. If a downgraded server is booted using the 15.5 Cluster Edition 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 15.0.1 Cluster Edition or 15.0.3 Cluster Edition version of **installmaster** removes configuration parameters that belong to 15.5 Cluster Edition, by deleting *sysconfigures* rows for configuration parameters that do not exist in *sysurconfigs*. After running **installmaster** the error messages will no longer appear when you start the server.
- 6. Keywords **decrypt_default**, **xmltable**, **path**, were added in Adaptive Server 15.5 Cluster Edition, making it impossible to create identifiers using these names. You must change applications if you used these names. See the Additional steps if you used 15.5 Cluster Edition features before downgrading topic for more information.

Additional Steps If You Used 15.5 Cluster Edition Features Before Downgrading

If you are rolling back after having used any of the 15.5 Cluster Edition features, additional steps may be necessary. Some steps must be completed before downgrading to 15.0.1 Cluster Edition, 15.0.1 cluster Edition ESDs, and 15.0.3 Cluster Edition, other steps are performed immediately after downgrading. Read the documentation on the features below that may require manual changes.

To correct errors related to:

In general, no additional steps are required when you are returning to an Adaptive Server version in which the feature was already available. However, more information is available on the Sybase Web site at the product manual page where you can search the Adaptive Server Enterprise documentation set for more information on your feature. <http://sybooks.sybase.com/nav/base.do>

1. If you used Real-Time Messaging features – 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*.
2. If you used New Sort Orders – If a new nocase sortorder for Chinese or Japanese character sets is configured as Adaptive server's default sortorder then, before downgrading to a 15.0.1 Cluster Edition or to a Cluster Edition ESD 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 Chapter 9, "Configuring Character Sets, Sort Orders, and Languages," in the *System Administration Guide* for details on how to change server's

default sort order. If **sp_downgrade** is called when the new nocase sort order is in use, you see the following error message, and the downgrade process is aborted:

```
Cannot downgrade to '%1!' server, which does not support
server's current default sortorder.
```

3. If you used Native XML – The XML Service feature of Adaptive Server includes the new **xmltable** function. You will get an error if you create views or stored procedures using the **xmltable** function in 15.5 Cluster Edition then return to 15.0.1 Cluster Edition.
4. If you used instead of Triggers – instead of triggers are objects stored in the system catalogs. Remove these objects before downgrading. When the 15.0.1 Cluster Edition or 15.0.3 Cluster Edition server is started, any unremoved instead of triggers remain in the system catalogs but do not execute.
5. If you used SQL User-Defined Functions – SQL user-defined functions are 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 15.0.1 Cluster Edition or 15.0.3 Cluster Edition version will result in misleading error messages.
6. If you installed the 15.5 automatic database expansion procedures using **installdbextend** 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 15.0.1 Cluster Edition or 15.0.3 Cluster Edition. To clear all auto-expansion thresholds that might exist on one or more segments before downgrading, run:

```
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 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.

Note: 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.

7. If you have Replication Issues with downgrade – When downgrading a server that has replication enabled on databases that contain encrypted data, perform one of the following before you start the downgrade procedure:
 - a) 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.
 - b) 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')
```

Shut down 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 15.0.1 Cluster Edition, 15.0.1 Cluster Edition ESD, or 15.0.3 Cluster Edition, run **installjsdb** script from the lower version anytime you downgrade 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 sybmgmt  
go  
sp_js_wakeup "start_js",1  
go
```


I/O Fencing

Sybase cannot guarantee data integrity unless you enable I/O fencing. If you do not use I/O 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 I/O fencing.

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 I/O fencing to prevent the noncooperating instance from making data writes.

In the Cluster Edition, Sybase supports the SCSI-3 Persistent Group Reservation (PGR) feature of SCSI-3 devices to provide I/O 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 I/O 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 I/O Fencing

Configuring I/O fencing affects all devices with I/O fencing capability system-wide.

1. Each instance in the cluster must run on a separate node.
2. 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.

- 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.
-
3. I/O 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
```
 4. 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 I/O fencing. You can also run the **qrmutil** utility:

```
qrmutil -Q<quorum path> --fence-capable=<device path>
```

Setting Up I/O Fencing

Learn how to set up I/O Fencing.

Sybase recommends that your I/O fencing use persistent DSF on HP-UX 11.31.

Note: HPIA I/O fencing does not support the HP VM.

1. Run **qrmutil** to confirm that a disk is I/O fencing capable. See Chapter 14, “System Changes,” in the Cluster User Guide for information about running **qrmutil**.
2. Before you enable I/O fencing, make sure you have read and write permissions for all database and quorum devices. Use the UNIX **chown**, **chgrp**, and **chmod** commands to correct the permissions if needed.

Troubleshoot SySAM

If SySAM errors are not addressed in a timely fashion, Adaptive Server may cease to function. Learn about the SySAM grace period [here](#).

