SYBASE®

Users Guide to Clusters

# Adaptive Server® Enterprise Cluster Edition

15.0.1, ESD #4

DOCUMENT ID: DC00768-01-1501-04

LAST REVISED: December 2008



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

# Contents

# About This Book

**Audience**

This manual is for Sybase® system administrators and database owners who install and use the Adaptive Server® Enterprise Cluster Edition. It includes installation and configuration instructions and descriptions of the features available in the Cluster Edition.

**How to use this book**

This manual includes:

- Part 1: Configuring the Adaptive Server Enterprise version 15.01 Cluster Edition
  - Chapter 1, “An Overview of the Cluster Edition,” is an overview of shared disk clustered architecture.
  - Chapter 2, “Client Applications and Client/ Server Interaction,”discusses Client-Library™ application changes.
  - Chapter 3, “Using Security with a Clustered Environment,” describes how to configure security for the Cluster Edition.
  - Chapter 4, “Using Monitoring Tables in a Clustered Environment,” describes monitor tables used in the Cluster Edition.
  - Chapter 5, “Managing the Workload,” describes how to manage the workload and provide failover for applications that access the Cluster Edition.
  - Chapter 6, “Cluster Cache Configuration,” describes how to configure and use named data caches in a cluster environment.
  - Chapter 7, “Using Temporary Databases,” describes local and global temporary databases, how to create and manage them, and how to bind users or applications to a temporary database or group of temporary databases.
  - Chapter 8, “Running Job Scheduler in a Clustered Environment,” describes how to run the Job Scheduler in a shared disk cluster environment across all instances in the cluster.

- Chapter 9, “Additional Topics,” describes differences between the Cluster Edition and other failover scenarios, explains more about locks and thresholds, and describes the recovery process and its use of distributed checkpoints.
- Chapter 10, “Troubleshooting,” provides instructions for handling common errors encountered with the Cluster Edition.
- Chapter 11, “Administering Clusters with the Adaptive Server Plug-in,” describes how to use the Adaptive Server Plug-in to manage the Cluster Edition.
- Chapter 12, “Administering Clusters Using sybcluster,” describes how to use the sybcluster utility to manage the Cluster Edition.
- Chapter 13, “System Changes,” describes new and changed stored procedures, commands, configuration parameters, system tables, global tables, and functions.
- Chapter 14, “The sybcluster Utility,” is the reference section for the commands used by the sybcluster utility.

- Part 2: General Configuration Issues
  - Chapter 15, “Configuring the Operating System,” provides configuration information for the Cluster Edition.
  - Chapter 16, “Customizing Localization for the Cluster Edition,” describes localization support for the Cluster Edition.
  - Chapter 17, “Adding Optional Functionality to the Cluster Edition,” describes how to add additional functionality to the Cluster Edition.
  - Chapter 18, “Logging Error Messages and Events,” describes how to use error logging in the Cluster Edition.
  - Chapter 19, “Setting Up Communications Across the Network,” describes how to set up network communications for the Cluster Edition.

**Other sources of information**

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

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

- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

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

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

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

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

**Sybase EBFs and software maintenance**

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

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

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

**Sybase certifications on the Web**

Technical documentation at the Sybase Web site is updated frequently.

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

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

2 Click Certification Report.

3 In the Certification Report filter select a product, platform, and timeframe and then click Go.

4 Click a Certification Report title to display the report.

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

1 Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.

2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.

3 Select Search to display the availability and certification report for the selection.

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

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

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

2 Click MySybase and create a MySybase profile.

## Conventions

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

***Table 1: Font and syntax conventions for this manual***

| Element | Example |
|---|---|
| Command names, procedure names, utility names, and other keywords display in sans serif font. | select<br>sp_configure |
| Database names and datatypes are in sans serif font. | master database |
| File names, variables, and path names are in italics. | *sql.ini* file<br>*column_name*<br>*$SYBASE/ASE* directory |

| Element | Example |
|---|---|
| Variables—or words that stand for values that you fill in—when they are part of a query or statement, are in italic in Courier font. | `select column_name`<br>`from table_name`<br>`where search_conditions` |
| Type parentheses as part of the command. | `compute row_aggregate (column_name)` |
| Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates "is defined as". | `::=` |
| Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces. | `{cash, check, credit}` |
| Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets. | `[cash \| check \| credit]` |
| The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command. | `cash, check, credit` |
| The pipe or vertical bar ( \| ) means you may select only one of the options shown. | `cash \| check \| credit` |
| An ellipsis (...) means that you can *repeat* the last unit as many times as you like. | `buy thing = price [cash \| check \| credit]`<br>`[, thing = price [cash \| check \| credit]]...`<br>You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment. |

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

  sp_dropdevice [*device_name*]

  For a command with more options:

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

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

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

  ```
  select * from publishers
  ```

- Examples of output from the computer appear as follows:

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

(3 rows affected)
```

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

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

**Accessibility features**

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

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

---

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

---

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

**If you need help**

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

PART 1

# Configuring the Cluster Edition

This part describes the procedures for configuring the Adaptive Server 15.0. Cluster Edition ESD #4 to run in a clustered environment.

CHAPTER 1

# An Overview of the Cluster Edition



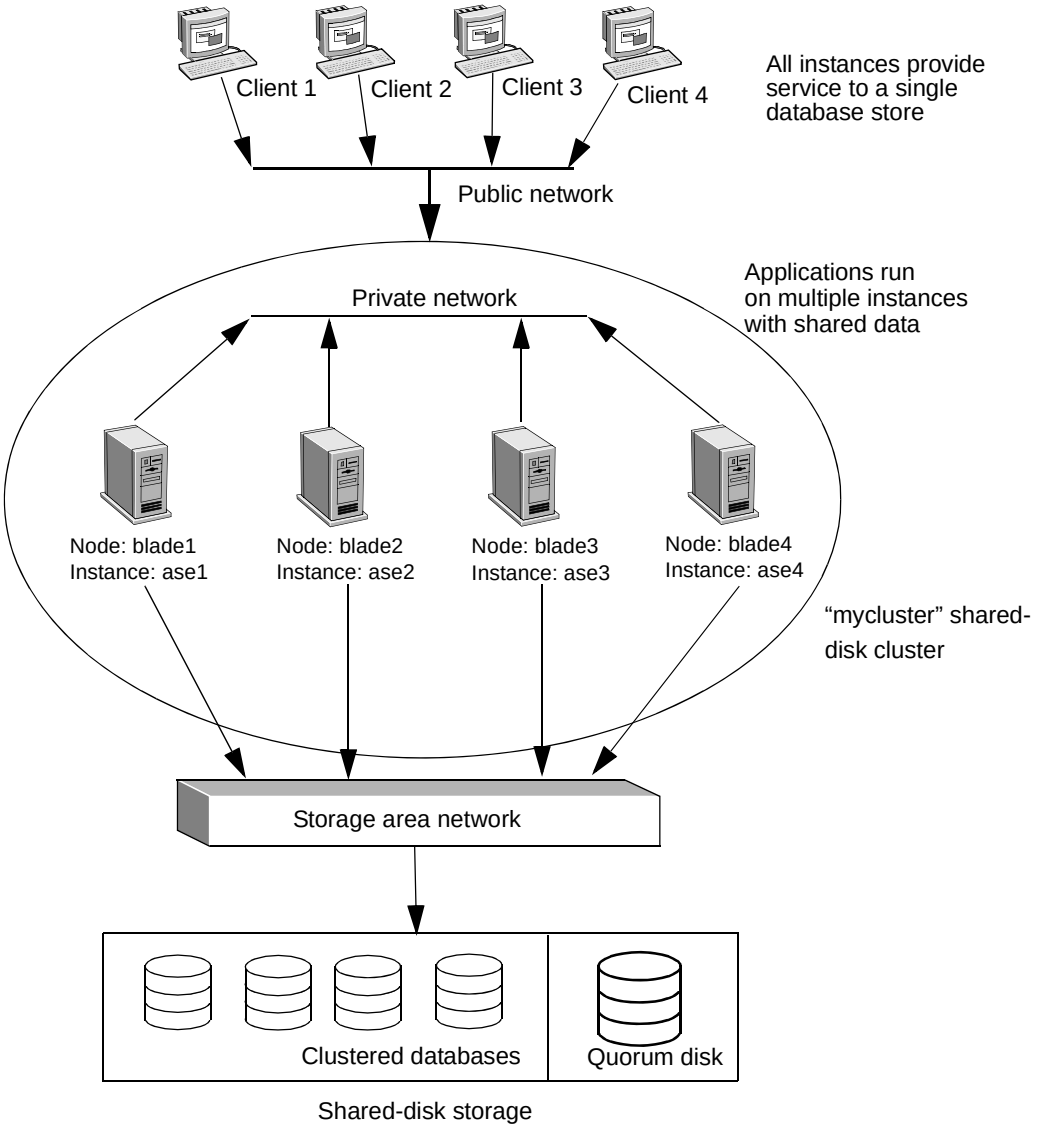
## What is the Cluster Edition?

The Cluster Edition allows you to configure multiple Adaptive Servers to run as a shared-disk cluster. Multiple machines connect to a shared set of disks and a high-speed private interconnection (for example, a gigabit ethernet), allowing Adaptive Server to "scale" using multiple physical and logical hosts.

In the cluster environment, each machine is referred to as a **node** and each Adaptive Server as an **instance**. Connected instances form a **cluster**, working together to manage a single set of databases on the shared disks. In each case, the instances present as a single system, with all data accessible from any **instance**. The Cluster Edition assigns SPIDs that are unique to the cluster, so the SPID identifies a single process across all instances in the cluster.

In the clustered system shown in Figure 1-1, clients connect to a shared-disk cluster named "mycluster," which includes the "ase1", "ase2", "ase3", and "ase4" instances running on machines "blade1", "blade2", "blade3", and "blade4", respectively. In this example, a single instance resides on each node.

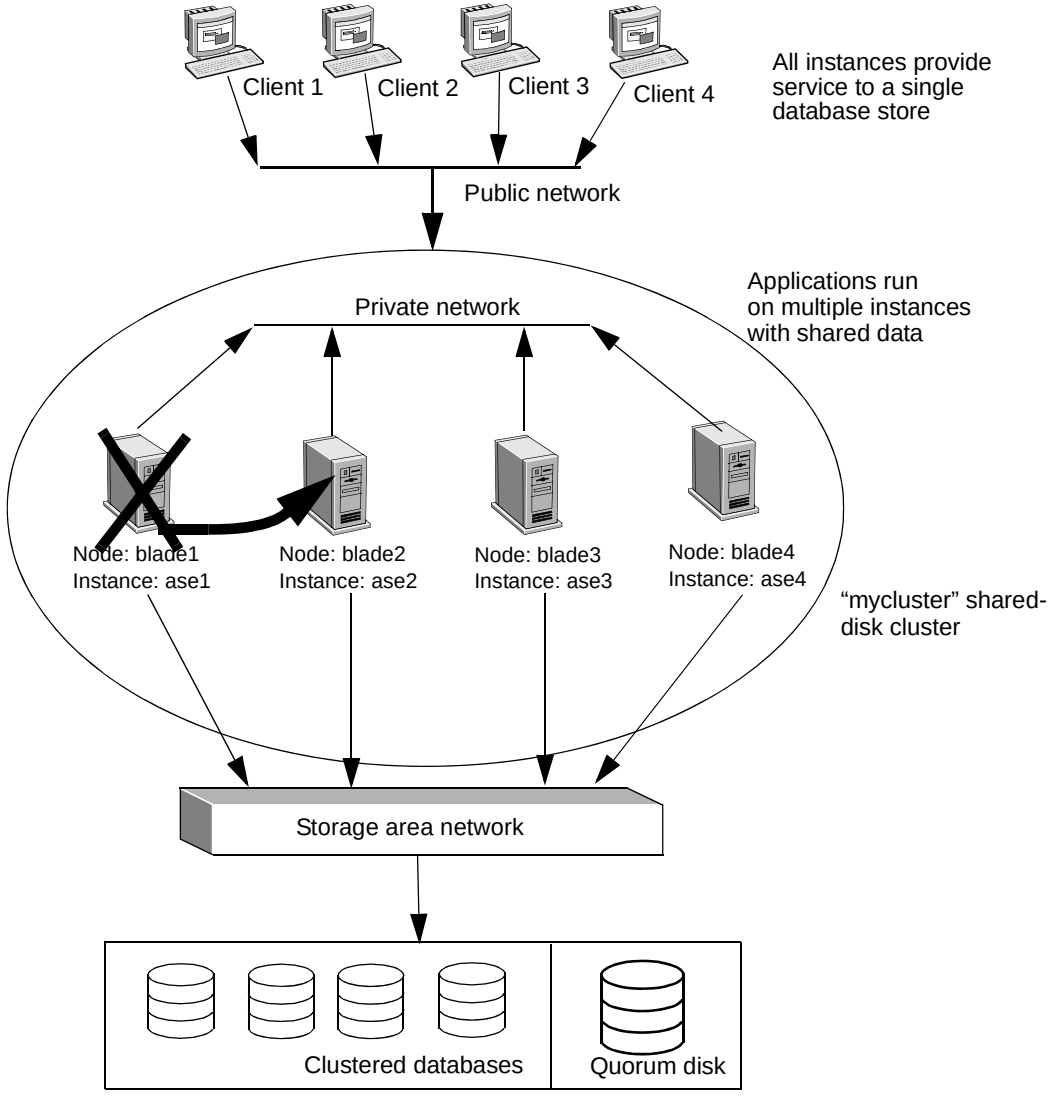
***Figure 1-1: Key cluster-aware components***




If one cluster member fails, its workload can be transferred to surviving cluster members. For example, if "ase1" fails, clients connected to that instance can fail over to any of the remaining active instances. See Figure 1-2 on page 5.

**Note** The Cluster Edition can handle multiple failures if they do not happen concurrently, and can recover fully from the initial failure before a subsequent failure occurs.

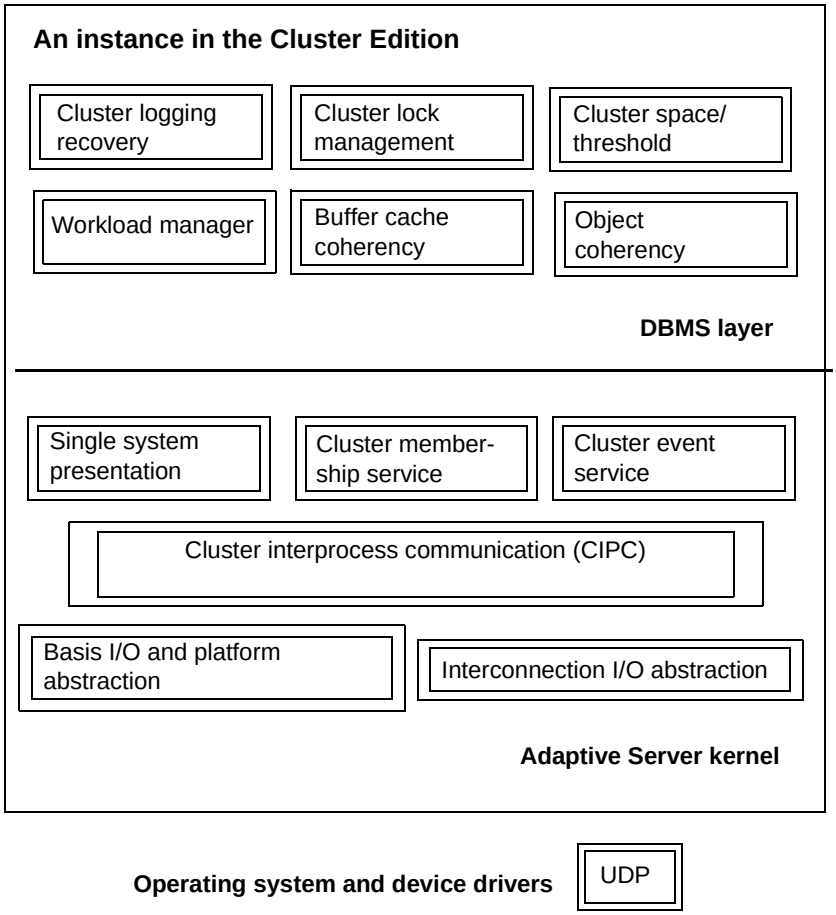
***Figure 1-2: How the cluster handles failure***



## Adaptive Server integrated clusterware

The Cluster Edition clusterware is integrated directly into Adaptive Server. No external clusterware is required to run Adaptive Server. Some of the components are new for the Cluster Edition; others are cluster-aware extensions of the existing Adaptive Server infrastructure. Figure 1-3 illustrates these components.

***Figure 1-3: Key cluster-aware components of an instance***




### The Adaptive Server kernel

These are new, native cluster infrastructure components.

- Cluster membership service – manages cluster membership and detects and handles instance failure.

- Cluster interprocess communication (CIPC) – provides messaging services and an interconnection abstraction layer that allows the instances to communicate with each other via redundant pathways.
- Cluster event service – supports a generic event-publishing and subscription mechanism for cluster-wide events.

DBMS layer

These key components in the Adaptive Server DBMS layer have been extended to work in the Cluster Edition environment:

- Buffer cache coherency – handles coherency issues related to the shared buffer cache and supports cache-to-cache transfer for allocation pages, index pages, data pages, object allocation map (OAM), and global allocation map (GAM) pages.
- Cluster lock manager – supports distributed locking for coherency control across the cluster.
- Cluster logging and recovery – handles logging from all instances, and fail over database recovery.
- Cluster space and threshold – handles space and threshold management in distributed environment.
- Object coherency – handles coherency issues related to sharing and transferring metadata and global variables. Object coherency must serialize updates to shared objects and make the latest changes available to all instances in the cluster.
- Workload manager – an Adaptive Server module that provides application-level management of resource allocation, availability, and load distribution.

## The cluster coordinator

The cluster coordinator handles specific tasks related to membership management and recovery. Any instance attempting to join an existing cluster first contacts the cluster coordinator.

There are no start-up parameters to indicate that a particular instance is the cluster coordinator, and you do not configure the cluster coordinator any differently than any other instance in the cluster. Initially, the cluster coordinator is the first instance you start. If the cluster coordinator exits, another instance dynamically assumes the coordinator role.

## The quorum device

The quorum device includes configuration information for the cluster and is shared by all cluster members. The quorum device must be on a raw partition and must be accessible to all nodes that host cluster instances.

Adaptive Server Cluster Edition uses the quorum disk for:

- A location to perform cluster membership management, including voting and arbitration for members joining
- A persistent place to store configuration data used by instances and the UAF
- A communications medium and synchronization point

The quorum device includes information about:

- The name of the cluster, the number of instances in the cluster, the path to the directories containing the interfaces file, log files, master device, and other required configuration information
- Cluster view records that indicate the state (up or down) of each instance in the cluster
- The area that Adaptive Server uses to determine the proper cluster membership when an instance failure is detected

Create the quorum device when you configure the cluster (see the *Installation Guide*). After the initial configuration, use sybcluster or the qrmutil utility to back up, restore, and reconfigure the quorum device. See Chapter 14, "The sybcluster Utility," and "qrmutil" on page 298 for more information.

## Database devices in the Cluster Edition

In the Cluster Edition, database devices—except those private devices that are used by local user temporary databases—must be raw devices (also known as character devices). You cannot use block devices as database devices because they can buffer disk writes at individual hardware nodes, causing data inconsistency among cluster instances.

You can create private devices on block devices. Private devices are used only by local user temporary databases. See your operating system documentation for more information about configuring raw devices.

You can create local user temporary databases on private devices, but you must create local system temporary databases on shared devices. For the Cluster Edition, you can use less expensive, local file system devices (block devices) for managing the storage needs of temporary data in the cluster. These devices are added as private devices and can only be used by local user temporary databases. See Chapter 7, “Using Temporary Databases,” for more information.

For example, on Linux systems the path */dev/sda* is a block device and should not be used. However you can bind this block device to a raw device such as */dev/raw/raw1*.

On Linux systems, you can distinguish character (raw) devices with the file type displayed using the command. Block devices include a `b` as the file type and character (raw) devices have a `c` as the file type:

```
[joeadminitrator@isles ~]$ ls -l /dev/sda
brw-rw----  1 root disk 8, 0 Nov 29 06:15 /dev/sda

[joeadministrator@isles ~]$ ls -l /dev/raw/raw1
crw-------  1 sybase sybase 162, 1 Nov 29 12:17 /dev/raw/raw1
```

On Solaris systems, the path */dev/dsk/c0t0d0s1* is a block device and should not be used. However, you can access this same storage as a character device with the path */dev/rdsk/c0t0d0s1*. Using an `ls -l` command on the character device indicates `raw` at the end of the symbolic link:

```
janeadministator% ls -l /dev/dsk/c0t0d0s1
lrwxrwxrwx   1 root     root          49 Apr 23  2007 /dev/dsk/c0t0d0s1 ->
../../devices/pci@780/pci@0/pci@9/scsi@0/sd@0,0:b
janeadministrator% ls -l /dev/rdsk/c0t0d0s1
lrwxrwxrwx   1 root     root          53 Apr 23  2007 /dev/rdsk/c0t0d0s1 ->
../../devices/pci@780/pci@0/pci@9/scsi@0/sd@0,0:b,raw
```

On HPIA systems, the path */dev/dsk/c0t0d0s1* is a block device and should not be used. However, you can access this storage as a character device with the path */dev/rdsk/c0t0d0s1*. You can distinguish character (raw) devices with the file type displayed using the `ls -l` command. Block devices include a `b` as the file type and character (raw) devices have a `c` as the file type:

```
[joeadminitrator@hpia207-HP-UX]: ls -l /dev/dsk/c0t0d0s1 ->
brw-r----  1 bin sys 31 0x000001 Sep 23 2008 /dev/dsk/c0t0d0s1

[joeadminitrator@hpia207-HP-UX]: ls -l /dev/rdsk/c0t0d0s1 ->
crw-rw-rw- 1 bin sys 188 0x000001 Sep 23 2008 /dev/dsk/c0t0d0s1
```

On IBM AIX systems, you can distinguish character (raw) devices with the file

type displayed using the `ls -l` command. Block devices include a `b` as the file type and character (raw) devices have a `c` as the file type. The path */dev/hdisk1* is a block device and should not be used. However, you can access this same storage as a character device with the path */dev/rhdisk1*:

```
janeadministator% ls -l /dev/hdisk1
brwxrwxrwx 1 root root 49 Apr 23 2007 /dev/hdisk1

janeadministrator% ls -l /dev/rhdisk1
crwxrwxrwx 1 root root 53 Apr 23 2007 /dev/rhdisk1
```

Database devices in the Cluster Edition must support SCSI-3 persistent group reservations (SCSI PGRs). The 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 (Sybase does support this configuration for test and development environments where data corruption is tolerated).

PGRs are a feature of the SCSI-3 protocol. However many storage area networks (SANs) that employ less expensive SATA disks still provide this functionality. Contact your storage vendor to verify your system supports SCSI-3 Persistent Group Reservation.

See also Chapter 9, “Additional Topics,” for information about devices and I/O fencing.

# How the Cluster Edition enhances the nonclustered edition

With the Cluster Edition, Sybase extends its support of symmetric multiprocessing (SMP), nonclustered servers and introduces an Adaptive Server that can work in a shared-disk environment.

You can group multiple nonclustered servers to provide a single-system view of shared databases that delivers improved reliability and ease of management.

A group of loosely coupled Adaptive Servers, each of which can function as a nonclustered server, work together to provide the user with a single database system image.

The Adaptive Servers in the cluster jointly manage a single installation of Adaptive Server databases residing on shared-disk devices.

The primary advantages of the Cluster Edition architecture are:

- Improved availability – the shared-disk nature of the architecture means that applications can continue to run as long as a single cluster member remains viable, even after several others have failed.
- Simple management – does not require data repartition as cluster membership changes, because data is shared by all instances.

**Note** The Cluster Edition provides a distributed architecture. Internode communication takes place via a network interconnection, not, as with nonclustered Adaptive Server, via shared memory (high-speed system memory). Applications that minimize internode messaging yield the best performance in the Cluster Edition environment.

## Using interconnected networks in the cluster

The Cluster Edition allows you to configure one or two interconnected networks between instances in the cluster. Although one interconnected network is sufficient, two interconnected networks allow for a redundant and more robust cluster. The interconnected networks form a series of logical links between every instance in the cluster. These links send messages between instances and are monitored by the Cluster Edition. If any failures are detected, the Cluster Edition reroutes traffic over the alternate networks between the instances.

An interconnection link may fail for various reasons: a physical failure such as a cable that is disconnected or broken, a power supply failure, such as a piece of network infrastructure equipment, or a software failure within the networking stack. The Cluster Edition detects these failures by monitoring the traffic flow between instances. Each instance monitors the messages sent over the various links. A link is considered operational as long as there are incoming messages.

An instance may not always send messages because the node has failed or the link is down. If a particular instance in the cluster is quiet, the Cluster Edition initiates an active probing mechanism to determine if the node supporting the instance can be contacted using the network link. This mechanism ensures that periods of inactivity do not falsely trigger a link failure and subsequent switching event.

If a link is flagged as inoperable, periodic attempts are made to establish if the link has been restored so normal operations can be resumed without manual intervention.

**Note** Link monitoring is automatically implemented when more than one network is defined in the Cluster Edition.

## Monitoring links between instances

The monCIPCLinks monitoring table monitors the state of the links between instances in the cluster. monCIPCLinks includes two states for each link: "passive" and "active."

**Note** A logical cluster and each instance in the cluster can have different states.

- A logical cluster has an overall, or global, state that determines, for example, whether the cluster is offline or online.
- A logical cluster also has an instance state that describes the state of a particular instance as it is perceived by a logical cluster.

See "Cluster and instance states" on page 97 for a detailed description of cluster "states."

The "passive" state is used to monitor day-to-day messages sent over the links. The Cluster Edition gathers the active state when there is no message traffic over the link. The status of the link is described as a "state," and each state has an age associated with it, in milliseconds. The states include "Up," "Down," and "In doubt". The state is "In doubt" when messages are not sent between the instances.

When the cluster is healthy, regular internode traffic is used to determine the state of the link. This is referred to as passive monitoring, and maintains the link's passive state. If the monitoring that determines the state occurs during a period of inactivity in the cluster, the defined state may become stale and unreliable (that is, a state that is determined to be Up during a period of inactivity may in fact be Down, but the inactivity prevents the monCIPCLinks table from showing this in the result set). This inactive state is described in the PassiveState column as "In doubt." Once a link is marked as "In doubt," the active link state monitoring it is triggered and the value described by the ActiveState column is valid.

Each of the active and passive states have an age associated with them, showing when the state was last updated. If the normal traffic is sufficient to maintain the link state, the active state is not updated and the age value associated with this state becomes large. The large value indicates that the associated state may no longer accurately represent the true state of the link.

If instances are not sending messages, the PassiveState is listed as In doubt, but the ActiveState shows the actual state: Up, In doubt, or Down.

This example shows a two-node cluster in which both links are running and have traffic flowing between them. Because the PassiveStateAge is 0 for all links, you can assume the output is a true reflection of the link state:

```
InstanceID    LocalInterface     RemoteInterface         PassiveState
      PassiveStateAge    ActiveState    ActiveStateAge

------------   --------------     ------------            --------------
      --------------     ------------   ---------------
2             ase2               ase2                    Up
      0                  Up             10300
2             blade2             blade1                  Up
      0                  Up             0
1             ase1               ase2                    Up
      0                   In doubt       17900