There may be more recent information published in the *SySAM Asset Management Users Guide* online at *SySAM FAQ*. For the most up-to-date troubleshooting information in this Installation Guide, go to the Sybase Product Manuals site *Product Manuals site*. If your Sybase product cannot find a required license, it continues to operate in grace mode, if a grace period is available. Determine the license status of your product using the output from **sp_lmconfig** command. The **Status** column shows **graced** for any license that is operating within a grace period.

Typically, you have a 30-day grace period to resolve any SySAM-related issues. When a license is awarded under grace, the Sybase product 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. Your Sybase product disables the functionality corresponding to the license if the issue is not fixed by the expiration date. Adaptive Server shuts down an expired license if ASE_CORE.

The Sybase product 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 the Enabling and changing e-mail notifications topic for more information. The warnings in the Adaptive Server error log and e-mail messages are repeated while the license is under grace mode.

Where to Look For SySAM Related Errors

All SySAM-related errors are printed in your Sybase product 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).
```

See the Error Log Locations topic for where to find error logs.

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 your Sybase product 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

Learn methods for troubleshooting SySAM here.

Table 10. 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 drop-down list and complete the installation. After the installation is complete, use sp_lmconfig to change the license type. See the <i>Reference Manual:Procedures</i> 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 <i>Version</i> 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 <i>Sybase Software Asset Management Users Guide</i>. 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 your support was not paid, you are not authorized to install an update. See Appendix B of <i>Sybase Software Asset Management Users Guide</i> for explanation of how SySAM uses date-based versioning. See “Updating your Sybase licenses” in Chapter 3, “Getting and Using your Licenses,” in the <i>Sybase Software Asset Management Users Guide</i> for how to update your license.

Description	What to do
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.
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.

Description	What to do
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 <i>feature_name</i> 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 <i>Sybase Software Asset Management Users Guide</i>. 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 <i>Sybase Software Asset Management Users Guide</i>.

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. - Run sp_lmconfig to see the configured values. - 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 <i>feature_name</i>" 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 <i>Property Name / Property Value</i> table. The edition value is displayed in the <i>PE</i> row and active and standby value is displayed in the <i>AS</i> row. Compare the values of <i>PE</i> and <i>AS</i> for the Adaptive Server with the same values recorded in the <i>VENDOR_STRING</i> and <i>ISSUER</i> fields of the optional feature license. They must match. Some features like <i>ASE_HA</i> 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.

Calling Sybase Technical Support

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

- The Adaptive Server error log.
 - `$SYBASE/ASE-15_0/install/<servername>.log`
- If Adaptive Server does not start, the server's properties file:
 - `$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/bin` 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.

Troubleshoot the Server

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 the Error Log Locations topic for more information.

The table, Troubleshooting guidelines for UNIX platforms, 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. 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.
After upgrading Adaptive Server, you cannot use srvbuild .	After you begin upgrading a server, you may be unable to use the same srvbuild session for other tasks. Exit and restart srvbuild .

Problem	Possible cause and solution
The installation program cannot connect to the upgraded Adaptive Server.	After you begin upgrading a server, you may be unable to use the same srvbuild session for other tasks. Exit and restart srvbuild .
The installation program detects reserved word conflicts.	See the Stopping Adaptive Server After a Failure topic.
The upgrade fails.	See the Recovering From a Failed Upgrade topic.

Error Log Locations

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

Logs for Installation Utilities

The information in the error logs helps to determine the reason and possible solution for an error message. Find the default error log locations for the installation, upgrade, and setup utilities here.

Table 12. Error log locations for installation utilities

Utility	Error log location
Install Anywhere	\$SYBASE/log/ASE_Suite.log
srvbuildres	\$SYBASE/\$SYBASE_ASE/init/logs/srvbuildMMDD.NNN Where: <i>MM</i> is the month. <i>DD</i> is the date. <i>NNN</i> is a three-digit number identifying the srvbuild session.
sqlupgrades	\$SYBASE/\$SYBASE_ASE/init/logs/sqlupgradesMMDD.NNN \$SYBASE/\$SYBASE_ASE/upgrade/upgrade.NNN (a temporary file created by the upgrade process)

Utility	Error log location
Install Anywhere	\$SYBASE/log/ASE_Suite.log.log

Logs For Sybase Servers

The information in the error logs helps to determine the reason and possible solution for an error message. Find the default error log locations for each Sybase server here.

Table 13. 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.log
Monitor Server	\$SYBASE/\$SYBASE_ASE/install/servername.log
XP Server	\$SYBASE/\$SYBASE_ASE/install/servername.log

Solutions to Common Installation Problems

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

Table 14. Solutions to common installation problems

Problem	Possible cause and solution
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: <pre>% 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 %</pre> The installation utilities will now use approximately 25 percent less window space.

Problem	Possible cause and solution
Cannot eject the CD or DVD 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. - 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 <i>remote</i> machine is not set correctly to display the installer to your <i>local</i> machine: The DISPLAY environment variable is not set correctly. To correct the problem, enter the following command at the UNIX prompt of the <i>remote</i> machine, where <i>host_name</i> is the name of the machine on which you want the installer to appear (that is, on your <i>local</i> machine):For C shell: <pre>setenv DISPLAY host_name:0.0</pre> For Bourne shell: <pre>DISPLAY=host_name:0.0; export DISPLAY</pre>
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: <pre>Xlib: connection to "host_name" refused by server Xlib: Client is not authorized to connect to Server xhost: unable to open display "host_name"</pre> To correct the problem: - Enter the following command at the UNIX prompt of your <i>local</i> machine, where <i>remote_machine</i> is the machine on which you are running the installer: <pre>xhost +remote_machine</pre> - Restart the installer.
Address already in use	This message from srvbuild means you entered a port number that is already in use: <pre>kernel: ninit: bind, Address already in use</pre> 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.

Problem	Possible cause and solution
Adaptive Server fails to start	The shared memory of the operating system may not be set high enough. To correct the problem, see the Installing the Server topic, 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 the debug mode: 1. Set the environment variable LAX_DEBUG to true before running the installer.
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: <pre>Msg 11018, Level 16, State 1: Procedure "xp_cmdshell", Line 2: XP Server must be up for ESP to execute. (return status = -6)</pre> Verify that there is an XP Server entry in the Adaptive Server <i>sys.servers</i> 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 <i>sys.servers</i> table. Verify that the XP server exists in the interfaces file or LDAP server. Use sp_addserver to add an entry to the <i>sys.servers</i> table.
Troubleshooting Re-source File Installations	If you encounter problems during the build, configuration, or upgrade process while using the srvbuild[res], sqlloc[res], or sqlup-grade[res], it may be that these utilities did not allow enough time for Adaptive Server to shut down properly. 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: <pre>% setenv SYBSHUTWAIT 120</pre>

Problem	Possible cause and solution
Configuration Failed	Check the location of the system shared libraries. The shared libraries <code>libXt.a</code> and <code>libX11.a</code> are normally stored in <code>/usr/lpp/X11/lib/R5</code>. The <code>libXm.a</code> library is normally located in either <code>/usr/lpp/X11/Motif1.2/lib</code> or <code>/usr/lpp/X11/R5/Motif1.2</code>. If these shared libraries are located in directories different from the above, set the <code>LIBPATH</code> environment variable to indicate these other directories. The shared libraries <code>libXt.so</code> and <code>libX11.so</code> are normally stored in <code>/usr/openwin/lib</code>. The shared library <code>libsocket.so</code> is normally located in <code>/usr/lib</code>. If the shared libraries on your operating system are located in directories other than those mentioned in the preceding paragraph, set the environment variable <code>LD_LIBRARY_PATH</code> to indicate the new locations.

If the Configuration Fails

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 the Server After a Failure

Learn how to stop the 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
```