1             blade1             blade2                  Up
      0                   Up             100
```

This example shows the same two-node cluster after the primary interconnected network fails. The PassiveState value for the link between the network endpoints “ase1” and “ase2” is “`In doubt`”, and the value for the PassiveStateAge is “large” (indicating that the ActiveState represents the true state of the links). The ActiveState value is younger and shows the links as “Down”:

```
InstanceID    LocalInterface     RemoteInterface         PassiveState
      PassiveStateAge    ActiveState    ActiveStateAge

------------   --------------     ------------            --------------
      --------------     ------------   ---------------
2             ase2               ase1                    In doubt
      13500              Down           700
2             blade2             blade1                  Up
      0                  Up             700
1             ase1               ase2                    In doubt
      13600              Down           400
1             blade1             blade2                  Up
```

```
0                    Up              400
```

**Note** There is a slight delay between the failure of a link and the time the active state truly reflects the state of the link

Ignore any state with the value “large” for ActiveStateAge since this indicates the link is old and the value may be inaccurate. When the link state is old and the value for ActiveStateAge is “large”, active monitoring is triggered by the absence of messages, but has not yet determined the link state.

# Suggested deployment scenarios

In general, the Cluster Edition supports the scenarios described in this section, except those that include the features listed in the Release Bulletin for your platform.

Most users considering clustered database architectures have these objectives:

- Increased availability – if a node fails, other nodes in the cluster continue to run and the database continues to be available.
- Increased manageability – multiple applications and databases can be consolidated into a single cluster, thus reducing management complexity and introducing economies of scale.
- Increased scalability – support for multiple nodes that allow clustered databases to scale beyond the limits imposed by single-node environments. Commonly referred to as “vertical scalability,” this means increasing the processing capacity of a single node by adding more CPUs, memory, host bus adaptors (HBAs), network interface cards (NICs), and so on.
- Reduced total cost of ownership – software that can be deployed on industry-standard, nonproprietary hardware, thereby reducing the cost of purchase, maintenance, and support.

## HA failover for OLTP applications

If you currently run a production environment using high availability (HA) capabilities, replace those currently provided by Sybase Failover in either active-passive (standby) or active-active (companion) configurations, or the operating system–provided clusterware (for example, Sun Cluster and Veritas Cluster) with one of the Cluster Edition scenarios described below.

Choose and configure a failover scenario that best represents what your company would want given the service-level agreement requirements of the application and financial constraints.

- *1:1 active-passive*

  Cluster nodes and instances are set up in pairs with an idle (passive) node and instance that wait for the corresponding active node to fail. This scenario is cost effective only in extreme environments where requirements do not tolerate service degradation in any failover scenario—including multiple failures.

- *1:1 active-active*

  Cluster nodes and instances are set up in pairs, and each pair services a separate application and database (instance) while monitoring each other in "companion mode" in case the other fails. Although this scenario mimics the current Sybase HA option and provides full utilization of resources, service levels during failed-over processing degrades unless you maintain resource capacity during normal processing that is low enough (less than 50%) to provide sufficient capacity for a single instance to run the workload of both.

- *N:1 (N active nodes covered by a single passive standby node)*

  This scenario provides a single passive standby node and instance to monitor an arbitrary number of active instances. This is the most cost-effective scenario; multiple node failures that fail instances over to a single node can lead to a level of service degradation that can make this scenario unacceptable. Configure the passive node with addition capacity (for example, CPUs and memory) to mitigate the degradation in the most likely failover scenarios.

- *N:M (N active nodes and instances covered by M passive standby nodes)*

This model provides an arbitrary number (M) of passive standby instances to monitor an arbitrary number (N) of active instances. This option reduces costs while covering most typical failure scenarios. The choice of the number of passive standby nodes is typically driven by cost, system-level mean time between failures (MTBF) statistics, and the business impact of running at degraded service levels in a multiple-node failure scenario. In most failure scenarios, you can configure passive nodes with additional capacity to mitigate this degradation.

## Horizontal scalability for DSS reporting applications

if you have applications in which the online activity consists primarily of large-scale reports and decision support systems (DSS) querying from a large user population, consider creating a multiple-node cluster in which all Adaptive Server instances service these same applications.

When organizing this system, pay attention to:

- The scalability of users, queries, and response time as the workload expands from one to *N* nodes.
- The load distribution of clients across instances and their relationship to instance capacity and performance.

## Horizontal scalability for OLTP applications

Applications that are read-and write-intensive—such as online transaction processing (OLTP) applications—traditionally pose challenges for scalability because of resource contention while maintaining the ACID properties of a database system. The horizontal scalability of OLTP applications when using shared-disk cluster technology adds additional challenges because of the data coherency and messaging that is required across the server instances in a shared-disk cluster. The physics of computers and networking does not allow OLTP applications to scale infinitely on a shared-disk cluster. As the number of nodes and the load and number of users increase, the amount of internode messaging necessary to maintain buffer coherency across the nodes increases exponentially.

The best method for scaling OLTP workloads on a shared-disk cluster is to partition the application and data into mutually exclusive sets (that is, to separate the data to different databases) to avoid processing coordination across server instances and access the data for an application from the same instance. Because of this, you must carefully consider how you partition "data" at a database level to eliminate the log and data contention across the participating instances. You can facilitate access to this segmented "data" through a single instance with logical clustering and workload management.

Logical clusters allow you to allocate distinct instances to different applications or workloads, logically enabling these groups of data to operate under a single cluster. This reduces inter-instance access and, combined with dedicated temporary databases for each of the instances, helps to deploy an OLTP application in the Cluster Edition that supports continuous data availability.

# New client technologies in the Cluster Edition

The Cluster Edition supports a single-system presentation. That is, multiple instances that make up the cluster appear to clients as a single system. New client technologies allow clients to connect logically to a cluster while remaining connected physically to individual instances. This logical connection allows Adaptive Server to redirect the client to various instances in the cluster and to dynamically provide the client with high-availability fail over data. See Chapter 2, "Client Applications and Client/ Server Interaction," and *New Features Open Server 15.0 and SDK 15.0 for Microsoft Windows, Linux, and UNIX* for more information.

New client technologies include:

- Login redirection – when the client reconnects to another instance in the cluster.
- Connection migration – when an established connection moves to another instance in the cluster.
- Extended high-availability failover – allows the client to fail over multiple times until it finds the first available or least-loaded instance.

# Support for replication

The Cluster Edition supports replication using Replication Server and the RepAgent thread. A clustered database can be a source or a destination in a Sybase clustered system. You can perform all of the tasks, such as configuring RepAgent or marking tables for replication, from any instance in the cluster. Replication status is coherent across the entire cluster.

## Configuring the RepAgent

When you configure a primary database in a clustered system, the server name you specify should be the cluster name. You can display the cluster name using select @@servername.

The syntax for sp_config_rep_agent does not require a cluster or instance name. By default, both the Cluster Edition and the nonclustered Adaptive Server edition assume the value of select @@servername. In the Cluster Edition, this statement returns the current cluster name. For example:

```
1> select @@servername
2> go

---------------------------
MYCLUSTER
```

## Starting the RepAgent

By default, RepAgent starts on the coordinator.

However, you can configure RepAgent to start on any instance in the cluster. For example, to configure RepAgent on the primary database pdb to always start on the “ase2” instance, enter:

```
sp_config_rep_agent pdb, "cluster instance name",
"ase2"
```

After configuration, you must restart RepAgent using sp_start_rep_agent for the new configuration to take effect.

To return to the default behavior with RepAgent always starting on the coordinator, enter:

```
sp_config_rep_agent pdb, "cluster instance name",
"coordinator"
```

When an instance starts on its node, it checks if databases are configured to start on its node. If yes, and if the database is marked to start automatically, the RepAgent starts.

When the coordinator starts, it starts all RepAgents not configured to start on a specific instance. If the coordinator fails, or is stopped with a graceful shutdown, a RepAgent starts on the new coordinator.

If a RepAgent is configured to start on a specific instance other than the coordinator, and this instance is shutdown or fails, the RepAgent starts on the coordinator.

**Note** The Cluster Edition does not support Adaptive Server Enterprise Replicator, which requires the unsupported dbcc logtransfer interface.

CHAPTER 2

# Client Applications and Client/ Server Interaction

This chapter describes client/server interaction and how to modify applications that make Open Client/Client-Library calls to support clusters. It describes using isql in a shared-disk cluster environment.



**Note** DB-Library can connect to an instance in a shared-disk cluster, but does not support the Sybase shared-disk or high-availability functionality.

The version of Open Client, jConnect, ODBC, OLE DB, and ADO.NET that ships with the Cluster Edition supports:

- Login redirection – the ability of an instance to redirect an incoming client connection to another instance prior to acknowledging the login. Login redirection occurs during the login sequence. The client application does not receive notification that it was redirected.
- Connection migration – occurs when an existing client is transferred from one instance of a cluster to another. See “Connection migration” on page 26 for information about when migration can occur, and connection criteria.
- Extended high-availability failover – in an extended failover configuration, Adaptive Server provides a list of failover addresses to “high-availability-aware” clients when they connect. This allows high-availability-aware clients or applications to failover multiple times if the instance to which they are connected fails.

These clients are not required to have a HAFAILOVER entry in their interfaces file or directory services. However, if they do have an HAFAILOVER entry in their interfaces file or directory services, the clients continue to use this entry until Adaptive Server sends them a list of failover addresses or servers to connect to. The clients always use the latest list Adaptive Server provides.

To implement login redirection and connection migration, make sure the application uses a current copy of the client libraries:

- If the application is linked to shared libraries – include the new client libraries in the library search path before the old libraries.
- If the application is statically linked – relink the application.

Use the CS_PROP_MIGRATABLE connection property to enable or disable connection migration. CS_PROP_MIGRATABLE is on by default. For more information, see the *Client Library Reference Manual.*

To implement extended failover, make sure the application uses a current copy of the client libraries and that you have enabled high availability. See *Using Sybase Failover in a High Availability System* for information about enabling high availability.

For information about using the failover features with jConnect, ODBC, OLE DB, and ADO.NET Driver, see "*New Features Open Server 15.0 and SDK 15.0 for Microsoft Windows, Linux, and UNIX*"

# Open Client

Login redirection, connection migration, and extended high availability failover are supported for these versions:

- Login redirection – OCS version 15.0
- Extended high-availability – OCS version 15.0 ESD #3
- Migration – OCS version 15.0, ESD #8

The SDK component that supports these versions is OpenClient/Client-Library.

# Enabling failover in Client-Library applications

Any existing application can connect to the Cluster Edition. However, to use extended high-availability (HA) capabilities, you may need to change application code.

- For existing HA applications based on existing Adaptive Server HA functionality, no application code changes are required.
- Existing non-HA applications may benefit from some aspects of the HA capabilities of the Cluster Edition with no code changes, or minor ones. However, in these cases, failover is not transparent: the application receives an error message when failover is first detected. The user must resubmit the batch or transaction to initiate failover.

  To enable failover for non-HA applications:

  - For isql, specify the -Q option when connecting to Adaptive Server.
  - For applications linked with Client-Library, set a corresponding connection property that enables failover.

To make failover transparent to users, the application must actively check for failover error status, and automatically resubmit the batch or transaction.

In all cases, you must update the Client-Library version used by the application to use the cluster-related HA capabilities.

To enable failover in Client-Library applications:

1 Set the CS_HAFAILOVER property at either the context or the connection level using the ct_config or ct_con_props Client-Library API calls:

```
ct_config (context, action, CS_HAFAILOVER, buf,
buflen, &outlen);
ct_con_props(connection, action, CS_HAFAILOVER,
buf, buflen, &outlen);
```

   See the *Client-Library/C Reference Manual* for more information about the CS_HAFAILOVER property.

2 If you attempt to connect to an instance that is down, behavior is the same as with a nonclustered Adaptive Server: Client-Library tries all the query entries for the instance name in the interfaces file until one of them works, or it has none left to try. Include query lines in the client-side interfaces file for all instances. Applications can connect to the cluster, which is represented by a series of interfaces file query entries. For information about the interfaces file, see the *Cluster Edition Installation Guide*.

3 When a successful failover occurs, the Client-Library issues a return value named CS_RET_HAFAILOVER, which is specific to several Client-Library API calls, including:

```
ret = ct_results(cmd, result_type)
ret = ct_send(cmd)
```

CS_RET_HAFAILOVER is returned from the API call during a synchronous connection (a routine that requires a server response and blocks until the response is received). In an asynchronous connection (a routine that requires a server response returns CS_PENDING immediately), these APIs issue CS_PENDING, and the callback function returns CS_RET_HAFAILOVER. Depending on the return code, the customer can perform the required processing, set up the context, and send the next command to be executed.

4 Rebuild your applications using the Open Client SDK with a version that is at least equal to the version of the Open Client SDK shipped with the Cluster Edition.

See *Using Sybase Failover in a High Availability System* for information about configuring applications for high availability.

# Client/server interaction

The features in this section, which use Open Client 15.0 libraries, are enabled automatically.

## Login redirection

Login redirection occurs at login time when an instance tells a client to log in to another instance because of load considerations.

You do not need to perform any additional configuration for client redirection.

Login redirection is used by the Adaptive Server workload manager to send incoming connections to specific instances based on the logical cluster configuration and the cluster's current workload.

If you attempt to connect to an instance that is down, behavior is the same as with a nonclustered Adaptive Server: the client tries all entries in the directory service of a given server until it can connect successfully. Because of this, your server entries in the directory service should contain connection information for all instances in the cluster.

This example includes the Adaptive Servers “ase1,” “ase2,” “ase3,” and “ase4,” on machines “blade1,” “blade2,” “blade3,” and “blade4,” running in the cluster “mycluster.”

```
ase1
   query tcp ether blade1 19786
ase2
   query tcp ether blade2 19786
ase3
   query tcp ether blade3 19786
ase4
   query tcp ether blade4 19786
mycluster
   query tcp ether blade1 19786
   query tcp ether blade2 19786
   query tcp ether blade3 19786
   query tcp ether blade4 19786
```

For example, if a client connects to cluster “mycluster,” it first tries to connect to the “ase1” instance. If “ase1” is down, it tries the next entry in the interfaces file, “ase2”, and so on. After a successful connection, the workload manager may redirect the client to another instance based on workload rules.

Although instances are tried in the order specified in the interfaces file, it can take a considerable amount time for a connection attempt to fail when hosts or the network are unreachable or down. You can expedite the retry attempt by adding a login timeout to the connection information.

In the example above, if you specified a login timeout interval that is shorter than the default for the connecting client, the client could attempt to connect to instance “ase2” more quickly.

For more information, see the isql -l parameter description in the Adaptive Server *Utility Guide* and CS_LOGIN_TIMEOUT property for *Client-Library Reference Manual*.

## Connection properties for login redirection

Set the connections properties to configure login redirection:

- CS_PROP_REDIRECT – enables and disables login redirection
- CS_DS_RAND_OFFSET – disables or enables making first query entry randomly retrieved from the directory service lookup for ct_connect. By default, this property is set to false.

For more information, see the *Client Library Reference Manual*.

## Connection migration

Connection migration occurs when an existing client is transferred from one instance of a cluster to another. For example, a connection may migrate because the instance to which it is currently connected is brought down for maintenance, or it may migrate for load balancing. The transfer is transparent to the client application. Connection migration allows the cluster to balance the load throughout the logical cluster.

Connection migration allows the workload manager to gracefully move existing connections between instances during administrative failover, failback, or logical clusters going offline. The workload manager can use migration for dynamic load distribution, during which some existing connections are migrated between instances to more evenly distribute load.

Connection migration is enabled automatically when an instance uses the Open Client 15.0. client libraries. You do not need to perform any additional configuration for connection migration.

### Difference between migration and failover

Migration is a planned, controlled event that Adaptive Server requests. Failover is an unplanned event that occurs after an Adaptive Server crash or network disconnect.

Applications are not aware of—and you do not need to write code for—migration. However, you must specifically code applications for failover support.

Migration restores the full session context on the new instance. Failover does not; it is up to the application to restore its own context.

### When can migration occur?

The workload manager can initiate a migration when a request can successfully be sent to a client. Specifically, migration can occur:

- To connections that have completed their login:
  - After an instance receives a new batch from the client, but before the batch is parsed and executed
  - After an instance completes the processing of a client batch but before it sends a final completion to the client.
  - When an instance is not executing any batch on behalf of the client.
- According to the workload manager's algorithm. The Work load Manager may migrate certain clients for load balancing or because the current instance is being brought down for maintenance.
- When a connection's context accommodates migration. The workload manager targets connections for migration, but these connections migrate only when their context allows this. In particular, no migration can occur inside a transaction.

## Migrated contexts

The source instance propagates the client's full context to the destination instance when it has successfully completed the migration, and the destination instance retrieves the context.

The client's full context is restored after the migration is complete, making migration completely transparent to the client. However, a migrated connection acquires a new spid.

The client's context consists of the following elements:

- The name of the current database
- Any pending batch of commands if the migration occurs in pre-batch mode
- The client's login record
- The client's language and character sets
- The client's capabilities
- Any monitor counters and statistic
- Any roles
- Any set options
- Trace flags 3604 and 3605

### Criteria for migration

Migration is an asychronous event: a request is issued for the task to migrate, and the task migrates only when it reaches a quiescent state. For the Cluster Edition, a quiescent state is one that:

- Is not executing a query batch
- Has no open transactions
- Has no session-level temporary tables
- Has no declared cursors
- Has not changed its password since its initial connection
- Has not run set user or set proxy
- Is not bound to an engine using the logical process manager
- Is not using real-time data services (RTDS)
- Is not a logical connection associated with an inbound site handler
- Has not kept a database open in single-user mode.

## Context migration

When Adaptive Server migrates an existing connection to another instance, it must also migrate some of the context from the existing connection, such as the current database and set options.

### Configuring a client migration

The idle migration timer and session idle timer configuration parameters control when idle clients migrate. The sum of these two parameters determines the upper limit for the number of seconds during which migration is expected to complete.

The default setting for idle migration timer is 60, measured in seconds. The default setting for session idle timer is 600, measured in seconds.

If you set idle migration timer to 0, the instance closes the connection on which a migration request was issued immediately after it sends the migration request. If you set session idle timer to 0, the instance invalidates any idle migration not completed before idle migration timer expires.

Setting session idle timer to a high value increases the chance for idle connections to migrate successfully. However, the instance must preserve the contexts of the migrating clients for a long time. It also means that setting both idle migration timer and session idle timer to 0 disables migration. For more information, see "New configuration parameters" on page 290.

The examples below describe various configurations for idle migration timer and session idle timer.

Both parameters set to the default

- If an instance issues a migration request against an idle client that cannot process asynchronous notifications, and the user or the application issues a command on the client before the expiration of the idle migration timer, the client migrates immediately and the command is executed on the destination instance.

  "Read ahead" clients read any pending data from the network before the application requests it.

- The instance closes the connection to the client, if the following are all true:
  - Both idle migration timer and session idle timer are set to their default values.
  - An instance issues a migration request against an idle, non read-ahead client.
  - The client remains idle for the duration of idle migration timer.
- If a command is issued on the client during the first 600 seconds (default value of idle migration timer) following the closure of the connection by the instance, the client migrates successfully and the command is executed on the target instance.
- An instance closes the connection to the client if the following are true:
  - Both idle migration timer and session idle timer are set to their default values.
  - An instance issues a migration request against an idle, non read-ahead client.
  - No command is issued against the client before the expiration of the idle migration timer.
- An instance invalidates the migration if the following are true:
  - Both idle migration timer and session idle timer are set to their default values.

- An instance issues a migration request against an idle, non read-ahead client.
- No command is issued against the client before idle migration timer expires.

  The client attempts to migrate as soon as it detects the migration request, because a command is issued. However, the instance rejects the migration request and the client tries to continue on the initial instance. Because the connection to the instance is closed (idle migration timer timed out), the client attempts an HA failover if it has HA capability, otherwise, it reports a disconnect to the application.

| | |
|---|---|
| idle migration timer set to 0, session idle timer set to 600 | If an instance issues a migration request against a idle, non-read-ahead client, the instance closes the connection to the client immediately after sending the migration request. If a command is issued against the client during the initial 600 seconds, the client migrates successfully, otherwise, it fails in the same manner described in the previous bullet. |
| idle migration timer set to 60, session idle timer set to 0 | The instance issues a migration request against an idle, non-read-ahead client. If a command is issued against the client before idle migration timer expires, the client migrates successfully. However, if no command is issued before idle migration timer expires, the client can migrate, since session idle timer is set to 0. |
| idle migration timer and session idle timer set to 0 | If both parameters are set to 0, neither instance migrates clients. |

## Extended high-availability failover

Adaptive Server provides a list of failover addresses to “HA-aware” clients when they connect. This allows high-availability-aware clients or applications to failover multiple times whenever the instance to which it is connected becomes unavailable. If the instance has not sent a failover list to the client, the client uses the HAFAILOVER entry information in the interfaces file.

This example allows an HA-aware client to failover if there is a network failure during login before the instance sends the extended high-availability list:

```
ase1
        query tcp ether blade1 19786
ase2
        query tcp ether blade2 19786

mycluster
        query tcp ether blade1 19786
```

```
query tcp ether blade2 19786
hafailover mycluster
```

The HAFAILOVER entry should use the cluster alias as the server name since a client application tries each query line until it establishes a connection to an instance in the cluster.

Extended failover requires Open Client 15.0, ESD #3 or later. The client libraries in the Cluster Edition contain ESD #8.

Open Client uses the CS_PROP_EXTENDEDFAILOVER property for extended failover. See the *Client-Library/C Reference Manual* for more information.

### Differences between HA failover and failover in clusters

From the client side, enabling high availability means the application receives the error code CS_RET_HAFAILOVER during a network failure, and the client library automatically reconnects to a server. In high availability, the application again attempts to connect first to the original server, and then tries the secondary server (as set in the interfaces file entry for the primary server) if the connection fails. If you use the Open Client libraries shipped with the Cluster Edition, the instance sends a failover target list to the client. The application uses this list to determine where to connect when the instance fails.

Whether you use high availability or the Cluster Edition, failover can occur at any point.

## Using *isql* in a clustered environment

By default, you can use isql to connect to an Adaptive Server instance in a shared-disk cluster environment. However, to connect to an Adaptive Server instance in a shared-disk cluster and turn on high-availability failover or extended high-availability failover for the client, you must start isql using the -Q option.

---

**Note** Although isql -Q can use the extended high-availability capabilities, it is not transparent: when an instance fails, isql receives an error and you must resubmit the batch or transaction.

---

See the *Utility Guide* for more information.

# Using remote procedure calls in a clustered environment

Remote procedure calls (RPCs) allow clients to initiate stored procedures on a remote server. The remote server runs the system procedure in its context as if it was requested by a local client, and sends back results to the original server over the network.

In addition to server-to-server RPCs, the Cluster Edition includes three additional classes of RPCs. The first class involves an RPC where the remote server is a cluster. The second class involves an RPC where the local, or originating, server is a cluster instance. The third class involves an RPC where both the local and remote servers are instances in the same cluster.

In earlier versions of Adaptive Server, the default value for cis rpc handling was 0. In the Cluster Edition, the default value is 1, which forces RPC handling to use Component Integration Service (CIS) as the default RPC handling mechanism instead of the site handler.

A cluster instance identifies itself to remote servers using the name of the cluster, not the name of the instance. The *@@servername* global variable returns the name of the cluster.

## RPCs where the remote server is a cluster

Because the Cluster Edition supports a single-system image, using a cluster as a remote server has minimal impact on the server sending the RPC request.

If the server sending the RPC is using cis rpc handling, the cluster perceives the inbound request as a regular client connection. The workload manager attempts to route the RPC to the appropriate instance and logical cluster based on the configured routing rules. The workload manager rules may dictate login redirection, as long as the initiating server indicates that it supports login redirection. Adaptive Server 15.0 and later support login redirection for RPC requests.

If the initiating server uses site handlers, the cluster workload manager is bypassed and the RPC runs in the system logical cluster of the instance that accepted the connection.

## RPCs where the local server is a cluster

The Cluster Edition does not support outbound RPCs through site handlers, and, by default, uses CIS RPC handling. Because outbound RPCs from a cluster are identified using the name of the cluster, configure remote servers to accept RPCs from the cluster, rather than from the individual instances.

## RPCs where local and remote servers are instances in the same cluster

Instances in a cluster occasionally use RPCs for intra-cluster communication. When the cluster is started, Adaptive Server automatically adds each cluster instance to the sysservers table, and removes any cluster instances from sysservers that are no longer in the cluster definition. This is also done when instances are dynamically added and dropped at runtime. These sysservers entries have the cluster instance status bit set. Because intra-cluster RPCs are intended for specific instances, the sysservers entries do not have the enable login redirection status bit set.

## sp_serveroption

The sp_serveroption system procedure includes the enable login redirection and cluster instance options.

enable login redirection

enable login redirection determines if RPC requests are sent to another cluster. The syntax is:

```
sp_serveroption server_name, 'enable login redirection', [ true | false ]
```

where:

- *server_name* – is the name of the remote server for which you are setting enable login redirection. This remote server must be included in sysservers.
- true – means the instance can redirect RPC requests to another cluster.
- false – means the instance cannot redirect RPC requests.

By default, enable login redirection is enabled. (You must have the sa_role to run sp_serveroption.)

cluster instance

cluster instance identifies sysservers entries that store instance information, where *instance_name* is the name of the instance you are adding.

```
sp_serveroption instance_name, 'cluster instance', [ true | false ]
```

By default, *cluster_instance* is disabled (set to false) for each remote server.

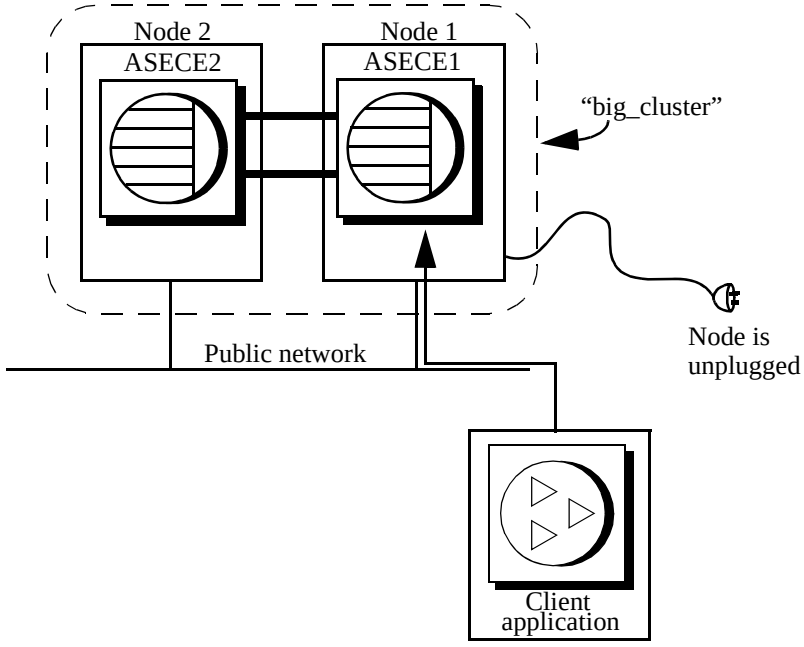
The Cluster Edition automatically manages the sysservers rows for instances in the local cluster. You need not manually set or clear the `cluster instance` flag.

# Reconnecting clients when a node loses power

If the network cable is removed from a machine or if a node to which a client is connected loses its power, the client-side socket becomes unreachable. The client socket waits, without results, for a reply from the server or waits for the cluster to issue a send operation.

In the situation shown below, a client application is connected to “big_cluster”, which consists of “Node1” and “Node2” on which instances “ASECE1” and “ASECE2” are running, respectively. A client application is connected to instance “ASECE1” running on “Node1”.

If the power is disconnected from “Node1”, the client application waits for contact from the node. The only way to avoid this situation is to configure the client application to assume the node is down after a specified amount of time. It then connects to another node in the cluster.




The operating system network detects a crash, disconnects the clients, and fails over the sockets from the remote side of the connection.

To reduce the time required to detect when a cluster loses a host or when a public network is disconnected from a node running an instance, you can:

- Set TCP keepalive to a reasonable value on the host on which the client is running.
- Set the client application's timeout value.

Setting TCP *keepalive* to a shorter value

The TCP keepalive parameter eventually marks the client socket as failed. However, because the default value of the TCP keepalive value is a long amount of time (in some systems it may be set to as long as two hours), it may be three or more hours before the client-side sockets fail over. Setting keepalive to a small value (several minutes) may not be practical for large organizations, but you can set keepalive to a period of time that is appropriate for your site, that works with the HAFAILOVER capabilities.

Set TCP keepalive on client machines. The appropriate values vary, depending on the operating system you use. See your client's operating system documentation for more information.

If you are testing for client timeouts, set the values for the parameters in the first two columns of Table 2-1 to a few minutes, and set the values for the parameters in the third column to a low number.

***Table 2-1: Setting TCP keepalive***

| Operating system | Amount of time parameter waits before probing the connection | Amount of time between probes for parameter | Maximum amount of time or attempts for parameter to probe connections before dropping them |
|---|---|---|---|
| Solaris | N/A | tcp_keepalive_interval<br>Measured in milliseconds | N/A |
| Linux | tcp_keepalive_time<br>Measured in seconds | tcp_keepalive_intvl<br>Measured in seconds | tcp_keepalive_probes<br>Measured as absolute number |
| Windows XP | KeepAliveTime<br>Measured in seconds | KeepAliveInterval<br>Measured in seconds | TCPMaxDataRetransmissionsions<br>Measured as absolute number |
| HP-UX | tcp_time_wait_interval<br>Measured in milliseconds | tcp_keepalive_interval<br>Measured in milliseconds | tcp_keepalive_kill<br>Measured in milliseconds |

### Set client's timeout value

There are two different timeout properties you can set for Client-Library program connections:

- CS_LOGIN_TIMEOUT – determines how long the client waits to connect to an unreachable host.
- CS_TIMEOUT – determines how long a client waits for commands to complete.

Based on how you configure the timeout event, the client either fails or fails over to another node.

You can configure clients to set the Client-Library CS_TIMEOUT parameter to determine how long to wait before they time out.

You must set the CS_TIMEOUT and CS_LOGIN_TIMEOUT parameters or the isql -t and -l parameters for clients to fail over during a sudden loss of power to the node.

For more information about the Client-Library parameters CS_TIMEOUT and CS_LOGIN_TIMEOUT, see the *Client-Library/C Reference Manual*. For information about CS_HAFAILOVER, see "Client/server interaction" on page 24.

For information about using the isql -t and -l parameters, see the *Adaptive Server Enterprise Utility Guide.*

CHAPTER 3

# Using Security with a Clustered Environment

This chapter discusses configuring security in a clustered environment.



For information about auditing, see “Adding auditing” on page 395.

## Using SSL in a clustered environment

The Cluster Edition allows the server name specified in the directory service entry to be different from the common name the SSL server certificate uses for performing an SSL handshake. This allows you to use a fully-qualified domain name for the SSL certificate common name (for example *server1.bigcompany.com*) and use the same certificate for multiple servers.

To add a common name to the interfaces file, use this format:

```
ase1
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
ase2
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
ase3
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
mycluster
   query tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
```

Where *common_name* is the fully-qualified domain name for the cluster node. *common_name* can include white space. Instances defined in the interfaces file may or may not use the same common name.

**Note** You can add only one SSL certificate to a master database. Because each instance in a cluster shares the same disk, they all use the same path for the SSL server certificate. Sybase recommends that all instances use the same common name.

For example, this is a sample interfaces file entry for cluster mycluster:

```
ase1
   master tcp ether blade1 19786 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade1 19786 ssl="CN='ase1.big server 1.com'"
ase2
   master tcp ether blade2 19886 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade2 19886 ssl="CN='ase1.big server 1.com'"
ase3
   master tcp ether blade3 19986 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade3 19986 ssl="CN='ase1.big server 1.com'"
mycluster
   query tcp ether blade1 19786 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade2 19886 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade3 19986 ssl=ssl="CN='ase1.big server 1.com'"
```

## Specifying a common name with sp_listener

sp_listener includes the CN=*common_name* parameter, which allows you to specify a common name for the SSL certificate. The syntax is:

sp_listener 'command','[protocol:]*machine_name*:*port_number*:"CN=*common_name*"', '*engine_number*'

Where CN=*common_name* is used only if you specify ssltcp as the protocol. If included, it uses the specified *common_name* to validate the *common_name* in the SSL certificate. If you do not include CN=*common_name*, Adaptive Server uses *server_name* to validate the common name in the SSL certificate. CN=*common_name* must match the common name entry in the certificate. If you include a fully-qualified domain name in the certificate, it must match the CN=*common_name*.

The attribute name “CN” is case insensitive, but the attribute value for the common name is case sensitive. For example, the attribute name may be “CN,” “Cn,” or “cn.”

For example, this specifies the common name `ase1.big server 1.com`:

```
sp_listener 'start','ssltcp:blade1:17251:"CN=ase1.big server 1.com"','0'
```

See the *Adaptive Server Reference Manual* for more information about sp_listener.

# Using LDAP as a directory service

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

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

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

The LDAP server stores and retrieves information about:

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

The LDAP server can be configured with these access restrictions:

- Anonymous authentication – all data is visible to any user.
- User name and password authentication – Adaptive Server uses the user name and password for UNIX platforms:
  - *$SYBASE/$SYBASE_OCS/config/libtcl.cfg* on 32-bit platforms
  - *$SYBASE/$SYBASE_OCS/config/libtcl64.cfg* on 64-bit platforms

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

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

When an LDAP server is specified in the *libtcl.cfg libtcl64.cfg* or *libtcl64.cfg* file (collectively the *libtcl*.cfg* file), the server information is searched for using the ordered list of directory services from the *libtcl*.cfg* file. If the information is not found there, it then searches the interfaces file.

For the Cluster Edition, an interfaces file may be set in the quorum file. When the quorum file specifies an interfaces file, the Cluster Edition ignores the directory services specified in *libtcl*.cfg* files.

If multiple directory services are supported in a server, then the order in which they are searched is specified in *libtcl*.cfg*. You cannot specify the search order with the dataserver command-line option. See "Multiple directory services" on page 46.

## LDAP directory services versus the Sybase interfaces file

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

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

Table 3-1 highlights the differences between the Sybase interfaces file and an LDAP server.

***Table 3-1: interfaces file versus LDAP directory services***

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

LDAP directory services support more attributes than the Sybase interfaces file. These attributes can include server version, server status, and so on. See Table 3-2 for a list of attributes.

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

Table 3-2 lists the Sybase LDAP directory entries.

***Table 3-2: Sybase LDAP directory definitions***

| Attribute name | Value type | Description |
|---|---|---|
| ditbase | *interfaces* file or *libtcl.cfg* | DIT base for object tree. If the *libtcl.cfg* file is specified, the *interfaces* file is ignored. The *libtcl.cfg* file can be overridden with ct_con_prop() for a specified connection. |
| dn | Character string | Distinguished name. Must be unique name that identifies the object. |
| sybaseVersion | Integer | Server version number. |
| sybaseServername | Character string | Server name. |
| sybaseService | Character string | Service type: Sybase Adaptive Server, Sybase SQL Server, or ASE. |
| sybaseStatus | Integer | Status: 1 = Active, 2 = Stopped, 3 = Failed, 4 = Unknown. |
| sybaseAddress | String | Each server address includes:<br>• Protocol: TCP, NAMEPIPE, SPX DECNET (entry is case sensitive).<br>• Address: any valid address for the protocol type.<br>**Note** dscp splits this attribute into Transport type and Transport address.<br>• Filter: None, ssl, or ssl="CN=common_name". |
| sybaseSecurity (optional) | String | Security OID (object ID). |
| sybaseRetryCount | Integer | This attribute is mapped to CS_RETRY_COUNT, which specifies the number of times that ct_connect retries the sequence of network addresses associated with a server name. |
| sybaseRetryDelay | Integer | This attribute is mapped to CS_LOOP_DELAY, which specifies the delay, in seconds, that ct_connect waits before retrying the entire sequence of addresses. |
| sybaseHAservername (optional) | String | A secondary server for failover protection. |

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

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

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

```
dn: sybaseServername=foobar, dc=sybase,dc=com
objectClass: sybaseServer
sybaseVersion: 1500
sybaseServername: foobar
sybaseService: ASE
sybaseStatus: 4
sybaseAddress: TCP#1#foobar 5000
sybaseRetryCount: 12
sybaseRetryDelay: 30
sybaseHAServernam: secondary
```

All entries in the LDAP directory service are called entities. Each entity has a distinguished name (DN) and is stored in a hierarchical tree structure based on its DN. This tree is called the **directory information tree** (DIT). Client applications use a DIT base to specify where entities are stored. See "The libtcl*.cfg file" on page 43.

In the example above, the entry describes an Adaptive Server named "foobar" listening on a TCP connection with a port number of 5000. The value 1, located between `TCP` and `5000`, means the entry is used for both QUERY and MASTER entries. This value should always be 1 for an LDAP directory service. This entity also specifies a retry count of 12 (times) and a retry delay of 30 (seconds). Once a client has found an address where a server responds, the login dialog between the client and the server begins.

You can find a complete list of Sybase's LDAP directory schema in UNIX *$SYBASE/$SYBASE_OCS/config*.

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

Since LDAP supports multiple entries for each attribute, each address attribute must contain the address of a single server, including protocol, access type, and address. See sybaseAddress in Table 3-2.

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

```
sybaseAddress = TCP#1#TOEJAM 4444
```

```
sybaseAddress = NAMEPIPE#1#\pipe\sybase\query
```

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

You can edit these entries with dsedit. See "Adding a server to the directory services" on page 45.

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

## The *libtcl*.cfg* file

You use the *libtcl*.cfg* file to specify the LDAP server name, port number, DIT base, user name, and password to authenticate the connection to an LDAP server.

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

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

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

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

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

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

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

where the *ldapurl* is defined as:

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

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

```
ldap=libsybdldap.dll ldap://seashore/d=sybase,dc=com
```

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

## Enabling LDAP directory services

To use a directory service, you must:

1 Configure the LDAP server according to the vendor-supplied documentation.

2 Add the location of the LDAP libraries to the Unix load library path environment variable for your platform.

3 Configure the *libtcl.cfg* file to use directory services.

Use any standard ASCII text editor to:

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

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

```
file libtcl.cfg:
ldap=libsybdldap.so ldap://host:port/ditbase??scope??bindname=
username?password

file libtcl64.cfg
ldap=libsybdldap64.so
ldap://host:port/ditbase??scope??bindname=username?password
```

For example:

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

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

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

**Table 3-3: ldapurl variables**

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

4 Verify that the appropriate environment variable points to the required third-party libraries. The Netscape LDAP SDK libraries are located in *$SYBASE/$SYBASE_OCS/lib3p* or *lib3p64*.The Unix load library path environment variable must point to this directory.

5 Add your server entry to the LDAP server using dscp or dsedit. See "Adding a server to the directory services" on page 45.

## Adding a server to the directory services

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

Each server entry is made up of a set of attributes. When you add or modify a server entry, you are prompted for information about server attributes. Some attributes are provided by default, others require user input. When a default value is provided, it appears in brackets "[ ]". See Table 3-2 on page 41 for accepted values.

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

Before you can add, delete, or modify an LDAP server entry, you must add the LDAP URL to the *libtcl.cfg* file. See "The libtcl*.cfg file" on page 43.

Use dsedit to add a server to the directory service:

1 Source *SYBASE.csh* or *SYBASE.sh* to set the environment variables.

2 cd to *$SYBASE/$SYBASE_OCS/bin*.

3 Execute dsedit.

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

5 Click Add New Server Entry.

6 Enter:

- The server name – this is required.
- The security mechanism – optional. This is the name of the high-availability failover server, if you have one.

7 Click Add New Network Transport and:

- Select the transport type.
- Enter the host name.
- Enter the port number.
- (Optional) enter the SSL filter string.

8 Click OK two times to edit dsedit.

To view the server entries, enter the following URL in Netscape, `http://host:port/ditbase??one`.

For example:

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

**Note** Microsoft Internet Explorer does not recognize LDAP URLs.

For more information about dscp, see the *Open Client/Server Configuration Guide*, in the 11.1.x Generic Collection at http://www.sybase.com/support/manuals.

## Multiple directory services

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

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

For example:

```
[DIRECTORY]
ldap=libsybdldap.so ldap://test:389/dc=sybase,dc=com
ldap=libsybdldap.so ldap://huey:11389/dc=sybase,dc=com
```

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

**Note** For more information, see the *Open Client Client-Library/C Programmer's Guide* and the *Open Client Client-Library/C Reference Manual* at http://www.sybase.com/support/manuals.

## Encrypting the password

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

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

```
bin/pwdcrypt
```

Enter your password twice when prompted.

pwdcrypt generates an encrypted password. For example:

```
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706
```

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

```
ldap=libsybdldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,
dc=com?password
```

Replace the password with the encrypted string:

```
ldap=libsybdldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,dc=com?
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706
```

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

## Performance

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

## Migrating from the *interfaces* file to LDAP

There is no direct method to upgrade an existing server using the *interfaces* file to one that uses lightweight directory services. To upgrade an earlier version of Adaptive Server to Adaptive Server version 15.0, see the *Installation Guide for Windows*.

Once you have upgraded the server, you can configure your server to use LDAP service.

1. Shut down the server. For information about starting and stopping clusters, see the *Users Guide to Clusters.* For shared memory servers, see the configuration guide for Adaptive Server 15.0, Chapter 2, "Starting and Stopping Servers."
2. Edit the *$SYBASE/$SYBASE_OCS/config/libtcl.cfg* or *libtcl64.cfg* file to add the directory service. See "Enabling LDAP directory services" on page 44.
3. Use dsedit or dscp to add the server, cluster, and instance entries for clustered servers to the directory service. See "Adding a server to the directory services" on page 45.
4. Use qrmutil to verify that the interfaces directory attributes in the cluster quorum file is empty for the cluster and instance definitions. For example, to show the values in the quorum file, enter:

```
$SYBASE/ASE-15_0/qrmutil --quorum_dev=path_to_your_quorum --
display=config
```

   You must reset the value if a path is defined for the *interface_dir* attribute on any instance, or for the cluster. Specifying this attribute with a path value forces the instance to use the interfaces file and overrides the information in the *libtcl.cfg* and *libtcl64.cfg* files.

For example, use these qrmutil commands to reset the value of the *interface_dir* attributes The value for *interface_dir* is two single quotes, meaning an empty string.

```
$SYBASE/ASE-15_0/bin/qrmutil --quorum_dev=path_to_your_quorum --
interface_dir=''

$SYBASE/ASE-15_0/bin/qrmutil --quorum_dev=path_to_your_quorum --
instance=name_of_instance_to_reconfig --interface_dir=''
```

5 Restart your server or cluster.

## Using LDAP directory services with the shared-disk cluster

The Cluster Edition can use LDAP directory services to specify its cluster and instance entries. You must specify an empty string for the *interface_dir* attribute in the cluster's quorum file. Do not use the dataserver parameter -i *interfaces_path* to specify the path to the interfaces file.

If you do not specify a value for *interface_dir*, the Cluster Edition uses the ordered list of directory services defined in the *$SYBASE/OCS-15_0/config/libtcl64.cfg* (for 64-bit servers and clients), or *$SYBASE/OCS-15_0/config/libtcl.cfg* (for 32-bit servers and clients). After the server searches the directory services defined in *libtcl64.cfg*, the interfaces file in the default location is searched.

Open Client applications can use LDAP directory service to store cluster and instance server entries. For example, for a cluster named "mycluster" with two instances ("ase1" and "ase2"), the interfaces looks like:

```
ase1

     master tcp ether blade1 10945
     query tcp ether blade1 10945
ase2

     master tcp ether blade2  10955
     query tcp ether blade2  10955
mycluster

     query tcp ether blade1 10945
     query tcp ether blade2 10955
```

You must use dsedit or dscp to add equivalent LDAP directory service entries for the server names "ase", "ase2", and "mycluster" to the LDAP directory service. See "Adding a server to the directory services" on page 45. For more information about dsedit and dscp, see the *Utility Guide*.

Clients can connect to any instance in the cluster using the cluster name (in this example, “mycluster”) or an instance-specific server name (“ase1” or “ase2”).

When SSL is used for clients to connect to a nonclustered Adaptive Server using SSL, the SSL filter is placed after the port number in the interfaces file. The directory service includes the common name, which you added with dsedit or from hand-editing. Typically, one SSL certificate with one common name is used for the entire cluster, rather than one for each instance. See “Using SSL in a clustered environment” on page 37.

This example adds the SSL filter to an interfaces file entry for the cluster “mycluster:”

```
mycluster

       query tcp ether blade1 10945 ssl="cn=mycluster.domain.com"
       query tcp ether blade2 10955 ssl="cn=mycluster.domain.com"
```

Entries added to an LDAP directory service must specify the common name with the SSL filter, `ssl="cn=mycluster.domain.com"`.

For example, this dscp session adds the example entry above for cluster “mycluster:”

```
%  dscp

>> open ldap

ok

Session 1 ldap>> add mycluster
Service: [ASE]
Transport Type: [tcp]
Transport Address: blade1 10945 ssl="cn=mycluster.domain.com"
Transport Type: [tcp]
Transport Address: blade2 10955 ssl="cn=mycluster.domain.com"
Transport Type: [tcp]
Transport Address:
Security Mechanism [] :
HA Failoverserver:
Retry Count:
Retry Delay:
Added mycluster

Session 1 ldap>> read mycluster

DIT base for object:  dc=domain,dc=com
Distinguish name: sybaseServername=mycluster, dc=domain,dc=com
```

```
Server Entry Version: 15001
Server Name: mycluster
Server Service: ASE
Server Status: 4 (Unknown)
Server Address:
Transport Type: tcp
Transport Address: yellowstar 2521 ssl="cn=mycluster.domain.com"
Transport Type: tcp
Transport Address: yellowstar 2525 ssl="cn=mycluster.domain.com"

Session 1 ldap>> quit
```

CHAPTER 4

# Using Monitoring Tables in a Clustered Environment

This chapter describes monitor tables for the Cluster Edition, and how to configure and manage them for the Cluster Edition.



## Changes for clusters

In a clustered environment, monitoring tables report on a per-instance basis instead of returning a cluster-wide result. This allows you to monitor the activities of processes and queries across the cluster to get a better understanding of the statistics for objects that may be opened on more than one instance and resource usage on each instance in the cluster. For example, if you query the monitoring tables about a table, this table may be opened or accessed by more than one instance in the cluster, so the descriptors for this table—and the associated statistics—may be in memory on the instance. Statistics are not aggregated for the cluster. The statistical results for all instances are returned as a unioned result set with rows collected from each instance. Each instance is identified in the result set with a row in the InstanceID column.

## Configuring the system view

The system_view is a session-specific setting that allows you to control the scope of monitoring data that queries return from the monitoring tables, sysprocesses, sp_who, and other commands. When you set system_view to cluster, queries on the monitoring tables return data from all active instances in the cluster. When you set system_view to instance, queries on the monitoring tables return data only for processes or objects that are active on the instance to which the client is connected.

Use the set command to configure the scope of the session:

```
set system_view {instance | cluster | clear}
```

where:

- instance – returns statistics for the local instance only. Cross-cluster requests are not sent to any other instance in the cluster.
- cluster – returns statistics for all instances in the cluster.
- clear – returns the system view to the configured default.

This example modifies the session settings so queries on the monitoring tables return data only for the instance to which the client is connected:

```
set system_view instance
```

This example modifies the session settings so queries on the monitoring tables return data for the cluster:

```
set system_view cluster
```

This example clears the current setting of the system view and returns system view to the default setting.