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, **preupgrade** 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.
3. After you resolve any problems, shut down Adaptive Server and use **sybcluster** or manual upgrade to complete the upgrade session.

Recovering From a Failed Installation or After the Files are Created

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 the Troubleshoot the Server topic.

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 the Troubleshoot the Server topic.
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.

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 the Troubleshoot the Server topic.

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. Having set this trace flag, user "sa" can now use the offline database and make the necessary changes to correct the upgrade failure.
2. 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.

3. 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.

4. 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/upgrade
```

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.

5. 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 **sybcluster** or restart the manual upgrade.
4. Select Upgrade if the cause of failure is unknown. Adaptive Server, and proceed with the upgrade.

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 the Finding Compiled Object Errors Before Production.

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.

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.

Use **dbcc upgrade_object** to find the following errors and potential problem areas that may require manual changes to achieve the correct behavior:

After reviewing the errors and potential problem areas, and fixing those that need to be changed, use **dbcc upgrade_object** to upgrade compiled objects manually instead of waiting for the server to upgrade the objects automatically. See the Using **dbcc upgrade_object** topic for more information.

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

Learn about 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.

1. For example, suppose you have the following statement:

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

2. 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")
```

3. Compare the output of the two queries.

- a) 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.
- b) 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*.

4. 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.
 - a) 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

The following topics cover syntax, parameters, permissions, Increasing the log segment, error reporting and examples for using the **dbcc upgrade_object**.

Syntax for Using dbcc upgrade_object

Learn the syntax for using **dbcc upgrade_object** here.

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

Parameters

Learn about parameters here.

- **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.

Permissions

Learn about permissions here.

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.

1. 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

Learn about the error reporting function here.

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.

Examples

Find examples for using **dbcc upgrade_object**.