```
set system_view clear
```

If you do not specify an InstanceID when you query a monitoring table or call a monitoring table RPC, the instance uses the current system_view configuration.

The session system view is inherited from its host logical cluster. selecting the *@@system_view* global variable to determine the current system view.

## Configuring monitoring tables

Use configuration parameters for the monitoring tables to determine their cluster-wide or instance-only behavior. By default, all monitoring table configuration values are applied cluster-wide. See “sp_configure” on page 282 for more information.

## Managing the message pipe

These parameters determine how the cluster and instances manages the memory used to store the data for the historical monitoring tables:

- deadlock pipe max messages
- errorlog pipe max messages
- sql text pipe max messages
- plan text pipe max messages
- statement pipe max messages

You can configure these parameters globally for the cluster and individually for each instance. These parameters allocate memory for the pipe. An instance can dynamically add memory to the pipe but cannot dynamically remove memory from the pipe, so if you reduce the size of the parameter, you must restart the instance for the new pipe size to take effect.

Below are some algorithms for determining the size for the parameters.

- For an individual instance, the memory required for the each pipe configuration is:

  *configuration_value* X *number_of_engines*

- To globally set the memory for each pipe configuration:

  *configuration_value* X *number_of_engines* X *number_of_instances*

- If you have set the value for pipe configurations differently for each instance, then the amount of memory required for the cluster is:

  (*instance_1_value* X *number_of_engines*) + (*instance_2_value* X *number_of_engines*) +. . . + (*instance_n_value* X *number_of_engines*))

## Changes for RPCs

If you invoke an RPC but do not include the InstanceID as a parameter, the monitoring tables use the system view setting to determine how to report the statistics. If you have set the system view setting to instance, all data gathering is local. If you have set the system view setting to cluster, the monitoring tables communicate with instances forming the all instances in the cluster, not the logical cluster.

## InstanceID added to monitor instances

Table 4-1 describes monitoring tables to which the Cluster Edition adds the InstanceID column.

***Table 4-1: monitoring tables with InstanceID column***

| | |
|---|---|
| monCachePool | monDataCache |
| monCachedProcedures | monDeviceIO |
| monDeadLock | monErrorLog |
| monEngine | monIOQueue |
| monLicense | monLocks |
| monOpenDatabases | monNetworkIO |
| monOpenPartitionActivity | monOpenObjectActivity |
| monProcess | monProcedureCache |
| moProcessLookup | monProcessActivity |
| monProcessObject | monProcessNetIO |
| monProcessSQLText | monProcessProcedures |
| monProcessWaits | monProcessStatement |
| monResourceUsage | monProcessWorkerThread |
| monSysPlanText | monState |
| monSysStatement | monSysSQLText |
| monSysWorkerThread | monSysWaits |
| monCachedObject | |

Table 4-2 describes monitoring tables that return identical information for all instances.

***Table 4-2: monitoring tables that include the same information for all instances***

| Table name | Description |
|---|---|
| monMon | Metadata view is identical on all instances. |
| monTableColumns | Metadata view is identical on all instances. |
| monTableParameters | Metadata view is identical on all instances. |
| monTables | Metadata view is identical on all instances. |
| monWaitClassInfo | List of descriptions is identical on all instances. |
| monWaitEventInfo | List of descriptions is identical on all instances. |

# New and changed tables for the cluster cache manager

## New tables

The Cluster Edition adds the monClusterCacheManager for statistical analysis for the cluster cache manager.

### monClusterCacheManager

Description

monClusterCachManager provides diagnostic information about the cluster cache manager daemon running on each instance. monClusterCacheManager reports cluster-wide information on a per-instance basis.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster. |
| DaemonName | varchar | Name of the cluster cache manager daemon. |
| RequestsQueued | int | Number of requests queued to the cluster cache manager daemon. |
| RequestsRequeued | int | Number of requests requeued to the cluster cache manager daemon. |
| RequestsServiced | int | Number of requests serviced by the cluster cache manager daemon. |
| TransfersInitiated | int | Number of transfers initiated by the cluster cache manager daemon. |
| Downgrades | int | Number of downgrades performed by the cluster cache manager daemon. |

| Name | Datatype | Description |
|---|---|---|
| Releases | int | Number of releases performed by the cluster cache manager daemon. |
| DiskWrites | int | Number of disk writes initiated by the cluster cache manager daemon. |
| SleepCount | int | Number of times the cluster cache manager daemon went to sleep. |
| AvgServiceTime | int | Average time (in milliseconds) spent servicing a request. |
| MaxQSize | int | Maximum number of requests queued to the cluster cache manager daemon at any time since the instance started. |

Required parameters None

### monConnectionMigration

Description monConnection Migration provides information about the connection currently migrating. monConnectionMigration reports cluster-wide information on a per-instance basis across the cluster.

Columns

| Name | Datatype | Description |
|---|---|---|
| PendingSPID | int | SPID of the pending migration session process. |
| LogicalCluster | varchar | Current logical cluster. |
| Instance | varchard | Current instance. |
| InstanceMigrationLogicalCluster | varchar | Migrating logical cluster. |
| MigrationInstance | varchar | Migrating instance. |
| Command | varchar | Migration trigger. |

Required parameters None

## Changed tables

### monOpenObjectActivity and monOpenPartitionActivity

monOpenObjectActivity and monOpenPartitionActivity add these columns for analyzing lock acquisitions.

| Name | Datatype | Description |
|---|---|---|
| PhsyicalLocks | int | Number of physical locks requested per object. |
| PhsycialLocksRetained | int | Number of physical locks retained. You can use this to identify the lock hit ratio for each object. Good hit ratios imply balanced partitioning for this object. |
| PhysicalLocksDeadlocks | int | Number of times a physical lock requested returned a deadlock. The Cluster Physical Locks subsection of sp_sysmon uses this counter to report deadlocks while acquiring physical locks for each object. |
| PhysicalLocksWaited | int | Number of times an instance waits for a physical lock request. |
| PhysicalLocksPageTransfer | int | Number of page transfers that occurred when an instance requests a physical lock. The Cluster Physical Locks subsection of sp_sysmon uses this counter to report the node-to-node transfer and physical-lock acquisition as a node affinity ratio for this object. |
| PhysicalLocksRetainWaited | int | Number of physical lock requests waiting before a lock is retained. |
| TransferReqWaited | int | Number of times physical lock requests waiting before receiving page transfers. |
| AvgPhysicalLocksWaitTime | int | The average amount of time clients spend before the physical lock is granted. |
| AvgTransferReqWaitTime | int | The average amount of time physical lock requests wait before receiving page transfers. |
| TotalServiceRequests | int | Number of physical lock requests serviced by the Cluster Cache Manager of an instance. |
| PhysicalLocksDowngraded | int | Number of physical lock downgrade requests serviced by the Cluster Cache Manager of an instance. |
| PagesTransferred | int | Number of pages transferred at an instance by the Cluster Cache Manager. |
| ClusterPageWrites | int | Number of pages written to disk by the Cluster Cache Manager of an instance. |

| Name | Datatype | Description |
|---|---|---|
| AvgServiceTime | int | The average amount of service time spent by the Cluster Cache Manager of an instance. |
| AvgTimeWaitedOnLocalUsers | int | The average amount of service time an instance's Cluster Cache Manager waits due to page use by users on this instance. |
| AvgTransferSendWaitTime | int | The average amount of service time an instance's Cluster Cache Manager spends for page transfer. |
| AvgIOServiceTime | int | The average amount of service time used by an instance's Cluster Cache Manager for page transfer. |
| AvgDowngradeServiceTime | int | The average amount of service time the Cluster Cache Manager uses to downgrade physical locks. |

### monCachedProcedures

monCachedProcedures adds these columns for the Cluster Edition.

| Name | Datatype | Description |
|---|---|---|
| RequestCnt | int | Number of times this procedure was requested from cache |
| TempdbRemapCnt | int | Number of times this procedure was remapped for the temporary database's ID. |
| AvgTempdbRemapTime | int | Average time (in milliseconds) spent remapping the temporary databases's ID. |

# Monitoring tables added for CIPC

The monCIPC, monCIPCmesh, and monCIPCEndpoints tables monitor the cluster interprocess communication protocol (CIPC). This protocol is used to pass messages between members of a shared-disk cluster.

## monCIPC

Description

monCIPC gives summary figures for total messaging within the cluster, as viewed from the current instance or all instances.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster. |
| ReceivedCount | int | Number of messages received by this instance. |
| TransmitCount | int | Number of messages sent by this instance. |
| Multicast | int | Number of messages sent that were addressed to all other instances in the cluster. |
| Synchronous | int | Number of those messages sent synchronously. |
| ReceiveSoftErrors | int | Number of recoverable errors received on this instance. |
| ReceiveHardErrors | int | Number of unrecoverable errors received on this instance. |
| TransmitsSoftErrors | int | Number of recoverable transmit errors on this instance. |
| TransmitHardErrors | int | Number of unrecoverable transmit errors on this instance. |
| Retransmits | int | Number of retransmissions performed by this instance. |
| Switches | int | Number of switches between the primary interconnect network and the secondary interconnect network. |
| FailedSwitches | int | Number of attempts to switch between primary and secondary interconnect networks that have failed. |
| RegularBuffersInUse | int | Number of buffers from the CIPC regular buffer pool currently allocated. |
| FreeRegularBuffers | int | Number of buffers available in the CIPC regular buffer pool. |
| MaxRegularBuffersInUse | int | Maximum number of buffers from the CIPC regular buffer pool allocated at any time since the server was started. |
| LargeBuffersInUse | int | Number of buffers from the CIPC large buffer pool currently allocated. |
| FreeLargeBuffers | int | Number of buffers available in the CIPC large buffer pool. |

| Name | Datatype | Description |
|---|---|---|
| MaxLargeBuffersInUse | int | Maximum number of buffers from the CIPC large buffer pool allocated at any time since the server was started. |

One row is returned in the monCIPC table for each instance in the cluster, if the system view is set to cluster; otherwise, a single row is returned.

Required parameters

None

## monCIPCEndpoints

Description

monCIPCEndpoints provides a detailed summary, giving traffic data for each bsystem within the cluster instance.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster |
| ReceivedCount | int | Number of messages received by this logical endpoint within the cluster |
| TransmitCount | int | Number of messages sent by this logical endpoint within the instance |
| ReceiveBytes | int | Number of bytes received by this logical endpoint within the instance |
| TransmitBytes | int | Number of bytes sent by this logical endpoint within the instance |
| ReceiveQ | int | Current number of messages queued for this logical endpoint |
| MaxReceiveQ | int | Maximum number of messages ever observed queued for this logical endpoint |
| DoneQ | int | Current number of messages for this logical endpoint, which have been processed and await further action |
| MaxDoneQ | int | Maximum number of messages ever observed for this logical endpoint, which have been processed and await further action |
| EndPoint | varchar | Name of CIPC endpoint |
| MaxRecvQTime | float | Maximum time (in milliseconds) a message spends in the queues of the current logical end point. |

| Name | Datatype | Description |
|---|---|---|
| AvgRecvQTime | float | Average time (in milliseconds) a message spends in the queues of the current logical end point. |

One row is returned for each logical endpoint in the instance. If the system view is set to cluster, a set of rows is returned for each node in the cluster.

Required parameters None

## monCIPCLinks

Description monCIPCLinks monitors the state of the links between instances in the cluster.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster. |
| LocalInterface | varchar30 | The name of the link's local network end point. Same name that appears in the *hosts* file for a server name. |
| Remote Interface | varchar30 | The name of the link's remote end point. Same name that appears in the *hosts* file for a server name. |
| PassiveState | varchar10 | The latest state listed in the traffic on the link. |
| PassiveStateAge | int | Time since the PassiveState column was updated, in milliseconds. |
| ActiveState | varchar10 | The latest state used as determined by active monitoring (when no traffic was present on the link). |
| ActiveStateAge | int | Time since the ActiveState column was updated, in milliseconds.. |

Required Parameters None.

## monCIPCmesh

Description monCIPCmesh gives summary figures for the mesh of connections, from the current instance to all other instances in the cluster, on a per instance basis.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster. |

| Name | Datatype | Description |
|---|---|---|
| FarInstanceID | tinyint | Instance number of the far-end instance in the cluster. |
| Received | int | Number of messages received by this instance from the FarInstanceID instance. |
| Dropped | int | Number of messages from the FarInstanceID instance that have been dropped, due to some lack of resource. |
| Transmitted | int | Number of messages transmitted to the FarInstanceID instance. |
| Resent | int | Number of messages re-sent to the FarInstanceID instance. |
| Retry | int | Number of packets retried to the FarInstanceID instance. |
| SendQ | int | Current number of messages waiting to be sent to the FarInstanceID instance. |
| MaxSendQ | int | Maximum observed number of messages awaiting transmission to the FarInstanceID node. |
| SentQ | int | Current number of messages sent but where notification of sending has not been processed. |
| MaxSentQ | int | Maximum number of messages sent, but notification of sending is not yet processed. |
| MaxSendQTime | float | Maximum amount of time (in milliseconds) spent by a message in the physical end point queues of the current channel. |
| AvgSendQTime | float | Average amount of time (in milliseconds) spent by a message in the physical end point queues of the current channel. |
| Mesh | varchar | The channel name for the connection. One of:<br>• Out of Band<br>• Message<br>• Large Message<br>• DMA |
| MinRTT | int | Minimum round trip delay observed for messages (applies only to UDP transport). |
| MaxRTT | int | Maximum round trip delay observed for messages (applies only to UDP transport). |
| AverageRTT | int | Average round trip delay observed for messages (applies only to UDP transport). |

One row is returned for each of the four connections to each of the other nodes in the cluster, up to the maximum configured. If the system view is cluster, a set of rows for each instance active in the cluster is returned.

Required parameters

None

# Monitoring tables added for temporary databases

## monTempdbActivity

Description

Provides statistics for all open local temporary databases, including global system tempdb when the instance is started in tempdb configuration mode. You must enable the object lockwait timing, enabling monitoring, and per object statistics active configuration parameters before monTempdbActivity can gather statistics.

Columns

| Name | Datatype | Description |
|---|---|---|
| DBID | int | Unique identifier for the database |
| InstanceID | tinyint | ID of the instance within the cluster |
| DBName | varchar(30) | Name of the database |
| AppendLogRequest | int | Number of semaphore requests from an instance attempting to append to the database transaction log |
| AppendLogWaits | int | Number of times a task waits for the append log semaphore to be granted |
| LogicalReads | int | Total number of buffers read |
| PhysicalReads | int | Number of buffers read from disk |
| APFReads | int | Number of asynchronous prefetch (APF) buffers read |
| PagesRead | int | Total number of pages read |
| PhysicalWrites | int | Total number of buffers written to disk |
| PagesWritten | int | Total number of pages written to disk |
| LockRequests | int | Number of requests for a lock on the object |
| LockWaits | int | Number of times a task waited for a lock for the object |

| Name | Datatype | Description |
|---|---|---|
| CatLockRequests | int | Number of requests for a lock on the system catalog |
| CatLockWaits | int | Number of times a task waited for a lock for the system table |
| AssignedCnt | int | Number of times this temporary database was assigned to a user task |
| SharableTabCnt | int | Number of shareable tables created |

Required parameters

- enable monitoring
- per object statistics active
- object lockwait timing

# New and changed tables added for general statistics

## New tables

The Cluster Edition adds the monSysLoad table to provide statistical information about each engine.

### monSysLoad

Description

Provides trended statistics on a per-engine basis.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster. |
| EngineNumber | smallint | Engine to which this row belongs. |
| SteadyState | real | Average value for this statistic since start-up. |
| Avg_1min | real | One-minute moving average for this statistic. |
| Avg_5min | real | Five-minute moving average for this statistic. |
| Avg_15min | real | Fifteen-minute moving average for this statistic. |

| Name | Datatype | Description |
|---|---|---|
| Max_1min | real | Maximum one-minute average since start-up. |
| Max_5min | real | Maximum five-minute average since start-up. |
| Max_15min | real | Maximum fifteen-minute average since start-up. |
| Max_1min_Time | datetime | *datetime* at which Max_1min occurred. |
| Max_5min_Time | datetime | *datetime* at which Max_5min occurred. |
| Max_15min_Time | datetime | *datetime* at which Max_15min occurred. |
| Statistic | varchar(30) | Name of the statistic this row represents. |
| Sample | float | Value of the metric at the last sample interval (that is, the "current" value of the metric). |
| Peak | float | The highest Sample value since the instance started (that is, the peak Sample value). |
| Peak_time | datetime | The date and time the Peak value was achieved |
| StatisticID | int | A fixed identifier for this statistic. You may wish to write applications to the fixed StatisticID instead of the localized Statistic name. |

There is one row per engine per statistic, with the exception of kernel run queue length, which is reported only for engine number 0.

Averages are computed using the "exponential moving average" technique, not geometric averaging. This makes averages values more resistant to momentary peaks and valleys.

You need not enable any monitoring configuration parameters to use monSysLoad. These statics are always available and do not conflict with other monitoring tools, such as sp_sysmon.

Required parameters

None

## Changed tables

### monEngine

The Cluster Editions adds a new column to the monEngine table.

| Name | Datatype | Description |
|---|---|---|
| IOCPUTime | int | The amount of time (in seconds) the engine has been waiting for issued IOs to complete. |

# Monitoring tables added for workload manager

The workload manager adds these monitoring tables for information reporting.

## monLogicalCluster

Description

Displays information about the logical clusters currently configured on the system.

Columns

| Name | Datatype | Description |
|---|---|---|
| ActionRelease | varchar(20) | Current action release mode for the logical cluster |
| LCID | int | Logical cluster ID. |
| Name | varchar(30) | Logical cluster name. |
| State | varchar(20) | Current state. One of:<br>• online<br>• offline<br>• failed<br>• inactive<br>• time_wait |
| DownRoutingMode | varchar(20) | Down routing-mode setting. One of:<br>• system<br>• open<br>• disconnect |

| Name | Datatype | Description |
|---|---|---|
| FailoverMode | varchar(20) | Failover mode setting, `instance` or `cluster`). |
| Attributes | int | Bitmask of logical cluster attributes. |
| Mode | varchar(20) | Start-up mode setting, `automatic` or `manual`. |
| SystemView | varchar(20) | System view setting, `instance` or `cluster`. |
| Roles | varchar(20) | Comma-delimited list of special roles for this logical cluster. The “system” logical cluster always has the system role. The open logical cluster has the “open” role. If the system logical cluster also has the open role, the value for this column is `system, open`. Logical clusters without any special roles return a NULL value. |
| LoadProfile | varchar(30) | Load profile associated with this logical cluster. |
| ActiveConnections | int | Number of active connections using this logical cluster. |
| BaseInstances | tinyint | Number of instances configured as base instances for this logical cluster. |
| ActiveBaseInstances | tinyint | Number of base instances on which this logical cluster is currently active. |
| FailoverInstances | tinyint | Number of instances configured as failover instances for this logical cluster. |
| ActionRelease | varchar(30) | Current action release mode for the logical cluster.. |
| ActiveFailoverInstances | tinyint | Number of failover instances on which this logical cluster is currently active. |

Required parameters None

## monLogicalClusterInstance

Description Displays information about the many-to-many relationship between instances and logical clusters.

Columns

| Name | Datatype | Description |
|---|---|---|
| LCID | int | Logical cluster ID |
| LogicalClusterName | varchar(30) | Logical cluster name |
| InstanceID | tinyint | ID of the instance within the cluster |
| InstanceName | varchar(30) | Instance name |
| Type | varchar(20) | Instance type |
| FailoverGroup | tinyint | Failover group of which this instance is a member (failover instances only) |
| State | varchar(20) | State of this instance with respect to the logical cluster |
| ActiveConnections | int | Number of active connections for this logical cluster on this instance |
| NonMigConnections | int | Number of active connections that do not support the connection migration protocol |
| NonHAConnections | int | Number of active connections that do not support the high availability failover protocol |
| LoadScore | real | Workload score for this instance using the load profile associated with its logical cluster |

Required parameters None

## monLogicalClusterRoute

Description Displays information about the configured routes (application, login, and alias bindings).

Columns

| Name | Datatype | Description |
|---|---|---|
| LCID | int | Logical cluster ID |
| LogicalClusterName | varchar(30) | Logical cluster name |
| RouteType | varchar(20) | Route type, one of `application`, `login`, or `alias` |
| RouteKey | varchar(30) | Application, login, or alias name associated with this route. |

Required parameters None

## monLogicalClusterAction

Description

Shows any actions currently running against logical clusters.

Columns

| Name | Datatype | Description |
|---|---|---|
| Handle | int | Unique handle you can use to cancel this action. |
| State | varchar(20) | State of the action, `active` or `complete`. |
| LCID | int | Logical cluster ID to which this action applies. |
| LogicalClusterName | varchar(30) | Logical cluster name of this logical cluster (denormalized to reduce joins). |
| Action | varchar(15) | Action being performed. A combination of the command running and its scope. For example, offline instance or failover cluster. |
| FromInstances | varchar(96) | A comma-separated list of `from instances` for this command and action (instance being brought offline). |
| ToInstances | varchar(96) | A comma-separated list of `to instances` for this command and action (instances being brought online). |
| InstancesWaiting | int | Number of instances waiting to go offline (this is a count of FromInstances that are in the `time_wait` state). |
| WaitType | varchar(20) | Current wait state for this action (one of `wait`, `until`, or `nowait`). |
| StartTime | datetime | Date and time the command was issued. |
| Deadline | datetime | Date and time the command must be finished (based on the time value supplied to the `wait` or `until` options). |
| CompleteTime | datetime | Date and time the command and action completed (when InstancesWaiting is zero and the action went from `active` to the `complete` state). Returns NULL for non-complete actions. |
| ConnectionsRemaining | int | Number of connections remaining to move as a result of this command. |

| Name | Datatype | Description |
|---|---|---|
| NonMigConnections | int | Number of connections to be terminated because they do not support the migration protocol. |
| NonAHConnections | int | Number of connections that do not support the high availability failover protocol. These connections are disconnected and cannot fail over when the command finalizes. |

Required parameters None

## monProcessMigration

Description Displays information about the connection currently migrating.

Columns

| Name | Datatype | Description |
|---|---|---|
| SPID | int | Pending migration session process |
| KPID | int | Kernel process identifier |
| Logical cluster | varchar(30) | Current logical cluster |
| Instance | varchar(30) | Current instance. |
| InstanceMigrationLogicalCluster | varchar(30) | Migration logical cluster. |
| MigrationInstance | varchar(30) | Migration instance. |
| Command | varchar(20) | Migration trigger. |

Required parameters None

## monWorkloadProfile

Description Displays currently configured workload profiles.

Columns

| Name | Datatype | Description |
|---|---|---|
| ProfileID | smallint | Workload profile ID |
| Name | varchar(30) | Workload profile name |
| RefreshInterval | varchar(15) | Frequency of load refresh, can be a time in the form *hh:mm:ss* or `DISABLED` or `AUTO` |

| Name | Datatype | Description |
|---|---|---|
| ConnectionsWeight | tinyint | Weight associated with the `active connections` metric |
| CpuWeight | tinyint | Weight associated with the `cpu utilization` metric |
| RunQueueWeight | tinyint | Weight associated with the `run queue` metric |
| IoLoadWeight | tinyint | Weight associated with the `io load` metric |
| EngineWeight | tinyint | Weight associated with the `engine deficit` metric |
| UserWeight | tinyint | Weight associated with the `user metric` metric |
| LoginWeight | smallint | Threshold for load-based login redirection |
| Dynamicthreshold | smallint | Threshold for dynamic load distribution (that is, post-login migration for load purposes) |

Required parameters

None

## monWorkload

Description

Displays the workload score for each logical cluster on each instance according to its load profile.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster |
| InstanceName | varchar(30) | Instance name |
| LCID | tinyint | Logical cluster ID |
| LogicalClusterName | varchar(30) | Logical cluster name |
| LoadProfile | varchar(30) | Name of the load profile used to generate the load score |
| LoadScore | int | Load score for this instance or logical cluster |
| ConnectionsScore | float | Weighted value for the `user connections` metric |
| CpuScore | float | Weighted value for the `cpu utilization` metric |
| RunQueueScore | float | Weighted value for the `run queue` metric |
| IoLoadScore | float | Weighted value for the `io load` metric |

| Name | Datatype | Description |
|---|---|---|
| EngineScore | float | Weighted value for the `engine deficit` metric |
| UserScore | float | Weighted value for the `user` metric |

Required parameters None

## monWorkloadRaw

Description Provides the raw workload statistics for each instance.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster |
| InstanceName | varchar(30) | Instance name |
| ConnectionsRaw | float | Raw value for the `user connections` metric |
| CpuRaw | float | Raw value for the `cpu utilization` metric |
| RunQueueRaw | float | Raw value for the `run queue` metric |
| IoLoadRaw | float | Raw value for the `io load` metric |
| EngineRaw | float | Raw value for the `engine deficit` metric |
| UserRaw | float | Raw value for the `user` metric |

Required parameters None

## monWorkloadPreview

Description Provides an estimate of how a load profile impacts the workload score without enabling the profile. monWorkloadpreview includes one row for each logical cluster and instance on which this logical cluster is running. The load score and components are based on the current profile for that logical cluster. The monWorkloadPreview table has one row for each combination of instance and load profile configured on the system, allowing the administrator to see how workload scoring would be done for each profile.

Columns

| Name | Datatype | Description |
|---|---|---|
| InstanceID | tinyint | ID of the instance within the cluster |

| Name | Datatype | Description |
|---|---|---|
| InstanceName | varchar(30) | Instance name |
| LoadProfileID | smallint | Load profile ID |
| LoadProfile | varchar(30) | Name of load profile used to generate the load score |
| LoadScore | int | Load score for this instance or logical cluster |
| ConnectionScore | float | Weighted value for the `user connections` metric |
| CpuScore | float | Weighted value for the `cpu utilization` metric |
| RunQueueScore | float | Weighted value for the `run queue` metric |
| IoLoadScore | float | Weighted value for the `io load` metric |
| EngineScore | float | Weighted value for the `engine deficit` metric |
| UserScore | float | Weighted value for the `user` metric |

Required parameters

None

CHAPTER 5

# Managing the Workload

This chapter describes how to manage the workload and provide failover for applications accessing the Cluster Edition.



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

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

By creating logical clusters on top of the physical cluster, you can partition applications with different workloads on the same system. The workload is managed at the application level, which means that you can manage incoming connections, failover policies, load distribution strategies, and planned downtime according to how each of your applications use the system.

The system administrator manages the workload using the Adaptive Server plug-in to Sybase Central or the command-line options described in this chapter. The system administrator:

- Configures and manages a logical cluster. This includes creating and dropping logical clusters, adding or removing instances from the cluster, modifying failover rules, starting and stopping the cluster or instances in the cluster, configuring routing rules, and so on.
- Selects or configures load profiles that the system uses to determine the current relative workload.
- Monitors instances in the cluster and the workload on each instance.

# Logical cluster resources

A logical cluster is assigned resources from the physical cluster:

- Instances – are a logical cluster's **base instances**, which means that they are started when a logical cluster starts and that a failback restores them.
- Failover resources – are an ordered list of instances on which a logical cluster is to run should 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.

# The system logical cluster

When you create a shared-disk cluster, Adaptive Server automatically creates a system logical cluster for it. The system logical cluster provides a logical cluster representation for daemons and enables management of certain tasks. It represents the physical cluster, contains all the instances of the physical cluster, and has the same name as the physical cluster. All background tasks, such as checkpoint and housekeeper, run on the system logical cluster. Special rules apply to the system logical cluster. Upon creation, this new system logical cluster is granted the open property, which means that all unbound connections are routed to it.

System administrators typically do not interact with the system logical cluster. However, the following do apply to the system logical cluster:

- Routing rules. For example, you can route logins used by system administrators to the system logical cluster.
- The open property.
- System view setting.
- The load profile.
- Login distribution mode.

The following do not apply to the system logical cluster:

- Failover resources and rules
- Commands that:
    - Create or drop resources of a logical cluster
    - Migrate, fail over, and fail back instances
    - Change a logical cluster or instance state
    - Change failover mode settings
    - Change start-up mode settings
    - Set the down routing mode

# Setting up a logical cluster

There are many possible options for configuring a logical cluster; the basic steps for setting up a working logical cluster are:

1 Create a logical cluster.

2 Add instances.

3 Assign routing rules.

4 Start a logical cluster.

In this example, we will create three logical clusters for the “mycluster” physical cluster: “SalesLC,” “HRLC,” and “CatchallLC”.

## Creating a logical cluster

**Note** You can also use the Adaptive Server Plug-in to create logical clusters. For more information, see “Adding a logical cluster” on page 215.

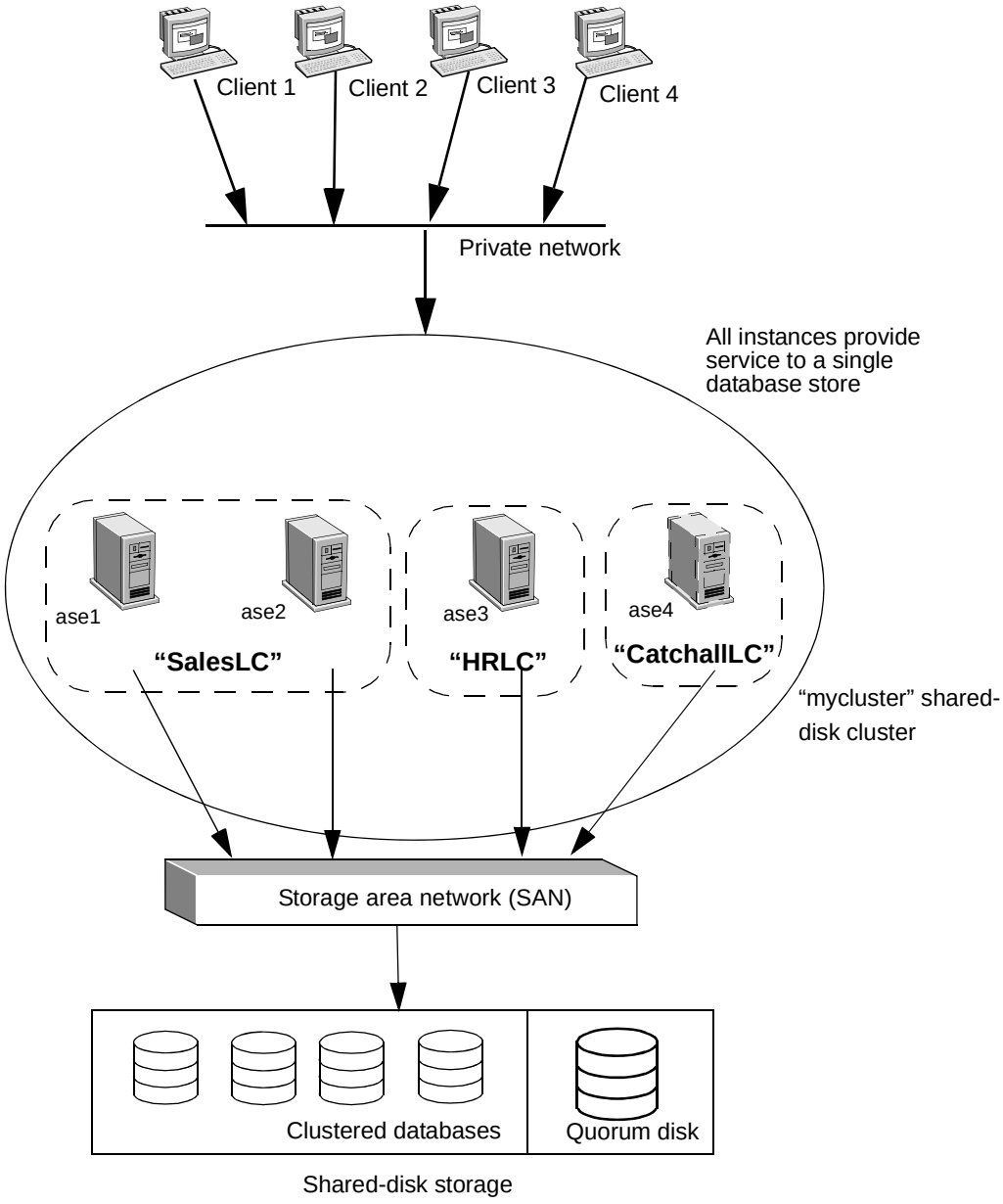
Create a logical cluster using sp_cluster logical, "create".

For example, suppose a physical cluster called “mycluster” that contains four instances: “ase1”, “ase2”, “ase3”, and “ase4”. We create three logical clusters:

- “SalesLC” – to handle applications and logins from the Sales Department.
- “HRLC” – to handle applications and logins from the Human Resources Department.
- “CatchallLC” – to use later for an open logical cluster.

To create “SalesLC”, “HRLC”, and “CatchallLC”, enter:

```
sp_cluster logical, "create", SalesLC
sp_cluster logical, "create", HRLC
sp_cluster logical, "create", CatchallLC
```


Client 1
Client 2
Client 3
Client 4
Private network
All instances provide service to a single database store
ase1
ase2
ase3
ase4
"SalesLC"
"HRLC"
"CatchallLC"
"mycluster" shared-disk cluster
Storage area network (SAN)
Clustered databases
Quorum disk
Shared-disk storage

## Adding instances to a logical cluster

Add instances to the cluster using sp_cluster logical, "add".

Add two instances to the mission-critical “SalesLC”:

```
sp_cluster logical, "add", SalesLC, instance, ase1
sp_cluster logical, "add", SalesLC, instance, ase2
```

Add a single instance to “HRLC”:

```
sp_cluster logical, "add", HRLC, instance, ase3
```

Add a single instance to “CatchallLC”:

```
sp_cluster logical, "add", CatchallLC, instance, ase4
```

## Adding routes to a logical cluster

Use sp_cluster logical, "add" to route clients to a target logical cluster. See “Assigning routing rules” on page 83 for more information.

For example, to route the applications “field_sales” and “sales_reports” to “SalesLC,” enter:

```
sp_cluster logical, "add", SalesLC, route, application,
"field_sales;sales_reports"
```

To route the login name “sales_web_user,” which is used by several sales applications via the Internet, to “SalesLC,” enter:

```
sp_cluster logical, "add", SalesLC, route, login,
sales_web_user
```

To route all clients using human resources applications to “HRLC”, enter an alias route:

```
sp_cluster logical, "add", HRLC, route, alias,
HR_SERVER
```

---

**Note** Make sure that you include each server alias in the client’s directory service, and that those query entries specify an address on which the physical cluster is listening.

---

For example, to set up an alias for the client’s directory service:

```
ase1
     query tcp ether blade1 19786
```

```
ase2
     query tcp ether blade2 19786

ase3
     query tcp ether blade3 19786

ase4
     query tcp ether blade4 19786

mycluster
     query tcp ether blade1 19786
     query tcp ether blade2 19786
     query tcp ether blade3 19786
     query tcp ether blade4 19786

HR_SERVER
     query tcp ether blade1 19786
     query tcp ether blade2 19786
     query tcp ether blade3 19786
     query tcp ether blade4 19786
```

See Chapter 13, “System Changes,” for complete syntax and usage information.

### Starting a logical cluster

To start a logical cluster, which places it in the online state, use sp_cluster logical, “online”.

For example, to start “SalesLC” and “HRLC,” enter:

```
sp_cluster logical, "online", SalesLC
sp_cluster logical, "online", HRLC
```

## Assigning routing rules

Each client connection to a shared-disk cluster is associated with a logical cluster. That association can be based on a routing rule; it can also be based on the lack of a routing rule, which means the connection is routed to the open logical cluster.

If the connection cannot be directed to a logical cluster, either because the target logical cluster is offline or the client does not support login redirection, the connection can be handled according to the target logical cluster's down-routing mode. See "Down-routing mode" on page 86 for information about specifying down-routing rules.

## Routing rules

Routing rules direct incoming client connections to the appropriate logical cluster. Once a connection is routed to a logical cluster, an Adaptive Server task entered for that route can be administered by logical cluster management capabilities.

There are three types of routing rules, or bindings. Each binding uses the name in the TDS login record.

Binding routes are listed in the order of precedence, from highest to lowest. Thus, a login route takes precedence over an application route, which takes precedence over an alias route.

- Login routes – establish a relationship between an Adaptive Server login and a logical cluster.
- Application routes – establish a relationship between an application name and a logical cluster.
- Alias routes – associate a server name with a logical cluster. Applications can choose a logical cluster from server aliases placed in the interfaces file. These aliases use unique server names.

  The alias entry can point anywhere in the physical cluster. The workload manager sends it to the correct instances via login redirection.

# Configuring logical cluster attributes

Every logical cluster possesses a set of attributes, or properties, that control different aspects of logical cluster behavior. Each attribute has a default value. You can accept the default value or change it to best suit your application environment.

To view the current settings for logical cluster attributes, use sp_cluster logical, "show". See “Viewing information about a logical cluster” on page 93 for more information.

Use sp_cluster logical to manage these attributes:

- Open – specifies a logical cluster to which clients without a specific routing plan are directed.
- System view – specifies whether monitoring and informational tools such as sp_who, sp_lock, and monitoring tables describe an instance in the cluster or the whole cluster
- Start-up mode – specifies whether a logical cluster must be started automatically or manually.
- Down-routing mode – specifies how client connections are routed if a logical cluster designated by the routing rule is not available.
- Failover mode – how and when failover instances are brought online.
- Fail-to-any – specifies whether any instance can be a failover resource or only a designated instance can be a failover resource.
- Load profile – provides a series of weighted metrics for determining the relative workload on an instance in a logical cluster.
- Login distribution mode – specifies how Adaptive Server distributes connections when a logical cluster spans multiple instances.

## Open logical cluster

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 reset the open attribute for another logical cluster. However, only one open logical cluster can exist per physical cluster.

To specify a new open logical cluster, use sp_cluster logical, "set". For example, to designate “CatchallLC” as the open logical cluster, enter:

```
sp_cluster logical, "set", CatchallLC, "open"
```

When you reset the open attribute for a new logical cluster, Adaptive Server automatically turns off the open attribute for the old open logical cluster.

You can use the open property with the down-routing mode to reserve one or more instances for the exclusive use of a specific logical cluster.

## Down-routing mode

Routing rules direct incoming client connections to the appropriate logical cluster. See “Assigning routing rules” on page 83. However, routing rules do not specify how connections should be handled when the target logical cluster is offline, or when redirection is required and the connection does not support redirection.

---

**Note** The Client-Library property CS_PROP_REDIRECT determines whether a client connection supports login redirection. By default, the value of CS_PROP_REDIRECT is true, and the client connection supports login redirection. See the *Client-Library/C Reference Manual* for more information.

---

You can specify a down-routing mode to direct connections when routing rules cannot be applied.

You can also use this attribute to reserve certain instances for specific connections. See “Resource reservation” on page 86 for more information.

Use sp_cluster logical, "set" to configure the down-routing mode. Values are:

- system – sends unroutable connections to the system logical cluster. system ensures the highest availability as the system logical cluster is always online for every instance. This is the default setting.
- open – sends unroutable connections to the open logical cluster. If the connection cannot be sent to the open logical cluster, or the connection does not support redirection, Adaptive Server applies the down-routing mode of the open logical cluster.
- disconnect – disconnects unroutable connections. This setting can enforce resource reservation by disconnecting clients that cannot be served by instances in the target logical cluster. See “Resource reservation” on page 86 for more information.

For example, to set the down-routing mode to open for “SalesLC”, enter:

```
sp_cluster logical, "set", SalesLC, down_routing,
"open"
```

This example ensures that, if “SalesLC” is unavailable, clients bound to that logical cluster are routed to the open logical cluster.

### Resource reservation

You can use the open property in conjunction with the disconnect down-routing mode to reserve one or more instances for the exclusive use of a specific logical cluster. Suppose, for example, that you want to reserve instance “ase3” for the exclusive use of the “HRLC” logical cluster:

1 Set the open property to a logical cluster that does not include “ase3.”

2 Set the down-routing mode of the open logical cluster to disconnect so that even clients that do not support redirection cannot access it.

Only connections with routing rules specifying “HRLC” can connect to “ase3.”

## System-view attribute

The system-view attribute controls how you see the cluster when using system stored procedures such as sp_who and sp_lock, or viewing monitor tables, fake tables, and so on. You can set the system_view attribute so Adaptive Server displays information about the current instance or the physical cluster.

For example, to set the system view for the instance:

```
sp_cluster logical, "set", SalesLC, system_view,
instance
```

To set the system view for the cluster:

```
sp_cluster logical, "set", SalesLC, system_view,
cluster
```

Setting the system view at a logical cluster level provides a default value for connections in a logical cluster. See Chapter 13, “System Changes,” for more information.

**Note** The system-view attribute is an administrative tool. Its current value does not affect the way applications perceive a logical cluster, its databases, or database objects.

## Start-up mode

The start-up mode attribute specifies whether a logical cluster starts automatically when the cluster starts, or whether the administrator starts it manually.

- In automatic mode, a logical cluster starts only when its first base instance comes online following a cluster restart. Thus, a logical cluster does not automatically come online when a failover instance comes online. This is the default value.

- In manual mode, a logical cluster comes online only after the administrator executes the online command.

For example, to change the startup mode for “SalesLC” to manual, enter:

```
sp_cluster logical, "set", SalesLC, startup, manual
```

## Failover mode

The failover mode specifies when and how a logical cluster runs on failover instances.

Use sp_cluster logical, "set" to specify the failover mode. Values are:

- instance – specifies that whenever an online instance fails (whether a base or failover instance), it is replaced with a failover instance on a 1:1 basis. For example, suppose logical cluster “SalesLC” runs on instances “ase1” and “ase2” with “ase3” and “ase4” as failover instances, and the failover mode is “instance.” If “ase1” fails, “ase3” comes online, and the cluster runs on “ase2” and “ase3” (or “ase4”, depending on the relative workload of the two failover instances). instance is the default value.
- group – specifies that base instances are replaced only when all base instances fail and that all failover instances then come online. Suppose the failover mode for “SalesLC” is “group.” If “ase1” fails, the cluster continues to run on “ase2”. No failover instances are brought online. However, if both “ase1” and “ase2” fail, then the cluster runs on “ase3” and “ase4”.

You can also designate multiple failover groups, so that even if instances in the first failover set fail, another set of failover instances is available to come online.

For example, to set the failover mode for “SalesLC” to “group”, enter:

```
sp_cluster logical, "set", SalesLC, failover, "group"
```

## Fail_to_any attribute

The fail_to_any attribute determines whether a logical cluster can fail over to any instance in the cluster, or only to designated failover instances. This attribute becomes important only if designated failover instances cannot be brought online.

Set the fail_to_any attribute using sp_cluster logical, "set". Values are:

- true – specifies that the system always selects other instances to act as failovers as long as any other instances in the cluster are online and available. This is the default value.
- false – specifies that only designated failover instances can be used.

For example, to turn off the fail_to_any attribute for “SalesLC”, enter:

```
sp_cluster logical, "set", SalesLC, fail_to_any, false
```

## Load profile attribute

Adaptive Server uses the load profile to provide a load score for each instance in a logical cluster. The load score determines when the workload manager directs connections to other instances to help balance the workload. You can use the sample load profiles tested and provided by Sybase or configure your own. See “Load profiles” on page 106 more information.

## Login distribution mode

The login distribution mode lets you specify how connections are distributed in logical clusters with multiple instances. The login distribution mode does not affect single-instance logical clusters.

Values are:

- affinity – specifies that the instance accepting a connection retains it as long as the target logical cluster is running on that instance.
  - If the load profile specifies a load threshold, and the load on the instance is too high, the workload manager redirects the connection to the least loaded instance in a logical cluster.
  - If the target logical cluster is not running on the instance, the workload manager redirects the connection to the least loaded instance in a logical cluster.

- round-robin – specifies that incoming connections are distributed in a round-robin fashion among the instances hosting a logical cluster. For example, if “SalesLC” is running on “ase1” and “ase2”, the workload manager sends the first connection to “ase1,” the next to “ase2”, and so on. Load scores are not included in the algorithm.

**Note** The Cluster Edition does not perform load-based redirection (affinity or round-robin) for connections with the sa_role. However, sa_role connections are redirected if a route directs them to a logical cluster running on another instance.

Sybase recommends

Sybase recommends affinity mode for transactional applications with short, frequent connections, and round-robin mode for read-mostly applications where an application server establishes a persistent pool of connections to Adaptive Server.

# Configuring failover

Any instance in a physical cluster can be a failover resource. Logical cluster failover rules do not impact infrastructure, lock remastering, or recovery. You can configure logical cluster failover by specifying:

- Failover resources – specific instances or groups of instances to which failover occurs.
- The failover mode – determines whether failover occurs for individual members of a logical cluster or only for the cluster as a whole. See “Failover mode” on page 88 for more information.
- The fail_to_any attribute – determines whether failover occurs only to designated failover resources or to any other instance if the designated failover resources are unavailable. See “Fail_to_any attribute” on page 88 for more information.

When you create a logical cluster, default settings for the fail_to_any attribute ensure that if a base instance fails, it is immediately replaced with another instance. This is the simplest failover strategy, and it is adequate for many sites.

If your site requires finer control of failover resources, you can change the default settings and direct failover to specific instances or groups of instances.

You can create up to 31 failover groups for each logical cluster. By grouping failover instances, you can give preference to specific failover groups. For example, you can ensure that instances in group 1 are considered before instances in group 2, and so on. The workload manager chooses failover instances within the group according to workload: instances with the lighter load are chosen first.

Instances can be a member of only one failover group per logical cluster. Thus, if instance “ase4” is in failover group 1 for “SalesLC”, it cannot also be in failover group 2 for “SalesLC”. However, “ase4” can simultaneously be in failover group 1 for “SalesLC”, failover group 2 for “HRLC”, and a base instance for “CatchallLC”.

When the workload manager needs to activate failover instances, it looks first in group 1, then in group 2, and so on until the failover condition is satisfied. If it cannot activate a configured failover resource, the workload manager checks the fail_to_any parameter configuration. If fail_to_any is true, the workload manager attempts to satisfy failover using any available instance. If fail_to_any is false, failover does not occur.

## Adding failover resources

Use sp_cluster logical, "add" to add failover resources to a logical cluster. You cannot add failover resources to the system logical cluster.

Each time you use sp_cluster logical, "add" to add failover resources, Adaptive Server creates one or more failover groups.

If you add one or more failover instances, separating multiple instances with semicolons, Adaptive Server places all instances in a single group.

For example, to add “ase3” as a failover group to “SalesLC”, enter:

```
sp_cluster logical, "add", SalesLC, failover, ase3
```

You can also add failover instances to existing failover groups. For example, suppose “ase3” is a member of failover group 1. To add “ase4” to failover group 1, enter:

```
sp_cluster logical, "add", SalesLC, failover, ase4, "1"
```

To view failover resource information, including the failover group ID, use sp_cluster logical, “show”.

# Managing logical clusters

This section describes how to manage logical clusters.

## User tasks and logical clusters

Each Adaptive Server task (SPID) runs inside a logical cluster. The lcid column in sysprocesses is a logical cluster ID that is hosting a given task. This ID can be passed to the lc_name() built-in function to determine the name of the corresponding logical cluster.

An individual task may run the lc_name() built-in to determine the current logical cluster.

## Managing the workload manager thread

The workload manager thread is a system-service thread that runs on each instance. When an instance starts, it automatically spawns the workload manager thread. This thread spends most of its time sleeping, but wakes up periodically to handle a logical cluster **action**, gather workload metrics, calculate the load on each instance, send load information to all instances, and other management duties.

You can view information about the workload manager by querying sysprocesses and monProcesses, using sp_who, and using other Adaptive Server capabilities for monitoring processes.

You may want to change the default value for maximum memory usage for the workload manager (see “Setting memory requirements for the workload manager” on page 92). Otherwise, the workload manager requires no maintenance.

### Setting memory requirements for the workload manager

Set the “workload manager cache size” configuration parameter to specify the maximum amount of memory that the workload manager can use. All memory used by the workload manager comes from the memory pool sized by workload manager cache size.

Connection migration consumes memory from this pool; each configured logical cluster, route, and load profile consumes memory from this pool. Actions resulting from commands such as failover and failback consume memory from this pool and continue to do so until the action is released.

Estimate the memory usage based on these guidelines:

- Four memory pages for each concurrent migrating connection
- One page for three logical clusters
- One page for two load profiles
- One page for 30 routes
- One page for 12 actions

Use sp_configure to set the maximum value of the memory pool in increments of 2K pages. For example, to set the value of workload manager cache size to 100 2K pages, enter:

```
sp_configure "workload manager cache size", 100
```

workload_manager_size is dynamic; you need not restart the server. The default value is 80, or 160KB.

The default value is sufficient for most installations. If you anticipate migrating a logical cluster with many concurrent connections, you may need to increase the size of the memory pool.

## Viewing information about a logical cluster

To view information about a logical cluster, you can:

- Use the built-in functions lc_name( ), lc_id( ), instance_name( ), and instance_id( ).
- Use the global variables *@@clustername*, *@@instancename*, and *@@instanceid*.
- Query the monitor tables.
- Use sp_cluster logical, show.

### Querying the monitor tables

The following monitor tables provide information about a logical cluster, the workload, and the workload profile:

- monLogicalCluster – provides summary information about the system's logical clusters.
- monLogicalClusterInstance – provides information about each instance in the system's logical clusters.
- monLogicalClusterRoute – provides information about configured routes.
- monLogicalClusterAction – provides information about actions in the system's logical clusters.
- monWorkload – provides the workload scores for each instance for each load profile.
- monWorkloadProfile – provides information about each load profile.

You can use Transact-SQL commands to query these tables for information. See Chapter 13, "System Changes," for a complete description of each of the monitor tables.

## Using sp_cluster logical, "show"

You can use sp_cluster logical, "show" to:

- View summary information about a specific logical cluster or all logical clusters. For example, to view information about "SalesLC", enter:

  ```
  sp_cluster logical, "show", SalesLC
  ```

  To view information about all logical clusters, enter:

  ```
  sp_cluster logical, "show"
  ```

- View information about actions. For example, to view information about all completed actions, enter:

  ```
  sp_cluster logical, "show", NULL, action, complete
  ```

  To retrieve information about cancelled actions for "SalesLC", enter:

  ```
  sp_cluster logical, "show", SalesLC, action,
  cancelled
  ```

  To retrieve information about active actions for Sales LC, enter:

  ```
  sp_cluster logical, "show", SalesLC, action, active
  ```

- View information about configured routes. You can query on a particular application, login, alias, or combination of these.

  To view information about the "sales_web_user" login route to "SalesLC", enter:

```
sp_cluster logical, "show", SalesLC, route, login,
sales_web_user
```

For complete syntax and usage information for sp_cluster logical, `"show"`, see Chapter 13, “System Changes.”

## Creating and dropping a logical cluster

- To create a logical cluster, use sp_cluster logical, "create". For example, to create “FinanceLC”, enter:

  ```
  sp_cluster logical, "create", FinanceLC
  ```

- To drop a logical cluster, use sp_cluster logical, "drop". a logical cluster must be offline or inactive before it can be dropped. sp_cluster logical, "drop" deletes the cluster and all routes, resources, and attributes associated with the cluster.

  For example, to drop “FinanceLC”, enter:

  ```
  sp_cluster logical, "drop", FinanceLC, cluster
  ```

## Adding resources to a logical cluster

Use sp_cluster logical, "add" to add resources to a logical cluster. You can add:

- Base instances – see examples in “Adding instances to a logical cluster” on page 82.
- Failover instances – see examples in “Adding failover resources” on page 91.

## Dropping resources from a logical cluster

Use sp_cluster logical, "drop" to drop one or more resources from a logical cluster. A base instance or a failover instance must be offline before it can be dropped.

You can drop:

- Base instances – for example, to drop instance “ase2” from “SalesLC”, enter:

  ```
  sp_cluster logical, "drop", SalesLC, instance, ase2
  ```

- Failover instances – for example, to drop failover instances “ase3” and “ase4” from “SalesLC”, enter:

```
sp_cluster logical, "drop", SalesLC, failover,
"ase3;ase4"
```

## Adding, moving, and dropping routes

- To add a route, use sp_cluster logical, "add". For example, to add a route for the logins “accounting” and “projects” to “SalesLC”, enter:

```
sp_cluster logical, "add", SalesLC, route, login,
"accounting;projects"
```

- To move a route from one logical cluster to another, use sp_cluster logical, “alter”. For example, to create a route to “My_LC” using the alias “SalesLC”, and then move the route from “My_LC” to “Your_LC”, enter:

```
sp_cluster logical, "add", My_LC, route, alias,
   SalesLC
sp_cluster logical, "alter", Your_LC, route, alias,
   SalesLC
```

- To drop a route, use sp_cluster logical, "drop". For example, to drop the login “projects” from “SalesLC”, enter:

```
sp_cluster logical, "drop", SalesLC, route, login,
projects
```

## Migrating connections

Use sp_cluster connection “migrate” to:

- Migrate the logical cluster or instance on which a connection (or another task) is running.
- Migrate an application or login to a logical cluster or instance for which it is not routed.

For example, this migrates the connection with a spid of 73 to the SalesLC logical cluster.

```
sp_cluster connection, "migrate", SalesLC, NULL, "73"
```

See “sp_cluster connection, migrate” on page 255.

Use sp_cluster logical to set a manual or automatic migration to another logical cluster or gather groups of connections to another logical cluster when predefined events occur. You can "gather" all qualified connections on the system or logical cluster to a designated logical cluster using the defined routing rules. The Cluster Edition looks for all connections that match the routing rules for this logical cluster, and migrates them to the specified logical cluster.

The syntax is:

```
sp_cluster logical, 'gather', lc_name
sp_cluster logical 'set', lc_name, 'gather', 'automatic | manual'
```

This gathers all the defined connections to the SalesLC logical cluster:

```
sp_cluster logical, 'gather', SalesLC
```

This sets the gathering to "manual" for the SalesLC logical cluster:

```
sp_cluster logical 'set', SalesLC, 'gather' 'manual'
```

For more information, see "sp_cluster logical, [ gather | set ]" on page 265.

# Administering failover, failback, and planned downtime

You can manually change the state of a logical cluster and its instances using sp_cluster logical, "action", when the action is one of:

- failover
- failback
- online
- offline
- deactivate

## Cluster and instance states

A logical cluster and each instance in the cluster can have different states.

- A logical cluster has an overall, or global, state that determines, for example, whether the cluster is offline or online.

- A logical cluster also has an **instance state** that describes the state of a particular instance as it is perceived by a logical cluster. For example, an online logical cluster may be online on its base instances and offline on its failover instances. This state may be independent of the actual Adaptive Server state, as a logical cluster may be offline on an instance that is actually up and running.

There are five states that are visible to the user. These states apply to a logical cluster state as well as the instance state. Table 5-1 describes each state at the global and instance levels.

***Table 5-1: Logical cluster states***

| State | At the global level | At the instance level |
|---|---|---|
| online | A logical cluster is online and running on one or more instances. | The online logical cluster is accepting and managing connections on the current instance. |
| offline | A logical cluster is not running on any instance. | A logical cluster is not running on the current instance, which cannot accept connections or consume resources. |
| inactive | Similarly to the offline state, a logical cluster is not running on any instance. Inactive logical clusters are not started automatically and do not participate in failover. The cluster achieves the inactive state only through the deactivate command. Once inactive, the cluster comes online only through the online command. | A logical cluster is not running on the current instance, which cannot accept connections or consume resources. In addition, the inactive instance cannot fail over and is not brought online after an automatic startup. This state is achieved only via the deactivate command. |
| failed | Similarly to the offline state, a logical cluster is not running on any instance. A logical cluster moves to the failed state when its active instances are shutdown with nowait or encounter a system failure when no failure resources are available. | A logical cluster is not running on the current instance, which cannot accept connections or consume resources. The failed state is reached via a shutdown with nowait or system failure. |
| time_wait | A transition state between online and offline or inactive. An online logical cluster must enter the time_wait state before becoming offline or inactive. During this time, any new connections are routed according to the down-routing mode, and existing connections migrate or disconnect. | A transition state between online and offline or inactive. When a logical cluster is online for an instance, it must enter the time_wait state before becoming offline or inactive. In the time_wait state, no new connections can be routed to a logical cluster or instance, but existing connections continue to run until they migrate or disconnect. |

## How states change

Cluster and instance states can change:

- Manually, when you execute a state change using the online, offline, failover, and failback commands; and sometimes with the action command
- Automatically, as a result of system changes

The initial state of the cluster or instance can dictate whether or not a state change is valid, and even the final state. Table 5-2 shows how different actions, executed manually, and states interact. States are described in rows; actions are described in columns. Each cell represents the new state when an action is applied to a logical cluster or instance in the initial state.

***Table 5-2: Interaction of action and state***

| | **offline** | **online** | **time_wait** | **failed** | **inactive** |
|---|---|---|---|---|---|
| Online | online | | | online | online |
| Offline | | offline/time_wait | | offline | offline |
| Failback instance | online | | | online | online |
| Failback cluster | | online/time_wait | | offline | offline |
| Failover instance | online | | | online | online |
| Failover cluster | | online/time_wait | | offline | offline |
| Cancel action | | | online | | |
| Modify wait | | | time_wait | | |
| Deactivate | inactive | inactive/time_wait | | inactive | |

States can also change as the result of system changes. Table 5-3 shows the effects of different system changes on the state of the cluster or instance.

***Table 5-3: Interaction of action and state***

| | **offline** | **online** | **time_wait** | **failed** | **inactive** |
|---|---|---|---|---|---|
| Instance joins cluster | online if automatic startup is configured | | | online if automatic startup is enabled | |
| Graceful shutdown | | time_wait | | | |
| System failure | | failed | failed | | |
| Shutdown with nowait | | failed | failed | | |

| | offline | online | time_wait | failed | inactive |
|---|---|---|---|---|---|
| Failover selection | online | | | | |

**Note** Logical cluster states are not retained following a total cluster restart. For example, suppose you execute the offline command for a logical cluster that is in automatic startup mode. The cluster is in the online state after you restart it.

## Asynchronous commands and logical cluster states

The sp_cluster logical commands deactivate, failback, failover, and offline are asynchronous. They stop an online instance that may have existing transactions. These transactions must be handled before the instance can actually be taken offline or made inactive. As a consequence, these commands can be allowed to complete at a later time and context.

When you execute any of these commands, the target instance is placed in the time_wait state, and no longer accepts new connections.

Each asynchronous command provides three "wait" options for handling existing transactions. Values are:

- wait – lets existing connections remain for a specified period of time, for example five minutes. Connections that support migration migrate as soon as they are quiescent. Connections that do not support migration disconnect when they become quiescent. HA-aware clients fail over; clients that are not HA-aware disconnect. Connections remaining after the specified time are terminated.
- until – lets existing connections remain until a specified time, for example 12:30 p.m. Otherwise, until and wait handle connections in the same way.
- nowait – terminates existing connections immediately. Connections that support migration must migrate immediately, or they are terminated.

**Note** If you do not specify a wait option when executing an sp_cluster logical asynchronous command, Adaptive Server assumes an infinite wait.

When the last connection using the instance disconnects, the instance state changes from time_wait to offline or inactive.

## Using action descriptors

Action descriptors let you track or change an action.

When an asynchronous command seeks to stop one or more instances, it generates an action descriptor. The action descriptor tracks the action, the wait option, and target instances in the time_wait state. You can view information about an action descriptor by querying the monLogicalCLusterAction table or executing `sp_cluster logical, "show", NULL, action.`

An action can be "active" or "complete." An action is active when at least one target instance remains in the time_wait state. An action is complete when all target instances are no longer in the time_wait state.

Using sp_cluster logical, action, you can manage action descriptors using these options:

- cancel – terminates active actions. Instances in the time_wait state due to that action return to the online state. Existing connections remain, even if they were marked for migration or termination.

  If the command that resulted in the creation of the action brought instances online, they remain online. For example, if an action results in the cancellation of a failover from s1 to f1, f1 remains online.

- modify_wait – changes the wait option (see "Asynchronous commands and logical cluster states" on page 100) and time associated with an active action. For example, if an action is created with a 10-minute wait, use modify_wait to change:

  - The time delay to 20 minutes
  - The time delay to the actual clock time of 4:30 p.m.
  - The wait option to nowait

- release – removes a completed action from the monLogicalClusterAction table.

Completed actions remain in the monLogicalClusterAction table so you can track their status. However, completed actions consume memory from the workload manager cache. Execute the release command after an action completes to free this memory.

**Note** Action information is stored in memory only. Restarting the full cluster clears all actions from the monLogicalClusterAction table.

## An example: scheduling and rescheduling a failover

You can execute an administrative failover for a cluster or an instance. The cluster or instance fails over to previously configured failover resources.

In this example, we fail over the "SalesLC" cluster, scheduling the failover for 2 a.m. So that we can later track or change the action, we also include the syntax that outputs an action handle:

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failover", SalesLC, cluster, NULL,
until, "02:00:00", @handle = @out_handle output
```

Suppose the command outputs the action handle "1234", and SalesLC enters the time_wait state. All new connections migrate to the failover resources. Any existing connections remaining after 2 a.m. are terminated, and "SalesLC" enters the offline state.

Suppose we find that we must migrate all connections immediately. We can use the action handle to reschedule an immediate failover. Enter:

```
sp_cluster logical, "failover", SalesLC, modify_time,
"1234", nowait
```

## Using failover, failback, online, offline, and deactivate

### failover

failover initiates a manual failover from a logical cluster's base resources to its failover resources. Failover resources must be set up previously using sp_cluster logical, "add". When initiating a partial-cluster failover, specify a list of base resources that are to fail over, and a list of failover resource to which the base instances will fail over.