1. **Example 1** – Upgrades all compiled objects in the active database:

```
dbcc upgrade_object
```
2. **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')
```
3. **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")
```
4. **Example 4** – Upgrades all stored procedures named **list_proc** in the *listdb* database.

```
dbcc upgrade_object(listdb, list_proc)
```
5. **Example 5** – Upgrades the stored procedure **list_proc**, which is owned by the login “jkarrik”.

```
dbcc upgrade_object(listdb,  
"listdb.jkarrik.list_proc")
```
6. **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")
```

Using Database Dumps in Upgrades

To upgrade using dump and load you can load 12.5 database dumps and transaction logs and upgrade the databases. However, there are some issues of which you should be aware:

1. 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.
2. 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.

For information on using **sp_checksource**, see the *Reference Manual: Procedures*.

1. After loading a database dump, run **sp_checksource** to verify the existence of the source text for all compiled objects in the database.
2. 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.

Manually Setting Up and Managing the Cluster

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 the Enabling **sybcluster** and the Plug-in After a Manual Configuration topic.

Before You Begin

Learn what to do before manually setting up and configuring the cluster.

Before you configure the cluster,

- Set the Sybase environment variables
- Create the raw devices

Setting the Environment

Set your Sybase environment. From the Sybase release directory, source the SYBASE . sh or SYBASE . csh file.

From the \$SYBASE directory where the environment variable scripts can be located, source the environment variable. For example:

```
. SYBASE . sh
```

or,

```
source 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 the 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).

Note: Local system temporary database must use shared disks.

- Quorum disk device (at least 20 MB)

- All other database devices

Configuring the Cluster Manually

Sybase strongly recommends that the total number of engines in all instances does not exceed the number of CPUs.

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 *sybssystemprocs* and create the *sybssystemprocs* database.
6. Install the system stored procedures.
7. Set up a local system temporary database for each instance.
8. Start the cluster.

Create 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, quorum disk devices, and other required configuration information. Choose any name for the cluster input 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 the [Reconfiguring the Cluster](#) topic 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
configuration file = common path to all server configuration files
primary protocol = udp | tcp | other
secondary protocol = udp | tcp | other
installation mode = shared | private
configuration 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
interfaces path = interfaces file path
config file = path to server configuration file for this instance
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
interfaces path = interfaces file path
configuration file = path to server configuration file for this instance
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.
- **installation mode** – shared or private installation mode
- **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.

In a private installation where all configuration files share the same path name, this is the common path.

- **interfaces path** – path to the interfaces file. Omit this parameter if you are using an LDAP environment. 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.

In a private installation where path names to individual server configuration files are not the same, this is the path to the current server configuration 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. Omit this parameter if you are using LDAP.
- **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
secondary 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
secondary 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 Clusters *Users Guide*.

Configuring the Interfaces File

If you use an interfaces file, it must include entries for the cluster server and all instances.

The syntax for the interfaces file is:

```
instance_name:
  master network_protocol machine_name port_number
  query network_protocol 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 the Create the Cluster Input File topic specifies raw device `raw11` as the quorum disk. This example also uses `raw11` for the quorum disk.

For example:

```
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= 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
```

Initializing and Creating *sybsystemprocs*

Learn how to initialize and create *sybsystemprocs* here.

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 = "150MB"
```

2. Create the *sybsystemprocs* database:

```
create database sybsystemprocs on sysprocsdev = 150
```

Installing the System Stored Procedures

Learn how to install system stored procedures here.

Run InstallAnywhere 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
```

Creating 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.

Create a runserver file 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.

Setting Up Local System and Temporary Databases

In a shared-disk cluster, each instance must have 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. You can create local system temporary database on shared disks only. See Chapter 6, “Using temporary databases” in *Cluster Edition User Guide* for more information about using local disks for temporary data.

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.

Configuring the Cluster for Automatic Takeover

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 the Starting the Cluster topic for information about manually restarting the cluster.

Environments that have I/O fencing enabled, **automatic cluster takeover** is guaranteed to be a safe operation. Environments that do not have I/O fencing, a malfunction of the algorithm may introduce data corruption. Therefore, this configuration parameter is present to disable the algorithm if it malfunctions. 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:

Manually Setting Up and Managing the Cluster

- Normally, after a graceful shutdown.
- After a system failure. See the Starting the Cluster After a System Failure topic.

Note: Sybase recommends that you do not alter the runserver files used for normal cluster startup.

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

Learn how to start the server after a system failure.

If the cluster fails 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 the automatic cluster takeover parameter to the default, which is 1. The instance attempting to connect to the cluster restarts the cluster, acting as the cluster coordinator and recovering the databases. See the Configuring the Cluster for Automatic Cluster Takeover topic for information about configuring the cluster to automatically restart.
2. If you have not enabled **automatic cluster takeover**, you must restart 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 normally.

Postconfiguration Tasks

Learn what you should do after you have manually configured the server.

To determine if the instances you configured are running, log in to the cluster and enter:

```
sp_cluster show
```

Then Verify that you can connect to the instances using these steps:

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 shuts down all the instances associated with the cluster.

Follow these steps to shutdown a cluster.

1. Log in to any instance. For example:

```
isql -Usa -P -Sase2 -ISYBASE/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 also shut down an instance from another instance in the cluster.