For example, you can fail over a portion of a logical cluster to a set of previously configured failover resources. Here, "SalesLC" is running on instances "ase1" and "ase2." To keep "SalesLC" running on "ase2", but fail over "ase1" to the previously defined failover resource "ase3", enter:

```
sp_cluster logical, "failover", SalesLC, instance,
ase1, ase3
```

In this example, the no wait option has been specified, which, by default, specifies an infinite wait.

### failback

failback reverses a failover. It initiates a manual failback from a logical cluster's failover resources to its base resources. When initiating a partial-cluster failover, you specify a list of failover resources that are to fail back and a list of base resources to which the failover instances will fail back.

In this example, we incrementally fail back "SalesLC", which is running on "ase3", so that "SalesLC" runs on "ase1." We specify a 2-minute wait.

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failback", SalesLC, instance,
ase3, ase1, wait, "00:02:00", @handle = @out_handle
output
```

### online

online starts a logical cluster or instances in a logical cluster, placing them in the online state.

For example, to start SalesLC on "ase1", enter:

```
sp_cluster logical, "online", SalesLC, ase1
```

See “Starting a logical cluster” on page 83 for more examples.

### offline

offline stops logical clusters or instances in the online or active state.

For example, to take “SalesLC” offline, wait 5 minutes, and store the action in an action handle in a local variable, enter:

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "offline", SalesLC, cluster, wait,
00:05:00, @handle=@out_handle output
```

### deactivate

deactivate is identical to offline, except it puts the cluster or instance in the inactive state. See “offline” on page 104 for more information.

# Distributing the workload

Each instance has a workload manager thread that is responsible for calculating the load on the current instance and sending that information to the other instances in the cluster. The workload manager is a system-service thread; it is spawned when the instance starts.

Adaptive Server uses a workload measurement algorithm to calculate a load score for each instance. This load score is a unitless number that can be compared with other load scores to determine relative workloads. Thus, you can compare load scores across a cluster or at different times for a particular cluster. A load score is meaningful only when compared to other load scores.

## Workload metrics

When calculating a load score, Adaptive Server considers five system-defined metrics and, optionally, one user-supplied metric.

- User connections – the capacity of an instance to accept a new connection, based on resource availability.

- CPU utilization – the capacity of an instance to accept additional work.
- Run-queue length – the number of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time.
- I/O load – outstanding asynchronous I/Os.
- Engine deficit – the difference in the number of online engines among instances in the cluster.

**Note** Engine deficit is measurable only when instances in the cluster have unequal numbers of engines. In this scenario, engine deficit adds a metric for maximum relative capacity to the load score.

- User metric – a customer-supplied metric specific to the user's environment. This metric is optional. See "Creating a user metric."

## Creating a user metric

You can add a site-specific metric to the workload measurement algorithm using the workload_metric built-in function. In a typical example, you might monitor response time using an external monitor, and then insert the response-time values into the algorithm.

Adaptive Server normalizes system-supplied metrics before including them in the workload algorithm. For compatibility, you must normalize the user-supplied metric as well. For example, if the maximum acceptable response time is 5 seconds and the measured response time is 2 seconds, the normalized value is 40, which can be entered into the workload algorithm using workload_metric.

See Chapter 13, "System Changes," for syntax and usage information for workload_metric.

## Weighting the workload metrics

Each component of the load score is weighted according to importance. The metric values are normalized, and the results are summed to give each instance an overall load score. Sybase supplies default values that are sufficient for most sites, but if you are including a site-specific metric, you may want to adjust the weights using sp_cluster profile.

## Load thresholds

Adaptive Server uses the load score for each instance to determine:

- How best to distribute incoming connections – login redirection.
- Whether to migrate existing connections – dynamic load distribution.

Load distribution is performed only for a logical cluster running on multiple instances. The Sybase load distribution strategy is to redirect work when one instance is overloaded and other instances are available. It does not seek to maintain perfectly balanced load scores. As a result, Adaptive Server maintains load thresholds for login redirection and dynamic load distribution. The load threshold is the percentage difference between the load on the current instance and the load on the least loaded instance currently participating in a logical cluster. That value must be met before Adaptive Server redirects a login or migrates an existing connection.

Adaptive Server maintains separate load thresholds for login redirection and connection migration. Typically, the login redirection threshold is lower than that for connection migration. You can use sp_cluster profile to configure the load thresholds when creating a load profile.

Hysteresis value

The load threshold tells Adaptive Server when to redirect connections from an overloaded instance to the least loaded instance. The hysteresis value guards against migration when the threshold value is met, but the actual load scores are too low to require migration.

For example, suppose the load score on the current instance is 2 and that of the least loaded instance is 1. The percentage difference is 100%, which meets the load threshold, but the actual load scores are so low migration is not appropriate.

## Load profiles

Load profiles consolidate all configurable aspects of the workload scoring system into a single named entity.

You can assign different load profiles to different logical clusters in the same physical cluster, thus enabling multiple applications with diverse workloads within the same physical cluster. You can also share the same load profile among multiple logical clusters.

For example, a shared-disk cluster can simultaneously host a DSS-based logical cluster that is primarily read-only and an OLTP logical cluster that handles frequent writes. The optimum thresholds for login redirection and connection migration for these two clusters may be very different. The ability to assign specific load profiles to each logical cluster allows each cluster to function more efficiently.

**Note** Adaptive Server gathers load statistics for each instance without regard to logical cluster. Thus, if two logical clusters run on the same instance, they will have the same raw data for that instance. However, each logical cluster will interpret and use that data according to its own load profile.

Sybase provides a preconfigured profile created for an OLTP environment. You can also create your own load profiles using sp_cluster profile.

# Using the sample load profiles

Sybase provides two sample load profiles:

- sybase_profile_oltp – is configured for OLTP environments. It tries to keep all connections on the same instance by disabling load-based login distribution and dynamic-load distribution. Emphasis is placed on requeue depth, which is a good predictor of response time.
- sybase_profile_dds – is configured for primarily read-only, DSS environments. It uses load-based login distribution and dynamic load distribution to distribute load across multiple instances, but places emphasis on balancing CPU usage and user connections.

Table 5-4 lists the metrics for sybase_profile_oltp and sybase_profile_dss.

***Table 5-4: Metrics for the sample load profiles***

| Workload metric | "sybase_profile_oltp" | "sybase_profile_dss" |
|---|---|---|
| Profile ID | 1 | 2 |
| User connections | 0 | 40 |
| CPU utilization | 10 | 40 |
| Run-queue length | 70 | 0 |
| I/O load | 20 | 10 |
| Engine deficit | 0 | 10 |
| User weight (not used) | 0 | 0 |
| Login threshold | 0 | 20 |
| Dynamic threshold | 0 | 50 |
| Hysteresis | 20 | 20 |

# Creating and configuring your own load profile

To create and configure your own load profile:

1 Create the empty load profile.

2 Build the load profile by specifying individual metric weights and thresholds.

3 Associate the load profile with a logical cluster.

## Creating the load profile

Build the empty load profile using sp_cluster profile, "create". For example, to create the profile "my_profile", enter:

```
sp_cluster profile, "create", my_profile
```

## Building the load profile

Build the load profile by specifying:

- A weight for each of the metrics that make up the load profile
- Load distribution thresholds

## Specifying weights for load profile metrics

Use sp_cluster profile, "set" to configure weights for each of the metrics covered in the load profile:

- User connections
- CPU busy
- Run-queue length
- I/O load
- Engine deficit
- User metric (optional, see “Creating a user metric” on page 105)

See “Workload metrics” on page 104 for a description of the workload metrics.

Set each metric individually using values between 0 and 255. For example, to set weights for “my_profile”, enter:

```
sp_cluster profile, "set", my_profile, weight, "user
connections", "0"

sp_cluster profile, "set", my_profile, weight,
"cpu busy", "20"

sp_cluster profile, "set", my_profile, weight, "run
queue", "30"

sp_cluster profile, "set", my_profile, weight,
"io load", "10"

sp_cluster profile, "set", my_profile, weight, "engine
deficit", "10"

sp_cluster profile, "set", my_profile, weight, "user
metric", "30"
```

## Specifying load distribution thresholds

Set load distribution thresholds values between 0 and 100 using sp_cluster profile, "set". A value of zero (0) disables that aspect of load distribution. You can set separate load thresholds for:

- Login redirection
- Dynamic load distribution

- The hysteresis value

For example, to turn off dynamic load distribution in “my_profile,” enter:

```
sp_cluster profile, "set", my_profile, threshold,
"dynamic", "0"
```

To set the login redirection threshold to 30 for “my_profile,” enter:

```
sp_cluster profile, "set", my_profile, threshold,
"login", "30"
```

To set the hysteresis value to 20 for “my_profile,” enter:

```
sp_cluster profile, "set", my_profile, threshold,
"hysteresis", "20"
```

## Associating the load profile with a logical cluster

To associate a load profile with a logical cluster, use sp_cluster logical, “set”. For example, to associate the profile “my_profile” with “SalesLC,” enter:

```
sp_cluster logical, "set", SalesLC, load_profile,
my_profile
```

## Changing a load profile

Each logical cluster is associated with a load profile, either a system load profile or a user-created load profile. To change the load profile, associate the new load profile with a logical cluster.

For example, to change the load profile for “SalesLC” from “my_profile” to “sybase_profile_oltp,” enter:

```
sp_cluster logical, "set", SalesLC, load_profile,
sybase_profile_oltp
```

You can then drop the old load profile. For example, to drop “my_profile,” enter:

```
sp_cluster profile, "drop", my_profile
```

# Troubleshooting

Sybase provides several trace flags that you can use to troubleshoot the workload manager.

***Table 5-5: Trace flags for the workload manager***

| Trace flag | Description |
|---|---|
| 16403 | Traces SPID through connection to and disconnection from a logical cluster. |
| 16404 | Traces routes and route resolution. |
| 16406 | Traces client redirection and migration. |
| 16414 | Traces changes in a logical cluster state machines, including transitions from online to offline and so on. |

CHAPTER 6

# Cluster Cache Configuration

This chapter presents the mechanics of configuring and using named data caches in a Cluster Edition environment.



Cluster cache configuration defines multiple named caches to have local or global caches according to application needs. This feature allows cluster instances to have local caches. Objects can be bound to local or global caches. Multiple buffer pool support provides better access performance to named cache support, by facilitating large I/Os.

Users can also partition an application to localize access of application data to a particular instance serving that application.

## Global caches

Global caches are defined for every instance in a cluster. For a global cache, the attributes like cache size, buffer pool setting, are the same across all instances. Global caches have only one entry in the sysconfigures table, and all instances in a cluster read from this entry to create the cache on their respective instances.

It is possible to change such attributes as cache size, buffer pool settings, of a global cache to be instance-specific. If a particular instance has local settings, the cache is created using them. If the instance has no local definition, it uses a global definition to create the cache. In other words, instances that have local definitions override the global definitions and settings.

**Note** You can increase the size of local and global caches dynamically, but you cannot reduce them dynamically.

# Local caches

Your application can define local caches for each instance in the cluster, to cater to the instance's specific needs and to bind the cache to an object. A global definition is not required for each instance-specific cache in the cluster.

Local caches are instance-specific. You can tailor their configuration to the needs of the instance or the logical cluster to which the instance belongs. Sybase recommends that you partition applications across instances within a cluster when a particular application usually runs in a particular instance. This maximizes the benefits of local caches, as you can configure them specifically for a particular application's access patterns.

An object can be bound to only one cache, either local or global, at any particular instance. If you do not bind the object to any cache, or in case of failover, where an instance-specific cache is not configured, it uses the default data cache. To aid efficient access, Adaptive Server maintains binding information on every instance.

**Note** A local cache definition on any instance overrides any global definition at that instance.

# Creating and configuring named data caches

sp_cacheconfig creates and configures both global and local data caches. When Adaptive Server is installed, it contains a single global cache called the default data cache.

## Getting information about named caches

You can see information about caches by entering:

```
sp_cacheconfig
go

Cache Name          Status           Type  Config Value   Run Value
------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default   0.00 Mb         8.00 Mb
------------ ------------       Total      0.00 Mb         8.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
       Config Size: 0.00 Mb, Run Size: 8.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size       Config Size     Run Size      APF Percent

-------- ------------- ------------ ------------ -----------
2 Kb        1638 Kb         0.00 Mb         8.00 Mb           10
(return status = 0)
```

## Creating a new cache

The maximum data cache size is limited only by the amount of available memory on your system. The memory required to create the new cache is taken from the Adaptive Server global memory.

When the cache is created:

- It has a default wash size.
- The asynchronous prefetch size is set to the value of global asynchronous prefetch limit.
- It has only the default buffer pool.

## sp_cacheconfig

Definition

Creates a new named cache. This syntax is extended from that of nonclustered Adaptive Server. It provides an extra option to specify the instance name for the local configuration at the end of the syntax. If you do not specify the instance name, the configuration is global.

Syntax

```
sp_cacheconfig "[cachename [,cache_size [P|K|M|G]"
[,logonly | mixed ] [,strict | relaxed ] ]
        [, "cache_partition = [1|2|4|8|16|32|64]"]
[, "instance instance_name"]
```

Parameters

*cachename*

is the name of the data cache to be created or configured. Cache names must be unique, and can be up to 30 characters long. A cache name does not have to be a valid Adaptive Server identifier, that is, it can contain spaces and other special characters.

*cache_size*

is the size of the data cache to be created or, if the cache already exists, the new size of the data cache. The minimum size of a cache is 256 times the logical page size of the server. Size units can be specified with P for pages, K for kilobytes, M for megabytes, or G for gigabytes. The default is K. For megabytes and gigabytes, you can specify floating-point values. The cache size is in multiples of the logical page size.

logonly | mixed – specifies the type of cache.

strict | relaxed – specifies the cache replacement policy.

cache_partition – specifies the number of partitions to create in the cache.

**Example assumptions** The following examples assume a shared-disk cluster named MYCLUSTER, which contains two instances:

- SALES_INSTANCE
- HR_INSTANCE

**Creating a named cache** We can create a named cache log_sales of size 100M which is specific to instance SALES_INSTANCE. Executing sp_cacheconfig on instance SALES_INSTANCE displays this output:

```
sp_cacheconfig 'log_sales','100M','instance SALES_INSTANCE'
go

  Cache Name              Status Type            Config Value Run Value
------------------------ ------ -------------- ------------ ------------
```

```
default data cache        Active Global,Default       0.00 Mb    8.00 Mb
SALES_INSTANCE:log_sales  Active Mixed           100.00 Mb  100.00 Mb
                                                 ---------- ----------
                                              Total 100.00 Mb 108.00 Mb
================================================================================
Cache: default data cache, Status: Active, Type: Global,Default
       Config Size: 0.00 Mb, Run Size: 8.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size   Wash Size    Config Size    Run Size    APF Percent
--------- ------------ ------------ ------------ -----------
2 Kb      1638 Kb         0.00 Mb    .00 Mb        10
================================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size  Config Size      Run Size   APF Percent
--------- ------------ ------------ ------------    -----------
2 Kb         20480 Kb    0.00 Mb        100.00 Mb    10
(return status = 0)
```

By default, an isql connection has a cluster view. All instance-specific caches are displayed at any instance. For example, instance HR_INSTANCE displays information about cache log_sales, which is the instance-specific cache for SALES_INSTANCE. If you want HR_INSTANCE to see only the list of the local caches specific to this instance, and the global caches, set the system view to instance.

Executing sp_cacheconfig on instance HR_INSTANCE displays:

```
set system_view instance
go
Cache Name        Status Type             Config Value   Run Value
----------------- ------ -------------- ------------ ------------
default data cache Active Global,Default    0.00 Mb     8.00 Mb
                                          ------------ ------------
                                       Total 0.00 Mb 8.00 Mb
================================================================================
Cache: default data cache, Status: Active, Type: Global,Default
       Config Size: 0.00 Mb, Run Size: 8.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size    Config Size   Run Size    APF Percent
--------- ------------ ------------ ------------ -----------
2 Kb         1638 Kb       0.00 Mb      8.00 Mb       10
(return status = 0)
```

**Restricting the output of cache information to instance level** To display caches at instance SALES_INSTANCE, execute:

```
sp_cacheconfig 'instance SALES_INSTANCE'
go
Cache Name               Status Type            Config Value   Run Value
------------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default          0.00 Mb         8.00 Mb
SALES_INSTANCE:log_sales Active Mixed           100.00 Mb       100.00 Mb
                                              ------------ ------------
                                          Total 100.00 Mb 108.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
       Config Size: 0.00 Mb, Run Size: 8.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size   Wash Size     Config Size   Run Size    APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb       1638 Kb       0.00 Mb       8.00 Mb       10
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size      Wash Size  Config Size     Run Size   APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        20480 Kb       0.00 Mb    100.00 Mb        10
(return status = 0)
```

This output displays both local and global configurations of instance SALES_INSTANCE.

**Querying the existence of named caches** To find out whether a specified cache already exists. Executing sp_cacheconfig on instance SALES_INSTANCE displays this output:

```
Cache Name               Status   Type    Config Value Run Value
------------------------ ------ ----- ------------ ------------
SALES_INSTANCE:log_sales Active Mixed    100.00 Mb   100.00 Mb
                                       ------------ ------------
                                    Total 100.00 Mb 100.00 Mb
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size    APF Percent
```

```
-------- ------------ ------------ ------------ -----------
2 Kb       20480 Kb       0.00 Mb       100.00 Mb    10
(return status = 0)
```

**Using Adaptive Server syntax to create global caches** This cache is created at all instances and memory is allocated at all instances for a global cache. To create global cache tempdb_cache, run sp_cacheconfig at instance SALES_INSTANCE:

```
sp_cacheconfig 'tempdb_cache', '100M'
go
```

```
Cache Name               Status Type             Config Value   Run Value
------------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default            0.00 Mb   8.00 Mb
tempdb_cache Active Global,Mixed                  100.00 Mb    100.00 Mb
SALES_INSTANCE:log_sales Active Mixed             100.00 Mb   100.00
Mb                                                ------------ ------------Total
200.00 Mb 208.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
       Config Size: 0.00 Mb, Run Size: 8.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size     APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      1638 Kb         0.00 Mb      8.00 Mb         10
===============================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size   Wash Size     Config Size   Run Size      APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb        20480 Kb     0.00 Mb      100.00 Mb     10
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb        20480 Kb      0.00 Mb       100.00 Mb    10
(return status = 0)
```

**Creating a named cache with a single global and multiple local configurations** All cache operations can be executed from any instance. For example, to create a bigger named cache, tempdb_cache, at SALES_INSTANCE, we can connect to instance HR_INSTANCE and execute:

```
sp_cacheconfig 'tempdb_cache','150M', 'instance
SALES_INSTANCE'
```

Executing sp_cacheconfig at instance SALES_INSTANCE displays:

```
Cache Name          Status Type           Config Value   Run Value
------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default 0.00 Mb        8.00 Mb
tempdb_cache Active Global,Mixed       100.00 Mb     100.00 Mb
SALES_INSTANCE:log_hr Active Mixed       150.00 Mb     150.00 Mb
SALES_INSTANCE:tempdb_cache Active Mixed 150.00 Mb     150.00 Mb
                                  ------------ ------------
                                Total 350.00 Mb 408.00 Mb
===========================================================================
Cache: default data cache, Status: Active, Type: Global,Default
       Config Size: 0.00 Mb, Run Size: 8.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size      Config Size   Run Size APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb         1638 Kb        0.00 Mb         8.00 Mb     10
===========================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 150.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size   Wash Size    Config Size     Run Size   APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb       30720         Kb 0.00Mb    150.00 Mb        10
===========================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size  APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb      20480 Kb       0.00 Mb      100.00 Mb     10
===========================================================================
Cache: SALES_INSTANCE:tempdb_cache, Status: Active, Type: Mixed
       Config Size: 150.00 Mb, Run Size: 150.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition:          1, Run Partition:   1
```

```
IO Size  Wash Size      Config Size   Run Size     APF Percent
-------------------------------------------------------------
  2 Kb  30720 Kb        0.00 Mb        150.00 Mb        10

(return status = 0)
```

**Note** The local configuration of the named cache tempdb_cache overrides the global configuration.

For example, if you set the system view to cluster, Adaptive Server may display all the configurations for a named cache, and you should ignore run values of any configuration which is not valid at that instance. For example, SALES_INSTANCE has a valid local configuration, cache tempdb_cache. Therefore, you should ignore the run values for global configuration.

Similarly, HR_INSTANCE has a valid global configuration. Therefore you should ignore the run values for local configuration of temdb_cache that are related to SALES_INSTANCE at instance HR_INSTANCE.

**Adding memory to an existing cache** To add memory, use the syntax documented in “System Changes” on page 247. The additional memory you allocate is added to the Adaptive Server page size pool. For example, the smallest size for a pool is 2K in a server with a logical page size of 2K. If the cache is partitioned, the additional memory is divided equally among the partitions.

**Note** If adding memory to an existing global cache fails at an instance, but succeeds at least one other instance, the server treats the operation as successful at the cluster-wide level. It is thus possible to have different run values for a global cache, but a single configuration value for the cache. sp_cacheconfig shows the run values of a global cache from the syscurconfigs entry of the current instance.

To increase tempdb_cache size to 200MB, in instance HR_INSTANCE, execute the following. Executing sp_cacheconfig 'tempdb_cache' on instance HR_INSTANCE displays this output:

```
    sp_cacheconfig 'tempdb_cache','200M'

Cache Name   Status Type          Config Value   Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed   200.00 Mb    200.00 Mb
                                 ------------ ------------
                              Total 200.00 Mb 200.00 Mb
```

```
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 200.00 Mb, Run Size: 200.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size   Config Size    Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb         40960 Kb     0.00 Mb      200.00 Mb     10
(return status = 0)
```

You can also increase the cache size of a local cache using the instance option documented on "sp_cacheconfig" on page 116.

**Allocating space for a new cache** If Adaptive Server cannot allocate the amount of memory you request, it allocates all available memory and issues an error message, telling you how many kilobytes have been allocated dynamically.

However, this memory is not allocated until you restart Adaptive Server. Adaptive Server notifies you of insufficient space, either because memory is unavailable, or because of resource constraints, which system administrators should ensure are temporary. If this behavior persists, a subsequent restart may fail.

For example, if the maximum memory is 700MB, tempdb_cache is 100MB, making the server's total logical memory 600MB, and you attempt to add 100MB to tempdb_cache, the additional memory fits into maximum memory. However, if the server can allocate only 90MB, it allocates this amount dynamically, but the size field of the cache in the configuration file is updated to 100MB. On a subsequent restart, since Adaptive Server obtains memory for all data caches at once, the size of pub_cache is 100MB.

**Decreasing a cache** When you reduce a cache size, restart Adaptive Server. For example, to decrease the size of tempdb_cache to 100M, use the following. Executing sp_cacheconfig 'tempdb_cache' on instance HR_INSTANCE displays:

```
sp_cacheconfig 'tempdb_cache', '100M'
go
Cache Name   Status Type           Config Value   Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed      100.00 Mb   200.00 Mb
------------ ------------      Total 100.00 Mb 200.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 200.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
```

```
IO Size    Wash Size   Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      40960 Kb      0.00 Mb       200.00 Mb       10
(return status = 0)
```

After restarting Adaptive Server and executing the command on HR_INSTANCE:

```
sp_cacheconfig 'tempdb_cache'
go
Cache Name    Status Type         Config Value  Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed 100.00 Mb     100.00 Mb
                                 ------------ ------------
                             Total 100.00 Mb 100.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size     APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb        20480 Kb      0.00 Mb     100.00 Mb       10
(return status = 0)
```

**Deleting a named cache** To delete a named cache completely, reset its size to 0:

```
sp_cacheconfig 'tempdb_cache','0'
```

You cannot delete a named cache if objects are bound to it, and Adaptive Server issues an error message.

If the named cache has multiple configurations, the entry corresponding to the cache in the configuration file is deleted, as are the entries corresponding to the cache in sysconfigures. The cache is deleted the next time the instance is restarted. If the cache has a single configuration, either global or local, cache entry is not deleted from either the configuration file or from sysconfigures. This entry is deleted by either restarting the cluster or by creating a new configuration for the named cache.

When you delete instance-specific configuration, a named cache reverts to its global configuration, if such a configuration exists. Executing sp_cacheconfig on instance SALES_INSTANCE displays this output:

```
sp_cacheconfig 'tempdb_cache', '0', 'instance SALES_INSTANCE'
go
Cache Name     Status   Type       Config Value   Run Value
```

```
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed     100.00 Mb    100.00 Mb
                                 ------------ ------------
                                Total 100.00 Mb 100.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size   Wash Size     Config Size      Run Size   APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb      20480 Kb       0.00 Mb     100.00 Mb     10
(return status = 0)
```

**Changing the cache type** To reserve a cache for use only by the transaction log, change the cache type to logonly. This change is dynamic. To create a logonly cache at HR_INSTANCE, enter the following. Executing sp_cacheconfig 'log_hr' on instance HR_INSTANCE displays this output:

```
sp_cacheconfig 'log_hr','logonly','instance HR_INSTANCE'

Cache Name           Status Type   Config Value    Run Value
------------------ ------ -------- ------------ ------------
HR_INSTANCE:log_hr Active Log Only 150.00 Mb       150.00 Mb
                                  ------------ ------------
                                Total 150.00 Mb 150.00 Mb
================================================================================
Cache: HR_INSTANCE:log_hr, Status: Active, Type: Log Only
       Config Size: 150.00 Mb, Run Size: 150.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size    Config Size Run Size     APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        30720 Kb     0.00 Mb      150.00 Mb         10
(return status = 0)
```

**Configuring cache replacement policy** If a cache is dedicated to a table or an index, and the cache has little or no buffer replacement when the system reaches a stable state, you can set the relaxed LRU (least recently used) replacement policy. The relaxed LRU replacement policy can improve performance for caches where there is little or no buffer replacement occurring, and for most log caches. To set relaxed replacement policy:

```
sp_cacheconfig 'log_sales','relaxed','instance SALES_INSTANCE'
go
```

```
Cache Name                Status Type   Config Value Run Value
------------------------- ------ -----  ------------ ------------
SALES_INSTANCE:log_sales Active Mixed 100.00 Mb     100.00 Mb
                                       ------------ ------------
                                  Total 100.00 Mb 100.00 Mb
================================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: relaxed LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size Config Size     Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb         20480 Kb 0.00 Mb         100.00 Mb         10
(return status = 0)
```

**Note** Setting the cache replacement policy is not dynamic and requires you to restart Adaptive Server.

# Configuring and using multiple buffer pools

Use sp_poolconfig to create multiple buffer pools.

## sp_poolconfig

Description

Creates multiple buffer pools.

Syntax

sp_poolconfig *cache_name* [, "*mem_size* [P\K\M\G]", "*config_pool*K" [, "affected_poolK"]] [, 'instance instance_name']

Parameters

- *cache_name* – the name of an existing data cache.
- *mem_size* – the size of the memory pool to be created, or the new total size for an existing pool with the specified I/O size. The minimum size of a pool is 256 logical server pages. Specify size units with P for pages, K for kilobytes, M for megabytes, or G for gigabytes. The default unit is kilobytes.
- *config_pool* – the I/O size performed in the memory pool where the memory is allocated or removed. Valid I/O sizes are multiples of the logical page size, up to eight times the amount of the logical page size.

- *affected_pool* – the size of I/O performed in the memory pool where the memory is deallocated, or the pool's attributes, such as wash size and prefetch limit, are modified. If *affected_pool* is not specified, the memory is taken from the lowest logical page size memory pool.

Examples

**Creating a 4K pool for named cache** Executing sp_poolconfig 'tempdb_cache' on instance SALES_INSTANCE displays:

```
sp_poolconfig 'tempdb_cache','25M','4K'
go

Cache Name    Status Type        Config Value  Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed   100.00 Mb    100.00 Mb
(1 row affected)
                                 ------------ ------------
                              Total 100.00 Mb 100.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      15360 Kb       0.00 Mb      75.00 Mb      10
4 Kb       5120 Kb      25.00 Mb     25.00 Mb      10
(return status = 0)
```

**Creating a pool configuration for local caches** You can create an 8K pool for named cache 'log_hr'. Executing sp_poolconfig 'tempdb_cache' on instance HR_INSTANCE displays:

```
sp_poolconfig 'log_hr','50M','8K','instance HR_INSTANCE'
go

Cache Name          Status Type    Config Value   Run Value
------------------ ------ -------- ------------ ------------
HR_INSTANCE:log_hr Active Log Only   150.00 Mb    150.00 Mb
(1 row affected)
                                   ------------ ------------
                                Total 150.00 Mb 150.00 Mb
================================================================================
Cache: HR_INSTANCE:log_hr, Status: Active, Type: Log Only
       Config Size: 150.00 Mb, Run Size: 150.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size    APF Percent
```

```
-------- ------------ ------------ ------------ -----------
2 Kb         20480 Kb       0.00 Mb       100.00 Mb      10
8 Kb         10240 Kb     50.00 Mb          50.00 Mb     10
(return status = 0)
```

**Note** There is no instance-specific configuration of buffer pools for a global cache. The instance option is used to detect the local cache configuration for which pool configuration is necessary.

## Moving memory between buffer pools

To create a new 8K pool and take memory from a 4K pool rather than from a default pool:

```
sp_poolconfig 'tempdb_cache','8M','8K','4K'
go
sp_poolconfig 'tempdb_cache'
go

Cache Name                    Status Type   Config Value   Run Value
--------------------------- ------ ----- ------------ ------------
tempdb_cache         Active Global,Mixed   100.00 Mb      100.00 Mb
(1 row affected)
                                           ------------ ------------
                                    Total 100.00 Mb 100.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
8 Kb         1632 Kb      8.00 Mb     8.00 Mb         10
2Kb         15360 Kb     0.00 Mb      75.00 Mb        10
4 Kb        3480 Kb     17.00 Mb      17.00 Mb        10
(return status = 0)
```

## Changing the wash size of a pool

The wash size is the point in the cache at which Adaptive Server writes dirty pages to disk for a memory pool.

```
sp_poolconfig cache_name, 'affected_poolK ', 'wash=size[P|K|M|G]'
        [, instance 'instancename']
```

To change the wash size of an 8K pool of named cache “log_hr” to 12480K, execute sp_poolconfig 'log_hr' on instance HR_INSTANCE to display:

```
sp_poolconfig 'log_hr','8K','wash=12480K','instance HR_INSTANCE'
go


Cache Name          Status Type     Config Value   Run Value
------------------ ------ -------- ------------ ------------
HR_INSTANCE:log_hr Active Log Only   150.00 Mb   150.00 Mb
(1 row affected)
                                  ------------ ------------
                               Total 150.00 Mb 150.00 Mb
===============================================================================
Cache: HR_INSTANCE:log_hr, Status: Active, Type: Log Only
       Config Size: 150.00 Mb, Run Size: 150.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size     Wash Size   Config Size  Run Size     APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        20480 Kb       0.00 Mb   100.00 Mb        10
8 Kb        12480 Kb      50.00 Mb   50.00 Mb        10
(return status = 0)
```

## Changing a pool’s local asynchronous prefetch percentage

Local asynchronous prefetch is the percentage of buffers in the pool that can be used to hold buffers read into cache by asynchronous prefetch, but that have not yet been used. To change a pool’s asynchronous prefetch percentage:

```
sp_poolconfig cache_name, "affected_poolK ",
        "local async prefetch limit=percent "
```

To change the local synchronous prefetch of named cache log_sales, execute sp_poolconfig 'log sales' on instance SALES_INSTANCE to display:

```
sp_poolconfig 'log_sales','2K','local async prefetch limit=20','instance
          SALES_INSTANCE'
go
Cache Name                Status Type   Config Value Run Value
------------------------ ------ ----- ------------ ------------
SALES_INSTANCE:log_sales Active Mixed   100.00 Mb     100.00 Mb
(1 row affected)
                                      ------------ ------------
```

```
                                  Total 100.00 Mb 100.00 Mb
--------------------------------------------------------------------------------
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: relaxed LRU, Run Replacement: relaxed LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb       20480 Kb       0.00 Mb     100.00 Mb        20
(return status = 0)
```

## Dropping a buffer pool

You can drop a buffer pool by setting its size to 0. The memory from this pool is added to the default pool. To drop pool 4k from named cache tempdb_cache, at instance SALES_INSTANCE:

```
sp_poolconfig 'tempdb_cache','0','4K'
go
sp_poolconfig 'tempdb_cache'
go
Cache Name                    Status Type Config Value Run Value
--------------------------- ------ ----- ------------ ------------
tempdb_cache           Active Global,Mixed  100.00 Mb    100.00 Mb
(1 row affected)
                                          ------------ ------------
                                    Total 100.00 Mb 100.00 Mb
--------------------------------------------------------------------------------
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size   Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
8Kb        1632 Kb       8.00 Mb     8.00 Mb     10
2Kb       18840 Kb       0.00 Mb    92.00 Mb     10
(return status = 0)
```

# Binding objects to named caches

sp_bindcache assigns a database, table, index, text object, or image object to a cache. Before you can bind an entity to a cache:

- The named cache must exist, and its status must be "Active."
- The database or database object must exist.
- You can bind tables, indexes, or objects only from the database where they are stored.
- To bind system tables, including the transaction log table syslogs, the database must be in single-user mode.
- You must bind a database from the master database.
- You must bind a database, user table, index, text object, or image object to a cache of type "Mixed." Only the syslogs table can be bound to a cache of type "Log Only."
- To bind an object to a cache, you must own the object or the database, or have system administrator status.
- Binding objects to caches is dynamic.

**Note** Cache binding or unbinding for local system temporary databases is not dynamic, and the owner-instance must be restarted for the bindings to take effect. Cache binding or unbinding for other temporary databases, including the global system temporary database, are dynamic.

## Syntax for binding objects

Syntax

```
sp_bindcache cache_name, dbname
        [, [owner. ] tablename [, indexname |"text only'] ]
```

Parameters

*owner* – optional if the table is owned by "dbo".

Examples

To bind a database SALES to a named cache sales_cache, enter:

```
sp_bindcache 'sales_cache', 'SALES'
```

Usage

Cache binding is valid for the entire cluster. There is no instance-specific cache binding. If you bind an object to a local cache, the instance that has the cache configured for it uses that cache, and all other instances use the default data cache.

**Note** For complete documentation of sp_bindcache, see the *Reference Manual*.

## Getting information about bound caches

For the syntax of sp_helpcache, see *Volume 3, Procedures*, in the *Reference Manual*.

sp_helpcache provides information about a cache and the entities bound to it when you provide the cache name.

For example:

```
sp_helpcache 'sales_cache'
--------------------------
Cache Name         Config Size Run Size   Overhead

------------------ ---------- ---------- ----------
sales_cache:SALES_INSTANCE 50.00 Mb 50.00 Mb 6.39 Mb
(1 row affected)
--------------Cache Binding Information: --------
Cache Name Entity Name Type Index Name
Status
---------- ----------- ---- ----------
sales_cache SALES     database
V
(return status = 0)
```

## Dropping cache bindings

There are two commands to drop cache bindings:

- sp_unbindcache unbinds a single entity from a cache.
- sp_unbindcache_all unbinds all objects bound to a cache.

Syntax

sp_unbindcache dbname [, [owner. ] tablename
[, indexname | “text only”] ]

Example

To drop the cache binding of database 'sales':

```
sp_unbindcache 'SALES'
```

Usage

- When you drop a cache binding to an object, all the pages currently in memory are cleared from the cache.
- You cannot run sp_unbindcache_all on a named cache when there are system or remote local temporary databases bound to the cache. Instead, use sp_unbindcache to unbind each of these databases first, then run sp_unbindcache_all.

# Modifying the configuration file

The named caches section in the configuration file accommodates instance information. The cache section of a global cache is similar to nonclustered Adaptive Server output.

The following definition is from the configuration file in a nonclustered Adaptive Server environment:

```
[Named Cache:tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

## Format of a local named cache

The following shows the format of a local named cache:

```
[Named Cache:log_sales]
    [Instance: SALES_INSTANCE]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    [Instance: SALES_INSTANCE]
```

```
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

The global definition should be declared first in the configuration file, and then the local definition. The server does not start otherwise. An instance-specific pool configuration on a named cache, for example, is not permitted if there is no corresponding instance-specific cache configuration. The following example is illegal:

```
[Named Cache:tempdb_cache]
cache size = 100M
cache status = mixed cache
cache replacement policy = DEFAULT
local cache partition number = DEFAULT
[2K I/O Buffer Pool]
pool size = DEFAULT
wash size = DEFAULT
local async prefetch limit = DEFAULT
[instance: SALES_INSTANCE]
pool size = DEFAULT
wash size = 40960K
local async prefetch limit = DEFAULT
```

## Extra line in local cache entries

Local caches and buffer pool definitions have an extra line (`[Instance: SALES_INSTANCE]`), which tells you that the configuration belongs to the instance SALES_INSTANCE. If a named cache has both global and local configurations, the cache section of the configuration file shows:

```
[Named Cache:tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT
    [Instance: SALES_INSTANCE]
    cache size = 150M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAUL

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

```
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = 40960k
    local async prefetch limit = DEFAULT
```

## Deleted named cache with global configuration

When a named cache is deleted, the cache configuration entry resembles:

```
[Named Cache: tempdb_cache]
    cache size = 100M
    cache status = deleted
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

## Named cache with local configuration

If the named cache is configured locally, the cache section entry resembles:

```
[Named Cache: tempdb_cache]
[Instance: SALES_INSTANCE]
    cache size = 100M
    cache status = deleted
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

## Deleted entries with valid configuration

The configuration file should contain no deleted entries that still have at least one valid configuration, such as the following cache section entry:

```
[Named Cache: tempdb_cache]
```

```
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT
    [Instance: SALES_INSTANCE]
    cache size = 150M
    cache status = deleted
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = 40960K
    local async prefetch limit = DEFAULT
```

## Creating a local configuration in the presence of a global configuration

When you create a local configuration of a named cache in the presence of global configurations, all pool entries are replicated for local configuration. For example, you may have the following global configuration:

```
[Named Cache: tempdb_cache]
    cache size = 100M
    cache status = mixed cahe
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT

[4K I/O Buffer Pool]
pool size = 25.0000m
wash size = DEFAULT
local async prefetch limit = DEFAULT
```

If you create a local configuration with size 120M on this global configuration, the cache section of the configuration file resembles the following:

```
[Named Cache:tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT
    [Instance:SALES_INSTANCE]
    cache size = 120M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT


[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance:SALES_INSTANCE]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT

[4K I/O Buffer Pool]
    pool size = 25.0000m
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance:SALES_INSTANCE]
    pool size = 25.0000m
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

# Limitations

- If a named cache has only local configurations and objects bound to it, you cannot create a global cache configuration. You see error 19817. For example:

```
sp_cacheconfig 'SALES', '200M', 'instance SALES_INSTANCE'
go
sp_bindcache 'SALES','SALES_DB'
go
sp_cacheconfig 'SALES', '300m'
go
-------
Error 19817, "New configuration on cache 'SALES' is not permitted."
```

To avoid this situation in the example above, unbind all objects bound to cache “SALES”. Create a new configuration for cache “SALES”, and bind the objects again.

- If you manually edit the configuration file to include multiple configurations of a named cache, and start the coordinator, the coordinator accepts only the configuration relevant to the coordinator’s instance. For example:

```
[Named Cache: Sales]
[Instance:SALES_INSTANCE]
   cache size = 100 M
   cache status - mixed cache
   cache replacement policy = DEFAULT
   local cache partition number = 2
   [Instance:HR_INSTANCE]
   cache size - 100M
   cache status = mixed cache
   cache replacement policy = DEFAULT
   local cache partition number = 2

[2K I/O Buffer Pool]
[Instance:SALES_INSTANCE]
   pool size = 100 M
   wash size = 4765 K
   local async prefetch limit = DEFAULT
   [Instance:HR_INSTANCE]
   pool size - 75M
   wash size = 3577 K
   local async prefetch limit = DEFAULT
```

After you start instance SALES_INSTANCE, the server can see only information about the SALES_INSTANCE configuration. The server ignores HR_INSTANCE.

WORKAROUND: To avoid this situation, add configuration information before starting instance HR_INSTANCE.

CHAPTER 7

# Using Temporary Databases

This chapter describes local and global temporary databases, how to create and manage them, and how to bind logins or applications to a temporary database or group of temporary databases. This chapter also describes private devices, how to create and manage them, and how to use them for local user temporary databases.



Temporary databases provide storage for temporary tables and other temporary working storage needs. The Cluster Edition supports both local and global temporary databases. Local temporary databases can be accessed only by the owning instance, and are used primarily to store session-specific private temporary objects: #tables, worktables, and fake tables. Global temporary databases can be accessed by all instances in the cluster and are used to store temporary objects that can exist beyond the current session.

The Cluster Edition supports:

- Local system temporary databases
- Local user temporary databases
- Global user temporary databases
- The global system temporary database (dbid of 2)

Temporary databases are inherited from the model database with these database options set:

- select into/bulkcopy
- trunc log on chkpt

A guest user is automatically added to the temporary database, and all users are granted create table permission.

# Types of temporary databases

## Local temporary databases

There are two kinds of local temporary databases:

- Local user temporary databases
- Local system temporary databases

The Cluster Edition lets you create temporary databases for each instance in the cluster. An instance-specific temporary database is called a local temporary database; the instance owning the local temporary database is called its owner instance.

Each local temporary database is bound to a particular instance and can be accessed only from that instance. You must create a local system temporary database for each instance in the cluster. Creation of local user temporary databases is optional.

### Local user temporary databases

You can create multiple local user temporary databases for each instance and then bind applications or logins to an individual local user temporary database or group of temporary databases.

### Local system temporary databases

A local system temporary database is the required default temporary database for each instance. You configure it when the cluster is configured or when a new instance is added to the cluster. The instance stores all session-specific temporary objects (such as # tables and worktables) in this database unless you create and use local user temporary databases for the instance. You must create local system temporary databases on shared storage. See also “Using private devices for temporary data” on page 151.

In a nonclustered Adaptive Server environment, the system temporary database (dbid 2) is added to the default temporary database group. In the Cluster Edition, the local system temporary database is not part of the default group for the instance. The local system temporary database is assigned to a session only if the default group for the current instance is empty and no other bindings have been specified.

---

**Note** For the Cluster Edition, the default temporary database assigned to a session is a local temporary database for the instance, not the system tempdb (with a dbid of 2). You may need to modify applications that, in a nonclustered environment assumed the default assigned temporary database as system tempdb (with a dbid of 2) in their actions, so that these actions are now applied to the assigned local temporary database.

For example, in a nonclustered Adaptive Server, if an application truncates the log of the default temporary database by:

```
dump tran tempdb with truncate_only
```

In the Cluster Edition, you must modify the application to truncate the log of the default assigned temporary database:

```
declare @tempdbname varchar(30)
select @tempdbname = db_name(@@tempdbid)
dump tran @tempdbname with truncate_only
```

---

## Global temporary databases

The system automatically creates a global system temporary database (dbid 2) when the master device is created. You can create additional global temporary databases.

You can access global temporary databases from any instance in the cluster, which allows you to create temporary objects that can be shared across sessions on the same or different instances. You cannot create temporary objects such as # tables and worktables in them as global temporary databases cannot be assigned to a specific session.

Global temporary databases are identical to normal user databases with one exception: they are re-created every time the cluster starts. Global temporary databases provide standard logging, log flushing, I/O behavior, and runtime rollback features. They are supported primarily for backward compatibility with earlier versions of Adaptive Server.

All objects in the global temporary database are lost when the cluster shuts down. However, should an instance fail, the instance failover recovery step recovers the objects in global temporary databases as well, which ensures that objects created during a session on the failed instance continue to be available from surviving instances. The global temporary database ensures that temporary shareable workspace continues to exist as long as even one instance in the cluster is active.

**Note** Applications created since Adaptive Server 12.5.0.3 may create shareable tables in user-created temporary databases. To enable existing applications to continue to work with the Cluster Edition, user-created temporary databases that include these shareable tables need to be dropped and recreated as global temporary databases with the same names.

## Summary information

Table 7-1 summarizes the basic characteristics of these databases.

***Table 7-1: Characteristics of temporary databases***

| Feature | Local system temporary database | Local user temporary database | Global user temporary database | Global system temporary database |
|---|---|---|---|---|
| Supported temporary database objects | Both session-specific temporary objects and regular database objects. | Both session-specific temporary objects and regular database objects. | Regular database objects only, no session specific temporary objects. | Regular database objects only, no session specific temporary objects* |
| Recovery | Re-created when owning instance restarts. | Re-created when owning instance restarts. | Re-created when the cluster reboots, and transactionally recovered when an instance fails. | Re-created when the cluster reboots, and transactionally recovered when an instance fails |
| Accessibility | Accessible from owning instance only, but it must be created on a shared device so that it can be created and dropped from other instances. | Accessible from owning instance only. | Accessible from any instance | Accessible from any instance |

| Feature | Local system temporary database | Local user temporary database | Global user temporary database | Global system temporary database |
|---|---|---|---|---|
| Creation | User-created, one (required) for each instance. | User-created, zero or more for each instance. | User-created, zero or more for the cluster. | System-generated (dbid = 2), one for the cluster |
| Binding allowed? | No | Yes | No | No |
| Storage (shared or private devices) | Shared storage only | Both shared and local storage | Shared storage only | Shared storage only |

* During tempdb configuration mode, the global system temporary database acts like a local system temporary database to the boot coordinator, and thus supports both session-specific temporary objects and regular database objects.

# Creating temporary databases

In the Cluster Edition, each cluster has a global system temporary database, and each instance in the cluster has a local system temporary database. Other temporary databases are optional. The global system temporary database is created when you install Adaptive Server. This section describes how to create the other temporary databases.

Sybase recommends that you create temporary databases of a similar size to allow applications to access each instance without regard to temporary space requirements.

## Creating local system temporary databases

You must create a local system temporary database for each instance during the initial startup of the cluster and later on whenever you add an instance to the cluster. Create the local system temporary database on a shared device. You can create or drop a local system temporary database from any instance, but you can access it only from the owning instance.

Create local system temporary databases using the Adaptive Server Plug-in or sybcluster. For more information, see the installation guide for your platform.

## Creating local and global user temporary databases

You can create local and global user temporary databases at any time.

You must create a local user temporary database from the owning instance. For example, to create the local temporary database local_temp1 on ase1, enter:

```
create temporary database local_temp1
    for instance ase1 on default = 50
```

You can create a global user temporary database from any instance in the cluster. For example, to create the global temporary database global_tempdb1 on ase1, log in to ase1 and enter:

```
create global temporary database global_tempdb1
    on default = 50
```

**Note** The create database phrase for instance *instance_name* is optional. If you do not include this phrase, Adaptive Server creates the temporary database for the current instance. See "create database" on page 248 for a description of its complete syntax.

# Binding users and applications to temporary databases

You can use sp_tempdb to bind logins and applications to temporary databases or temporary database groups, and to create temporary database groups in both the nonclustered Adaptive Server and the Cluster Edition. In the Cluster Edition, which supports both local and global temporary databases, you can bind logins or applications to local user temporary databases or temporary database groups, but not to global temporary databases.

The default temporary database group is system-generated and always present. It is empty unless you use sp_tempdb to add local temporary databases to it. In the Cluster Edition, the local system temporary database is not part of the default group.

The sp_tempdb user interface includes the option unbindall_gr, which removes all bindings to the specified group in either the nonclustered Adaptive Server or the Cluster Edition. In addition, the unbind option includes an *instance_name* parameter that is specific to the Cluster Edition.

## Creating and managing temporary database groups

Creating and managing temporary database groups is the same all editions of Adaptive Server.

**Note** In the Cluster Edition, the local system temporary database is not, by default, a member of the default group.

- To create a temporary database group, use sp_tempdb create. For example, to create a temporary database group named tempdbgroup1:

```
sp_tempdb "create", "tempdbgroup1"
```

- To drop a temporary database group, use sp_tempdb drop. For example, to drop the temporary database group named tempdbgroup1:

```
sp_tempdb "drop", "tempdbgroup1"
```

- To add a database to a temporary database group, use sp_tempdb add. For example, to add database local_tempdb1 to the temporary database group named tempdbgroup1:

```
sp_tempdb "add", "local_tempdb1", "tempdbgroup1"
```

- To remove a database from a temporary database group, use sp_tempdb remove. For example, to remove database local_tempdb1 from the temporary database group named tempdbgroup1:

```
sp_tempdb "remove", "local_tempdb1", "tempdbgroup1"
```

See the *Reference Manual, Procedures* for complete syntax and usage information for sp_tempdb.

## What you can bind

You can bind:

- A user login or application to multiple local user temporary databases, but only to one database per instance.

  For example, you can bind “sa” to local_tempdb1 on “ase1” and local_tempdb2 on “ase2”, but you cannot also bind “sa” to local_tempdb3 on “ase1”.

- A user login or application to only one database group. Temporary database groups are visible from all instances.

For example, if you bind isql to the default group, you cannot also bind isql to another temporary database group.

- A user login or application to individual temporary databases or a temporary database group, but not to both.

  For example, if "sa" is bound to local_tempdb1 on "ase1", it cannot also be bound to the default group.

## How the session binding is determined

When a session is initiated, Adaptive Server checks to see if a binding is applicable. If more than one binding is applicable, the Cluster Edition determines which binding is used based on the following algorithm. Only one binding is supported per session.

1 If a binding exists for the current login, that binding is used. Otherwise, the Cluster Edition searches for a binding for the current application. If both are found, use the binding for the login: a login binding takes precedence over an application binding.

  For example, if user "sa" is bound to tempdb1 on instance "ase1", and isql is bound to tempdb2 on "ase1", then, when user "sa" uses isql to initiate a session on "ase1", only the "sa" user binding is used.

  **Note** If the first binding fails, the Cluster Edition automatically assigns a member of the default group. The Cluster Edition does not search for another binding.

2 If the binding is to a database, the database is assigned. If the binding is to a database group, a member database from the current instance, selected in a round-robin fashion, is assigned. Local temporary databases that are members of a group can be assigned only to a session established on the owner instance. They cannot be assigned to a session on another instance using the group binding.

3 If a binding is not found or an assignment cannot be made, an assignment is attempted for a member of the default group—selected in a round-robin fashion.

4 If an assignment cannot be made to a member of the default group, the assignment is made to the local system temporary database.

## Creating and managing bindings

sp_tempdb bind binds a login or application to a local temporary database or database group. For example, to bind the “sa” login to the default group, enter:

```
sp_tempdb "bind", "LG", "sa", "GR", "default"
```

To bind isql to local_tempdb1 on ase1, enter:

```
sp_tempdb "bind", "AP", "isql", "DB", "local_tempdb1"
```

sp_tempdb unbind removes one or more bindings. For the Cluster Edition, this command includes an instance_name parameter that lets you drop the binding to a particular local temporary database.

For example, to unbind isql from local_tempdb2on “ase2”, enter:

```
sp_tempdb "unbind", "AP", "isql", NULL, "ase2"
```

---

**Note** If isql is bound to a database group, the above command removes the binding for the group. If multiple database bindings exist for isql, the above command removes only the binding for “ase2”; bindings to temporary databases on other instances are unaffected.

---

To unbind a user login or application from a database group or any temporary databases, use the unbind parameter with only the login or application name. For example, to remove the binding for the "sa" login:

```
sp_tempdb "unbind", "lg", "sa"
```

To unbind all logins and applications to a particular group, use sp_tempdb unbindall_gr. For example, to remove all bindings to tempdbgroup1, enter:

```
sp_tempdb "unbindall_gr", "tempdbgroup1"
```

To unbind all login and application bindings to a particular database, use sp_tempdb unbindall_db. For example, to unbind all bindings to localtempdb1, enter:

```
sp_tempdb "unbindall_db", "localtempdb1"
```

## Displaying group and binding information

To display a list of existing groups, group members, and login and application bindings, use sp_tempdb show. For example, to view the members of the temporary database group “tempdbgroup1” and their owner instances, enter:

```
sp_tempdb "show", "gr" "tempdbgroup1"
```

To display a list of active sessions assigned a temporary database, use sp_tempdb who. For example, to displays active session assigned the temporary database “localtempdb1”, enter:

```
sp_tempdb "who", "localtempdb1"
```

## Dropping temporary databases

Some restrictions apply to dropping temporary databases:

### Dropping all but the last local system temporary database

Because a local system temporary database is always in use if its owner instance is running, you cannot drop a local system temporary database while its owner instance is running.

To drop a local system temporary database, shut down the instance, and then drop the local system temporary database from another instance.

### Dropping the last local system temporary database

To drop the local system temporary database of the last instance:

1. Use sp_tempdb_markdrop to mark the database to be dropped. For example, if ase3_tdb1 is the last local system temporary database, mark it with this command:

   ```
   sp_tempdb_markdrop ase3_tdb1
   ```

2. Shut down and restart the last instance. The local system temporary database marked for “drop” is not used.
3. Drop the temporary database from this instance.

# Restrictions for temporary databases

In the Cluster Edition:

- Attempting to access a local temporary database owned by another instance results in this error:

```
Error Number : 969

You can access database '<local-tempdbname>' only
from its owner instance '<owner-instance-name>'. You
cannot access local temporary databases from non-
owner instances except to use CREATE DATABASE and
DROP DATABASE with local system temporary databases.
```

In general, you must execute stored procedures and commands that access a local temporary database from the owner instance. Alternatively, you can use the following methods to execute the operation on a local temporary database of a remote instance:

- Use connect to *instance_name* to connect to the owner instance before you access any of the local temporary databases.
- Use sp_remotesql to send the Transact-SQL statement to the remote instance. For example, to execute a query on local temporary database local_tempdb_ase1 owned by instance ase1:

```
sp_remotesql "ase1", "select * from local_tempdb_ase1..sysobjects"
```

- For stored procedures, you can execute the procedure specifying the instance name. For example, to execute sp_spaceused on local_tempdb_ase1 for instance “ase1”:

```
ase1.local_tempdb_ase1.dbo.sp_spaceused
```

**Note** Use db_instanceid to determine the owner instance ID of a local temporary database.

- Some stored procedures and commands that perform operations on all databases in a non-clustered Adaptive Server, may skip these operations on remote local temporary databases in the Cluster Edition. To make sure these operations occur in all databases, execute the operation again for each of the skipped databases from the owner instance.

  For example, if you execute sp_dbcc_faultreport without specifying a database name, it skips remote local temporary databases from the fault report.
- You cannot grant or revoke user-defined roles to users of a local temporary database.
- You cannot grant or revoke permissions on objects in a local temporary database.