1. Log in to any instance. For example:

```
isql -Usa -P -Smycluster -ISYBASE/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 start first and that includes the **dataserver** with the **cluster-input** option. This option tells Adaptive Server to write the new configuration information to the quorum device.

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 *Clusters 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 the Plug-in After a Manual Configuration

To use **sybcluster** or the Adaptive Server plug-in to manage the cluster after a manual configuration, start a Unified Agent on each node in the cluster, and then deploy the connection agent to each node.

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 *Clusters Users Guide* for complete syntax and usage information for **sybcluster** and ASE plug-in.

A sybcluster Sample Session Configuration Values

Look here for an example of how to configure a typical shared-disk cluster using **sybcluster**.

This example assumes a primary and secondary network. If you do not specify a secondary 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 the Creating the Cluster Server Using **sybcluster** topic.

Table 15. sybcluster prompts and configuration values entered for sample session

Parameter	Value
Cluster name	mycluster
Number of instances	
Number of agents	
Cluster node ID	1
Configuration type	
Is the cluster to be configured using a private \$SYBASE installation?	N
Quorum Device	
Full path and name for the quorum device	/dev/rhdisk11
Page Size	
The page size in kilobytes	2kB
Master device	
Full path and name for the master device	/dev/rhdisk12
Size of master device [MB]	30MB
Size of master database [MB]	13MB
PCI device (optional)	

A sybcluster Sample Session Configuration Values

Parameter	Value
Full path to the PCI database device	/dev/raw/raw20
Size of the PCI database device [MB]	24MB
Size of the PCI database [MB]	24MB
Sybase system procedure device	
Full path and name for sybssystemprocs device	/dev/rhdisk13
Size of sybssystemprocs device (MB)	160MB
Size of sybssystemprocs database (MB)	152MB
System database device	
Full path and name of system database device	/dev/rhdisk14
Size of system database device (MB)	6MB
Size of system database (MB)	6MB

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

Parameter	Value
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.) You must use a shared disk for local system temporary database devices.	mycluster1_tempdb
Path to local system temporary database device	/dev/rhdisk15
Local system temporary database device size (MB)	40MB
Local system temporary database name	mycluster_tdb_1
Local system temporary database size (MB)	40MB

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.) You must use a shared disk for local system temporary database devices.	mycluster2_tempdb
Path to local system temporary database device	/dev/rhdisk16
Local system temporary database device size (MB)	40MB

A sybcluster Sample Session Configuration Values

Parameter	Value
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.) You must use a shared disk for local system temporary database devices.	mycluster3_tempdb
Path to local system temporary database device	/dev/rhdisk17
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
Veritas Cluster Server support (optional – appears only if VCS is supported on your system)	

Parameter	Value
Integrate the Cluster Edition server with VCS? [Y]	N
Check whether each cluster device has IO fencing capability? [Y]	

Sample sybcluster Session

Look here for a **sybcluster** sample session.

This sample session assumes:

- A shared configuration
- An interfaces file for server lookup
- Support for Java in the database

```

sybcluster -U uafadmin -P -F
hpcblade2:9009,hpcblade1:9009,hpcblade3:9009,hpcblade4:9009
> create cluster
Enter the name of the cluster: mycluster
Cluster mycluster - Enter the maximum number of instances: [ 4 ]
How many agents will participate in this cluster: [ 4 ] 4
Verifying the supplied agent specifications...
1) hpcblade1.sybase.com 9009 2.5.0 Linux
2) hpcblade2.sybase.com 9009 2.5.0 Linux
3) hpcblade3.sybase.com 9009 2.5.0 Linux
4) hpcblade4.sybase.com 9009 2.5.0 Linux
Enter the number representing the cluster node 1: [ 4 ] 1
2) hpcblade2.sybase.com 9009 2.5.0 Linux
3) hpcblade3.sybase.com 9009 2.5.0 Linux
4) hpcblade4.sybase.com 9009 2.5.0 Linux
Enter the number representing the cluster node 2: [ 4 ] 2
3) hpcblade3.sybase.com 9009 2.5.0 Linux
4) hpcblade4.sybase.com 9009 2.5.0 Linux
Enter the number representing the cluster node 3: [ 4 ] 3
4) hpcblade4.sybase.com 9009 2.5.0 Linux
Enter the number representing the cluster node 4: [ 4 ] 4
Will this cluster be configured using private SYBASE
installations? (Y/N) : [ N ]
----- 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: /hpcblade_cfs/q/
pd16218942/d3.dbs
Enter any traceflags:
----- Page Size -----
Enter the page size in kilobytes: [ 2 ] 8
----- 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.
```

A sybcluster Sample Session Configuration Values

```
Enter the full path to the master device: /hpcblade_cfs/q/
pd16218942/d4.dbs
Enter the size the Master Device (MB): [ 120 ] 500
Enter the size the Master Database (MB): [ 52 ] 100
----- Sybase System Procedure Device -----
Sybase system procedures (sybsystemprocs) are stored on a device.
Enter the System Procedure Device path: /hpcblade_cfs/q/
pd16218942/d5.dbs
Enter System Procedure Device size (MB): [ 152 ] 200
Enter the System Procedure Database size (MB): [ 152 ] 200
----- System Database Device -----
The system database (sybsystemdb) stores information about
distributed transactions.
Enter the System Database Device path: /hpcblade_cfs/q/pd16218942/
d6.dbs
Enter the System Database Device size (MB): [ 24 ] 100
Enter the System Database size (MB): [ 24 ] 100
----- PCI Device -----
Pluggable Component Interface (PCI) provides support for Java in
database by loading off-the-shelf JVMs from any vendor. If you
want to use JVM, create a device for it.
Enable PCI in Adaptive Server (Y/N): [ N ] y
Enter the full path to the PCI device: /hpcblade_cfs/q/pd16218942/
pci.dbs
Enter the size the PCI Device (MB): [ 96 ]
Enter the size the PCI Database (MB): [ 96 ]
-----
Does this cluster have a secondary network: [ Y ] n
Enter the port number from which this range will be applied:
[ 15100 ] 17005
-----
Enter the SYBASE home directory: [ /remote/quasr5/adong/aries/
release/lamce_s1 ]
Enter the environment shell script path: [ /remote/quasr5/adong/
aries/release/lamce_s1/SYBASE.sh ]
Enter the ASE home directory: [ /remote/quasr5/adong/aries/
release/lamce_s1/ASE-15_0 ]
Enter path to the dataserver configuration file: [ /remote/
quasr5/adong/aries/release/lamce_s1/mycluster.cfg ]
-----
You will now be asked for the instance information on a node by
node basis.
-- Cluster: mycluster - Node: hpcblade1.sybase.com - Agent:
hpcblade1.sybase.com:9009 --
Enter the name of the cluster instance: instance1
Enter the interface file query port number for instance instance1:
10665
Enter the primary protocol address for instance1:
[ hpcblade1.sybase.com ]
----- Local System Temporary Database -----
The Local System Temporary Database Device contains a database for
each instance in the cluster.
Enter the LST device name: LST
Enter the LST device path: /hpcblade_cfs/q/pd16218942/d7.dbs
Enter LST device size (MB): 200
Enter the LST database name: [ mycluster_tdb_1 ]
```

```

Enter the LST database size (MB): [ 200 ] 50
Do you want to add another instance to this node? (Y or N): [ N ]
-- Cluster: mycluster - Node: hpcblade2.sybase.com - Agent:
hpcblade2.sybase.com:9009 --
Enter the name of the cluster instance: instance2
Enter the interface file query port number for instance instance2:
15465
Enter the primary protocol address for instance2:
[ hpcblade2.sybase.com ]
----- Local System Temporary Database -----
The Local System Temporary Database Device contains a database for
each instance in the cluster.
Enter the LST device name: [ LST ]
Enter the LST database name: [ mycluster_tdb_2 ]
Enter the LST database size (MB): [ 150 ] 50
Do you want to add another instance to this node? (Y or N): [ N ]
-- Cluster: mycluster - Node: hpcblade3.sybase.com - Agent:
hpcblade3.sybase.com:9009 --
Enter the name of the cluster instance: instance3
Enter the interface file query port number for instance instance3:
16730
Enter the primary protocol address for instance3:
[ hpcblade3.sybase.com ]
----- Local System Temporary Database -----
The Local System Temporary Database Device contains a database for
each instance in the cluster.
Enter the LST device name: [ LST ]
Enter the LST database name: [ mycluster_tdb_3 ]
Enter the LST database size (MB): [ 100 ] 50
Do you want to add another instance to this node? (Y or N): [ N ]
-- Cluster: mycluster - Node: hpcblade4.sybase.com - Agent:
hpcblade4.sybase.com:9009 --
Enter the name of the cluster instance: instance4
Enter the interface file query port number for instance instance4:
15220
Enter the primary protocol address for instance4:
[ hpcblade4.sybase.com ]
----- Local System Temporary Database -----
The Local System Temporary Database Device contains a database for
each instance in the cluster.
Enter the LST device name: [ LST ]
Enter the LST database name: [ mycluster_tdb_4 ]
Enter the LST database size (MB): [ 50 ]
Would you like to save this configuration information in a file? [
Y ]
Enter the name of the file to save the cluster creation
information: [ /hpcblade_cfs/q/s16218942/mycluster.xml ]
-----
Create the cluster now? [ Y ]
-----

```

Configuring the 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
```


Upgrading Servers with Replicated Databases

After completing an upgrade, you cannot scan any part of the transaction log that existed prior to the upgrade, so you must follow the procedure outlined in this section if your server contains replicated primary databases (this includes replicated RSSD). This procedure helps to ensure that data from a replicated database has been successfully replicated to the replicate database.

Warning! You must have a valid **dump database** and a **dump transaction** before executing the procedures in the procedures for upgrading a replicated database.

The procedures described here do not upgrade Replication Server itself. For information on upgrading Replication Server, see the Replication Server documentation.

These steps precede the database upgrade procedure. Each step represents a topic in this section:

- Suspending transaction processing and replication activities
- Draining the transaction logs for primary databases
- Draining the RSSD transaction log
- Disabling the secondary truncation point

Determining If Existing Servers Contain Replicated Databases

Determine if existing servers contain replicated databases before upgrade.

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 displays the list of databases enabled for replication agent.

- a) If you use another application to replicate data, LTM or Mirror Activator, run the following command in each database including the master database:

```
1> dbcc gettrunc
2> go
```

If the command returns “1” for “*secondary trunc state*” in any database, replication is enabled in that database.

Suspending Transaction Processing and Replication Activities

You must suspend transaction processing and replication activities before upgrading a replicated database.

Make sure that no users create subscriptions for data in the database you are upgrading until the upgrade procedure is finished.

1. Before beginning 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.
2. Defer the upgrade process while the subscriptions are being created so that Replication Server does not attempt to access the database being upgraded.
3. Execute **rs_helproute** in each Replication Server System Database (RSSD) being upgraded.
4. 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 on to the next step which is to shut down all applications that use the databases you are upgrading.
5. Shut down all applications that use the databases you are upgrading.
6. Use the **admin who** command in Replication Server to find the existing Data Server Interface (DSI) connections to the data server being upgraded.
7. 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
```
8. Leave the DSI connections to the RSSD databases running.

Draining the Transaction Logs for Primary Databases

Ensure that the Replication Server on each primary database you are upgrading completely processes the pre-upgrade log.

Follow these steps to ensure the transaction log is completely 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>, "tracoon", "9201"
2> go
```

If you cannot access the console where Adaptive Server is running, 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 uses a different method to replicate data, ensure that data is replicated by consulting the Replication Server queues. To do this, run the following Replication Server command:

```
1> admin who, sqm
2> go
```

3. Use **isql** to update one row in a single replicated table in the primary database:

```
1> update table set column = column
2> where key = unique_value
3> go
```

The **update** command helps track whether all modifications to the replicated database have been sent to the Replication Server.

Note: 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 columns in the table.

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 earlier in this process:

```
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 above.
7. Repeat the update steps above 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:

- a) Rep Agent – log into the Adaptive Server, and stop the Rep Agent:

```
1> sp_stop_rep_agent database
2> go
```

- b) LTM or Mirror Activator – stop them.

After draining the transaction logs, disallow all other activity in the databases. If activity does occur, you must drain the logs.

9. Instruct Replication Server to reject incoming connections from Replication Agent, LTM or Mirror Activator by logging in to the Replication Server and suspending 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, ensure that the Replication Server processes all transactions in the RSSD transaction log before upgrading the databases.

To ensure that 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.

6. If this is a replicated RSSD, log in to the Replication Server of the RSSD and issue:

```
1> sysadmin hibernate_on, 'Replication_Server_name'
2> go
```

Disabling the Secondary Truncation Point

When upgrading a primary database, the Rep Agent, Log Transfer Agent or Mirror Activator 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 be shut down prior to the upgrade.

For each primary database and replicated RSSD, disable the secondary truncation point:

1. 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
```

2. 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 the first step.

3. Truncate the primary transaction log running the command:

```
1> dump tran primary_dbname with truncate_only
2> go
```

4. Stop Replication Server then zero out the Replication truncation point. In the RSSD for the Replication Server, run:

```
1> rs_zeroltm primary_servername, primary_dbname
2> go
```


Using Sybsystemprocs

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. This is because 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 140MB, 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. 140MB 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, two options exist for creating a new database with sufficient space for the upgrade. Learn about these options here.

- The first option is to drop the old database and device and create a new *sysprocsdev* device.
- The second option is to leave the old database and 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.

To increase the size of the **sybsystemprocs** database:

1. If you do not have a current backup of your old database, create one now.
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
```

- a) If there is space available on another device, expand *sybsystemprocs* to a second device, or initialize another device that is large enough. See the Increasing Device and Database Capacity for System Procedures topic for more information.
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. Look here for how to do this.

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.

1. Determine which device or devices you must remove.

Warning! Do **not** remove any device that is in use by a 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)
```


- **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 devices that are used by *sybssystemprocs* and other databases.

Make a note of the names of the devices to use in the following steps.

2. Drop *sybssystemprocs*:

```
1> use master
2> go
1> drop database sybssystemprocs
2> go
```

Note: In versions of Adaptive Server Enterprise earlier than 15.x, use **sysdevices** to determine which device has a low through high virtual page range that includes the **vstart** from step 2.

In version 15.x, select the *vdevno* from *sysusages* matching, the *dbid* retrieved in step 1.

3. Remove the device or devices using **sp_configure**:

```
1> sp_configure "allow updates", 1
2> go
1> delete sysdevices
    where name in ("devname1", "devname2", ...)
2> go
1> sp_configure "allow updates", 0
2> 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** that were returned.

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 *sybssystemprocs* 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 server installation directory. Enter:

```
isql -Usa -Ppassword -Sserver_name -i$SYBASE/ASE-15_0/scripts/  
installmaster
```

Obtaining Help and Additional Information

Use the Sybase Getting Started CD, Product Manuals site (SyBooks™ Online), and online help to learn more about this product release.

- The Getting Started CD (or download) – contains release bulletins and installation guides in PDF format, and may contain other documents or updated information.
- Product Manuals at <http://sybooks.sybase.com/> – is an online version of Sybase documentation that you can access using a standard Web browser. You can browse documents online, or download them as PDFs. In addition to product documentation, the Web site also has links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, Community Forums/News groups, and other resources.
- Online help in the product, if available.

To read or print PDF documents, you need Adobe Acrobat Reader, which is available as a free download from the *Adobe* Web site.

Note: A more recent release bulletin, with critical product or document information added after the product release, may be available from the Product Manuals Web site.

Index

A

- Accessing SPDC
 - Sybase product download
 - online Web portal 13
- auxiliary servers
 - configuring 55

C

- command line 29
- configuration steps 33
- configuring the cluster 50
- Create the cluster with the plug-in 42
- creating a cluster with sybcluster 50
- creating the cluster
 - starting the cluster 33

D

- databases 86
- devices 86
- disabling
 - secondary truncation point 157
- downgrade steps
 - PCI functionality 99
- downgrading
 - Cluster Edition 102
 - features 102
 - Job Scheduler 104
 - overview 99
- draining
 - primary databases 154
 - transaction logs 154
 - RSSD 156

E

- environment variables
 - setting 35
 - starting 35
 - unified agents 35

F

- failed upgrade
 - recovery 95

G

- GUI mode 27

H

- Host ID
 - how to get
 - obtaining 25

I

- I/O Fencing
 - data integrity 105
 - enabling 105
 - setting up 106
- increasing
 - default database sizes 87
 - device and database capacity
 - system procedures 160
- infiniband 20
- input file
 - create 134
- installing 27, 29, 30

L

- LDAP libraries
 - environment variables 69
 - location of 69
- logical clusters
 - adding 46

M

- manual configuration
 - enabling sybcluster
 - enabling the plug-in 144
 - master device
 - quorum device 138
- migrating 97
 - BCP 97
 - high availability 98

N

- network protocol 20
- new license server 16

O

- obtaining
 - license 13
- options for the server editions
 - optional packages 2

P

- PC_Client
 - separate products 9
- PC-Client
 - product description
 - directory layout 7
- planning the installation 11
- postdowngrade 101
- postupgrade
 - update statistics
 - installpcidb 94
- preinstallation tasks 23
- preparing 86
- preupgrade
 - Java in database 73
- private installation 34
- private installation modezzx 27
- private interconnect technology 20

Q

- quoted identifiers 77

R

- reconfiguring the cluster 143
- reserved word check 77
- reserved words
 - conflicts 77
- response file 29

S

- served license

- generating 14
- server
 - advantages to the cluster edition 1
 - installation options 1
 - overview 1
 - single-system presentation 1
 - workload management 1
- Server
 - component descriptions
 - directory layout 5
- shared installation 33
- shared memory parameters
 - adjusting 24
- shutting down unified agent 36
- silent mode 30
- simulated cluster on a single node 35
- Storage Foundation
 - creating a cluster 36
- style conventions
 - fonts 4
- sybsystemprocs database
 - increasing the size of 159
- system administrator password
 - setting 61
- system catalog changes
 - upgrade 75
- system requirements 20
 - HP 19
 - PC-Client 21

T

- technical support
 - calling 114
- troubleshoot
 - SySAM
 - grace period 107

U

- unattended mode 30
- uninstalling
 - server
 - registry entries 30, 71
 - the old server
 - existing server 31
- unserved license
 - generating 15
- updating .jar files 95

- updating .xml files 95
- upgrading 86
 - ASE Plug-in 88
 - input file 92
 - interactively 93
 - Java in the database 96
 - manual
 - overview 82
 - manually 82
 - sybcluster 90
- upgrading compatibility
 - versions 73

V

- verifying readiness for upgrade 90

W

- with the installer 27
- workflow
 - prerequisites
 - server installation 3