- You cannot use a local temporary database as the default database for any login; any attempt to do so causes sp_addlogin and sp_modifylogin to fail.
- You cannot include a referential integrity constraint that references a column on a local temporary database except from a table on the same local temporary database. create table, alter table, and create schema fail when they attempt to create a reference to a column on a local temporary database from a different database.
- You cannot encrypt a column with an encryption key located in a local temporary database unless the table resides on the same local temporary database. create table and alter table fail when they attempt to encrypt a column with an encryption key located on the local temporary database and the table resides in another database.
- You cannot specify database recovery order for any local system temporary database. Local system temporary databases are user-created, but they are always recovered with system databases.
- You cannot use sp_configure to change configuration options from inside the local temporary database context.

# Private device support for local databases

The Cluster Edition allows you to use inexpensive storage, such as local disks and file system devices, to provide instance-specific temporary space needs. You can use such storage, created as a private device, to create instance-specific local user temporary databases on the private device. However, do not use private devices for local system temporary databases. Local system temporary databases must be created on shared storage.

In the Cluster Edition, a private device is owned and accessed by one cluster instance only. That a device is private is a *logical* attribute; the physical path to the device may or may not be accessible from a non-owner instance.

In the sysdevices table, a private device is identified by the bitmap status2 column, and the instance that owns the device by the instanceid column of the row belonging to the device.

An instance can own many private devices, but a private device is owned by a single instance.

A private device can be used only for local user temporary databases, on the instance that owns the device. It cannot be used as default storage; if you invoke sp_diskdefault on a private device, the stored procedure fails and an error message appears.

A private device cannot be mirrored, unmirrored, or remirrored.

## Using private devices for temporary data

To achieve optimal performance of temporary data in the Cluster Edition, Sybase recommends that you use fast local disks for local user temporary databases. Local system temporary databases must be created on shared storage. To limit the use of shared storage for temporary data:

1 Create one or more private devices on local disks.

2 Create one or more local user temporary databases on the private devices.

3 Add these local user temporary databases to the default temporary database group.

A user connection that has no explicit temporary-database binding will be assigned a member database from the default group. This limits the use of local system temporary databases created on shared storage to internal system tasks while all user connections use local user temporary databases created on fast local disks.

## Creating private devices using *disk init*

Use disk init to create a private device. disk init includes an optional instance parameter that marks the device private to an instance.

This example creates a device named *private_dev1* that is private to the cluster1_1 instance:

```
disk init
name = "private_dev1",
physname = "/usr/u/sybase/data/private_dev1.dat",
vdevno = 2, size = 5102, instance = "cluster1_1"
```

You can execute a disk init command from any of the instances in the cluster. The command is shipped to the owning instance internally. The owning instance, however, must be up and running to process the command. If it is not, the command fails and the private device is not created.

## Reinitializing private devices using *disk reinit*

As part of the procedure for restoring the master database, use disk reinit to restore the private device entries in the master database sysdevices table. disk reinit includes the optional instance parameter, which marks a device as private to an instance.

This example restores the private_dev1 device as a private device for the instance cluster1_1 in the master database sysdevices table:

```
disk reinit
name = "private_dev1"
physname = "/usr/u/sybase/data/private_dev1.dat"
vdevno = 2, size = 5120, instance = "cluster1_1"
```

You can execute this command from any instance in the cluster. If the owning instance is up and running, the command is shipped to it.

If the owning instance is not up and running, disk reinit inserts a row in sysdevices, corresponding to the device, but does not activate it until the instance is running. This is useful when devices must be restored before issuing disk refit.

---

**Note** Take extra care regarding the parameters you pass when you restore a private device for an instance that is not running. Adaptive Server cannot verify the physical path of a private device in such cases; it is assumed that the user provides a valid physical path.

---

## Dropping private devices using *sp_dropdevice*

Use sp_dropdevice to drop a private device. You can execute this procedure from any instance in the cluster. However, it fails to drop a private device if the owning instance is not running. For details about using sp_dropdevice, see the *Reference Manual: Procedures*.

## Displaying private device information using *sp_helpdevice*

sp_helpdevice reports whether or not a device is private. If the device is private, sp_helpdevice prints the name of the instance in the cluster that owns the private device.

In this example, `'regular_dev'` is a regular shareable device, while `'private_dev1'` is a private device owned by instance `'cluster1_1'`.

```
1> sp_helpdevice
2> go
 device_name
       physical_name
       description
       status cntrltype vdevno      vpn_low     vpn_high

 ------------------------
       ------------------------------------------------------------------------
------------------

         ------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
----------------------------------------

          ------ --------- ----------- ----------- -----------
 master
       /remote/percy_dev/user1/sdc-DBs/master.dat
       file system device, special, dsync on, directio off, default disk,
       physical disk, 512.00 MB, Free: 351.00 MB
          3         0           0           0      262143
 private_dev1
       /percy_dev/user1/sdc-DBs/private_dev1.dat
       file system device, special, private 'cluster1_1', MIRROR DISABLED, non
       serial writes, dsync off, directio on, physical disk, 2.00 MB
          2050         0           1           0        1023
 regular_dev1
       /percy_dev/user1/sdc-DBs/dev1.dat
       file system device, special, dsync off, directio on, physical disk, 2.0
       0 MB
        2         0           2           0        1023
 tapedump1
       /dev/nst0
       unknown device type, disk, dump device
          16         2           0           0       20000
 tapedump2
       /dev/nst1
       unknown device type, tape,         625 MB, dump device
          16         3           0           0       20000

(5 rows affected)
(return status = 0)
```

If disk refit is pending on a device, sp_helpdevice reflects it in its output.

In this example, `'regular_dev'` is a regular shareable device, while `'private_dev1'` and `'private_dev2'` are private devices owned by instance `'cluster1_1'`. Further disk refit is pending on `'private_dev1'`.

```
1> sp_helpdevice
2> go
 device_name
       physical_name
       description
       status cntrltype vdevno      vpn_low     vpn_high

 ------------------------

         ------------------------------------------------------------------------
-------------------

         ------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------

         ------ --------- ----------- ----------- -----------
 master
       /remote/percy_dev/user1/sdc-DBs/master.dat
       file system device, special, dsync on, directio off,
       default disk, physical disk, 512.00 MB, Free: 351.00 MB
          3         0           0           0      262143
 private_dev1
       /percy_dev/user1/sdc-DBs/private_dev1.dat
       file system device, special, private 'cluster1_1',
       REFIT PENDING, MIRROR DISABLED, nonserial writes, dsync off, directio
       on, physical disk, 2.00
MB
       2050         0           1           0        1023
 private_dev2
       /percy_dev/user1/sdc-DBs/private_dev2.dat

        file system device, special, private 'cluster1_1', MIRROR DISABLED, non
        serial writes, dsync off, directio on, physical disk, 2.00
 MB
       2050         0           3           0        1023
 regular_dev1
       /percy_dev/user1/sdc-DBs/dev1.dat
       file system device, special, dsync off, directio on, physical disk, 2.00
 MB
       2         0           2           0        1023
 tapedump1
       /dev/nst0
       unknown device type, disk, dump device
```

```
            16          2           0           0       20000
 tapedump2
       /dev/nst1
       unknown device type, tape,         625 MB, dump device
       16          3           0           0       20000

(6 rows affected)
(return status = 0)
```

## Using *create database* and *alter database* with a private device

Since a private device belongs to only one instance in the cluster, it can be used only for local user temporary databases on the instance that owns the private device. When a private device is used to create or extend a database or its log, unless it is for a local user temporary database of the instance that owns the device, create database or alter database fails and an error message appears.

## Using *disk refit*

If there are no private devices in the cluster, disk refit operates as usual.

If there are one or more private devices in the cluster, disk refit becomes a two-phased operation:

- Phase One performs disk refit only for those devices accessible to the instance that initiated the command— regular shareable devices and private devices owned by that instance.
- Phase Two performs disk refit only for the private devices of the instances in the cluster, one instance at a time. disk refit is complete only when both these phases are completed successfully.

❖ **Executing disk refit when there are private devices in the cluster**

You must restore the sysdevices table correctly before you execute disk refit. Use the disk reinit command to add regular or private devices for any missing device entry in the sysdevices table.

For more information on restoring sysdevices, see the *System Administration Guide, Volume 2*.

1. Restart the cluster in single-user mode. This is done by starting an instance in the cluster with the -m option. Single-user mode requires only one instance running in the cluster, and no other active connections to the cluster. You must set trace flag 3608.
2. Issue the disk refit command on the instance. This removes all rows in sysdatabases and sysusages, and rebuilds parts of them from the devices in sysdevices, which are either regular sharable devices accessible by all instances in the cluster, or are private to this instance. At the end of Phase One, the cluster is shut down.

   Adaptive Server prints messages at the end of Phase One that indicate whether disk refit is complete. If it is not, Adaptive Server also prints messages to inform the user which other instances in the cluster have private devices and must perform Phase Two.

   Alternatively, you can use sp_refit_admin 'status' to find instances with private devices on which disk refit has not yet run.

   A sample output:

```
1> sp_refit_admin 'status'
2> go
You need to run DISK REFIT on the following instances to refit their
private devices.

 instance
   device

status
------------------------------------------------------------
   ------------------------------------------------------------

         ------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

----------------------

cluster1_1

private_dev1
```

```
REFIT

PENDING

(1 row affected)

(return status = 0)
```

3 For each instance that have private devices and which still have to run disk refit:

- Start the instance using the -m option and trace flag 3608.
- Issue disk refit on that instance.

disk refit rebuilds sysdatabases and sysusages rows for private devices owned by this instance. When Phase Two for the instance is successfully completed, the cluster shuts down.

4 Even if you see no errors while executing disk refit, you must check the consistency of sysdatabases and sysusages before resuming normal operation. “Troubleshooting” on page 175 contains sample SQL statements that help to check and fix common problems encountered by disk refit.

5 Resume normal operation.

CHAPTER 8

# Running Job Scheduler in a Clustered Environment

This chapter describes how to run Job Scheduler in a shared-disk cluster environment. For information about installing, starting, and configuring Job Scheduler, see the *Job Scheduler Users Guide*.



Adaptive Server version 15.0.1 CE allows you to run the Job Scheduler in a shared-disk cluster environment. A single Job Scheduler services all job scheduling requests originating from all instances in the cluster. Job Scheduler runs on the boot coordinator.

If you manually start the Job Scheduler after the cluster is started, the cluster selects the coordinator to host the Job Scheduler.

## Installing and configuring Job Scheduler

The coordinator instance running the Job Scheduler is also known as the "Job Scheduler instance." To set up and start the Job Scheduler, follow directions in Chapter 2, "Configuring and Running Job Scheduler," in the *Job Scheduler Users Guide.* See the installation guide for your platform for instructions for installing and configuring Job Scheduler in the Cluster Edition.

# Running Job Scheduler in a clustered environment

If a user performs a Job Scheduler action on an instance other than the Job Scheduler instance, the instance issues a request to the Job Scheduler instance. The Job Scheduler processes the request and sends the reply back to the requesting instance.

# Shutting down Job Scheduler

Shutting down the Job Scheduler in a shared-disk cluster is the same as shutting it down in a nonclustered Adaptive Server configuration.

From the Adaptive Server Sybase Central plug-in:

1 From any Adaptive Server instance, right-click the Scheduled Jobs folder (the Scheduled Jobs folder appears under each instance, and each instance can initiate all Job Scheduler commands from the plug-in).

2 Select Administer to open the Job Scheduler Administration dialog.

3 From the Task Configuration tab, select Stop.

From the command line, run:

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

# Redirecting scheduled jobs

When the Job Scheduler agent attempts to connect to an instance to execute a scheduled job and the instance is too busy, the Workload Manager may redirect the connection to another instance, so any instance in the cluster can perform the scheduled jobs. This happens automatically and does not require you to reconfigure Job Scheduler.

Generally speaking, the Workload Manager takes care of login redirection. You can affect how connections are redirected by setting rules at the logical cluster level. However, if you do not want a scheduled job to be redirected, set the new job property that allows redirection:

- From Adaptive Server plug-in – unselect the Allow Redirection option in the Job Properties dialog box.

  The default behavior is to allow the Workload Manager to redirect a scheduled job connection according to the workload rules that exist at the time the connection is made.

- From the command line – set the redirection property no_conn_redirection when you create or modify a job. For example, to set the property for a job named find_old_logins using sp_sjobcreate, enter:

```
sp_sjobcreate @name='jname=find_old_logins',
@option='jcmd=exec
sp_find_old_logins,jproperties=no_conn_redirection=
true'
```

CHAPTER 9

# Additional Topics

This chapter describes some elements of cluster architecture in greater detail. These discussions provide supportive information; in general, a database administrator cannot influence the features described here using Transact-SQL commands or configuration parameters.



Many subsystems of nonclustered Adaptive Server work in a shared-disk cluster environment; others have been developed specifically for the Cluster Edition. These subsystems are:

- Lock manager
- Buffer manager
- Cluster interprocess communication (CIPC)
- Recovery (start time and failover handling)

## Locks

All database data and metadata objects can be accessed and cached by any instance in the cluster. As a consequence, these objects are subject to distributed locking and cache coherency enforcement, as are runtime global data structures and variables.

The cluster lock manager (CLM) provides distributed locking services for sharing globally accessible and cacheable objects. The CLM creates and queues locks on behalf of locking requests. It arbitrates locking requests for access rights from any instance to any global object in the shared-disk cluster environment.

Locks can have task, transaction, or instance ownership. Locks managed by the local lock manager have task or transaction ownership. Locks managed by the CLM have instance ownership; they are shared among all processes or transactions on that instance, and are retained until another instance requests a conflicting lock mode.

## Deadlocks

For all-pages locking in a nonclustered edition of Adaptive Server, if the server fails to acquire a page lock, it retries the attempt instead of waiting for the lock (which can cause a deadlock). The deadlock retries configuration parameter determines the number of retries Adaptive Server attempts.

The Cluster Edition includes a physical lock, which ensures page consistency across nodes. However, a physical lock is not included in the deadlock detection mechanism, so a server cannot detect a deadlock.

The Cluster Edition retries failed attempts to acquire a lock. However, when the server exceeds the value for deadlock retries, it fails or rolls back the query instead of waiting on the lock and risking an undetected deadlock.

## Retention locks

The Cluster Edition also uses retention locks, which are cluster-wide locks granted to instances in the cluster with ownership shared among all processes on the instances. Ownership of a retention lock is retained until another instance claims a conflicting lock mode, or the resource associated with the lock is claimed by another instance. The use of retention locks reduces the need to acquire and release locks, and thus reduces intra-instance messaging.

The Cluster Edition uses several different retention locks.

All locks are retention locks.

# Memory

The Cluster Edition requires more memory than a standalone SMP Adaptive Server. The additional memory supports internode messaging and cluster locking.

- You can configure the memory for internode messaging using the CIPC regular message pool size configuration parameter. The default value is 8MB.
- To support distributed cache coherency, the Cluster Edition automatically configures locks for buffer caches and system descriptors when an instance starts.

  Each buffer in the data cache is automatically configured with a physical lock. The overhead for each physical lock is approximately 24.6% for databases with 2KB pages and 12.3% for databases with 4KB pages. Thus, for a 100M data cache, the additional memory overhead for physical locking is 24MB for 2KB pages and 12MB for 4KB pages.

  The overhead for system descriptors averages about 1.5KB per open object. For best performance, the lock hashtable size parameter is automatically tuned to 8 locks per bucket. For a large cache configuration, lock hashtable size may be automatically tuned to several megabytes.

# Thresholds

A user-created stored procedure executes when free space on a segment drops below the threshold maintained by Adaptive Server.

All databases have a last-chance threshold on the log segment, which is an estimate of the number of free log pages that are required to back up and truncate the transaction log. The last-chance threshold (LCT) is a defined amount of available space; a specified number of free pages on a log segment.

The LCT also defines the action to be taken when the amount of space falls below the specified threshold value. This threshold monitors free space within a database, preventing the transaction log from running out of space.

When you add a threshold to a segment, you must specify the stored procedure used to monitor this threshold. By default, Adaptive Server uses sp_thresholdaction for the last-chance threshold.

For users, threshold maintenance in cluster processing is indistinguishable from that of a nonclustered Adaptive Server. However, there are some changes in dbcc commands, described below.

## *dbcc thresholds* output

dbcc thresholds prints segment structure. The Clustered Edition output differs because segment structure itself is changed. For example, the sg_below and sg_above threshold pointers for the segment are numbers that specify the positions of the two thresholds in the dbt_thresholds array.

## *dbcc dbtable* output

dbcc dbtable prints the dbt_thrmgr_info, which contains cluster threshold management data, and prints the segment structures in the table.

## *dbcc dbrepair* with *remap* option

dbcc dbrepair with remap:

- Rebuilds the disk map in dbtable
- Recalculates the segment free page counts
- Sets up the thresholds in each segment according to the threshold levels on the segment and the total free page count

You can use dbcc dbrepair with remap to specify either the segment number or segment name, but you can specify only one segment at a time to remap.

If you do not specify the segment to remap, Adaptive Server remaps all the segments in the database. You can use the optional parameters at the end of dbcc dbrepair to specify the segment for remapping.

dbcc dbrepair includes the option fixalloc; to use both remap and fixalloc, add remap before fixalloc. Adaptive Server version 15.1.1 allows you to specify the segment number or segment name at the end of this command. This example specifies the log segment for the pubs2 database:

```
dbcc dbrepair(pubs2,"remap","fixalloc", -1,"logsegment")
```

To remap the log segment in the pubs2 database without using fixalloc, enter:

```
dbcc dbrepair(pubs2, "remap",NULL, -1,"logsegment")
```

To remap all segments in the pubs2 database, use:

```
dbcc dbrepair(pubs2, "remap")
```

dbcc dbrepair with the remap option is used by sp_addsegment, sp_dropsegment, sp_modifysegment, sp_extendsegment, sp_logdevice, and sp_placeobject. For more information on these stored procedures, and on dbcc dbrepair, see the most recent version of the *Adaptive Server Enterprise Reference Manual: Commands*.

### *dbcc dbrepair* with *newthreshold* option

dbcc dbrepair with newthreshold:

- Loads thresholds in a database, from systhresholds to dbtable->dbt_thresholds.
- Sets up the threshold pointers in each segment according to the levels of the thresholds on the segment, and the segment free page count.
- Specifies the segment name or number, and reads all thresholds in systhresholds into dbt_thresholds. Only dbt_thresholds segment's thresholds are set up according to the thresholds that belong to the dbt_thresholds segment and the segment free page count.

dbcc dbrepair with newthreshold is used by sp_addthreshold, sp_dropthreshold, and sp_modifythreshold. For more information about these stored procedures and complete documentation of dbcc dbrepair, see the most recent version of the *Adaptive Server Enterprise Reference Manual: Commands*.

## Cluster interprocess communication

Cluster interprocess communication (CIPC) is a critical subsystem in the cluster architecture. Most cluster subsystems use CIPC to communicate with other instances in the cluster.

**Note** A high-speed CIPC network is necessary to gain acceptable Cluster Edition performance.

Tasks running on instances within the cluster use CIPC to communicate with other tasks in other instances in the cluster. CIPC information specified in the cluster input file and stored on the quorum device is used to connect all instances in the cluster. Within the input file, CIPC supports a primary and an optional secondary interconnect. The secondary interconnection, if present, is used for buffer-cache transfers, reducing the load on the primary interconnection.

# Recovery

The Cluster Edition provides failover recovery, during which time the database remains online and can continue to be used. However, if you request data that was modified by the failed instance, and must be recovered, the user process is blocked until the data is brought to a consistent state by the failover recovery process.

The Cluster Edition supports:

- Recovery that occurs at cluster start-up. This is the same as the nonclustered Adaptive Server recovery that happens when you start the server during a cold restart.
- Failover recovery. (In this document, “failover” refers to an instance failover, not a client connection failover.) This recovers data that a failed instance modified while other instances were using the same database in a shared-disk cluster.

Recovery begins when the cluster membership service detects a failure. The recovery event handler on the coordinator instance recovers the system databases. Then the surviving instances (including the coordinator instance) recover the user databases in parallel. One of the PCM threads on each instance performs the recovery task. Other threads on the coordinator instance, as well as other instances, pursue other activities.

If trace flag 3483 is on, distributed failover recovery is disabled, and user databases are recovered in serial order by the recovery event handler on the coordinator instance.

Processes trying to access other processes trying to access data modified by the failed instance are blocked until recovery of the failed instance is complete.

The Cluster Edition recovers user databases in the order specified by their database ids (dbid), and does not use any user-defined recovery order during instance failover recovery.

## Recovery algorithm

The recovery process in a nonclustered Adaptive Server is similar to that of the Cluster Edition:

***Table 9-1: Recovery steps for clustered- and nonclustered servers***

| Nonclustered recovery steps | Cluster Edition recovery steps |
|---|---|
| 1) Estimate recoverable log boundaries – the recoverable log is from the oldest active transaction recorded in the recovery checkpoint to the end of the log | 1) Estimate recoverable log boundaries (same as in nonclustered Adaptive Server). |
| 2) Analysis – scans the recoverable log forward, from the oldest active transaction to the end of the log, builds recovery information as incomplete transactions and so on. This information is used in the redo, undo, and post-undo passes. | 2) Analyze the database's log being recovered (same as in nonclustered Adaptive Server). |
| 3) Redo – scans the recoverable log forward, and re-executes operations specified by log records, as needed. | 3) Redo. In addition, the locks for incomplete transactions are acquired. |
| | 4) Reserve compensation log record (CLR) space to undo incomplete transactions on the failed instance. |
| | 5) Release all locks acquired by the failed instance prior to the failure, except for those acquired at step 3 for incomplete transactions. |
| | 6) Fill free space information (threshold manager recovery). |
| 4) Undo – works from the end of the log back to the beginning of the oldest incomplete transaction, and undoes operations specified by log records for incomplete transactions. For each transaction, Adaptive Server logs a CLR. | 7) Undo. In addition, logical locks for incomplete transactions are released as completed. |
| 5) Post-undo – logs a checkpoint record on the database, fills free space count recovery information (threshold recovery), and clears caches. | 8) Post-undo, including flushing all dirty buffers on the recovery instance. The Cluster Edition does not perform a checkpoint after instance failover recovery. |

**Note** See the *System Administration Guide: Volume 2*, *Chapter 11*, for information about the specific steps needed to complete the passes in this algorithm.

### Single transaction log

There is only one log per database for the cluster, and it is logically partitioned. A log record marker for the Cluster Edition contains the instance id, which it used internally to scan log records logged by specific instances, especially during failover recovery. A checkpoint log record always has an instance id with a value of zero, so it is always included in the log scan, regardless of the instance id specified in the log scan.

## Distributed checkpoints

Checkpoints control recovery in both nonclustered Adaptive Server and Cluster Edition systems, but the process for which they supply the transaction information differs. In the Cluster Edition, the oldest active transaction found is the oldest active transaction across the entire cluster, and dirty buffers are flushed on all clustered instances.

In both types of processing, Adaptive Server finds the oldest active transaction information at the time the checkpoint record is logged, uses it as a cutoff point, and writes the checkpoint record to the log. The dirty buffers are then flushed, up to the current flush sequence number. This checkpoint record is registered as a recovery checkpoint only after all the dirty buffers are flushed.

## Quorum device heartbeat

Periodically, each instance in the cluster “checks in” with the quorum device, creating a quorum device heartbeat that enables instances to monitor the status of the cluster. Instances use the device heartbeat to:

- Determine if the quorum device is accessible. If an instance fails to write the heartbeat to the quorum device, the instance may have has lost its storage area network (SAN) link, or it may be blocked from the cluster devices.
- Read the heartbeat values from the quorum disk while it is starting. If the instance detects no changes in the heartbeat after a configured period of time, the instance determines the cluster is not running.

Instances do not use the quorum heartbeat to detect instance failure.

## Configuring the quorum device heartbeat

You can configure how often the heartbeat happens and the number or times an instance attempts to detect the quorum heartbeat before it assumes the quorum device is not functioning with the quorum heartbeat interval and the quorum heartbeat retries configuration parameters, respectively.

quorum heartbeat interval

The quorum heartbeat interval configuration parameter specifies the number of seconds between quorum heartbeats. The default is 5, so each instance writes a quorum heartbeat once every 5 seconds. The minimum value is 1 second and the maximum value is 60 seconds.

Most sites need not tune this configuration parameter. Setting quorum heartbeat interval to a lower number increases the heartbeat overhead, but speeds the detection of a lost disk link, so instances terminate more quickly if they are blocked or have lost their SAN link. Setting quorum heartbeat interval to a higher value reduces the heartbeat overhead, but delays the detection of a lost disk link. The amount of overhead caused by the heartbeat depends on the performance of your disk subsystem.

quorum heartbeat retries

The quorum heartbeat retires configuration parameter specifies the number of times an instance attempts to detect a quorum heartbeat before determining that the quorum device is no longer running, and exiting. The default is 2 (the instance terminates after the third consecutive quorum heartbeat failure because the first two failed). The minimum value is 0, indicating that the instance should terminate upon the first quorum heartbeat failure, and the maximum value is 32,768.

Tuning this to a lower number causes an instance to fail over more quickly when access to the quorum device is lost, potentially improving application recovery times. Tuning this to a higher number degrades application recovery, reducing the chances that a transient disk access problem causes an instance failure.

# Using Infiniband

Adaptive Server 15.0.1 and later supports Infiniband (IPoIB) for internal communication between nodes in a cluster.

**Note** Infiniband is not supported on HPIA systems for Adaptive Server CE version 15.0.1 ESD #4.

To use InfiniBand:

- Configure the host channel adaptor (HCA)
- Use a InfiniBand software stack
- For Linux, the Cluster Edition requires OFED 1.2 from OpenFabrics
- InfiniBand software comes with Solaris 10

**Note** Solaris 10 includes InfiniBand. See your operating system documentation for information about installing and configuring InfiniBand.

## Setting the buffer space

The Cluster Edition must have sufficient buffering space to ensure adequate performance, particularly on faster interconnects. The default values for buffer space on Linux and Solaris systems are too small for gigabit Ethernet and InfiniBand. You must modify them to ensure adequate networking performance.

### Configuring buffer space

Linux

Use a command similar to the following to set the buffer space to an appropriate size on your system:

```
/sbin/sysctl -w net.core.rmem_max=value
/sbin/sysctl -w net.core.wmem_max=value
```

On most platforms, the default value for rmem_max is set to about 128KB, which is too small for the Cluster Edition. Increase rmem_max to the largest value your platform allows (1 megabyte is a good starting value):

```
/sbin/sysctl -w net.core.rmem_max=1048576
```

Solaris

Use a command similar to the following to set the buffer space to an appropriate size on your system:

```
ndd -set /dev/udp udp_max_buf value
```

## Configuring InfiniBand in a cluster

After you configure the host channel adapter (HCA), the */etc/hosts* file includes a corresponding host name and IP address. Use this host name or IP address when Adaptive Server Plug-in or sybcluster asks for this information during the cluster configuration. If you manually configure the cluster, add the IP address.

CHAPTER 10

# Troubleshooting

This chapter provides instructions for troubleshooting common errors.

# Verifying the cluster environment

Many errors that occur using the Cluster Edition result from configuration problems in the cluster environment. Sybase recommends that before you configure your cluster:

- Verify that you have set your environment variables by sourcing the *SYBASE.csh* or *SYBASE.sh* file located in *$SYBASE*.
- Run dataserver -v from each node to verify that all required libraries are installed on the host.

  If any system libraries are missing, you see an operating system error, and the data server version number does not appear. Correct this problem before proceeding. If dataserver displays the version string without error, you can assume that all required system libraries are installed.
- Verify that each node in the cluster can read from and write to each database device. Use the operating system ls -l command to test whether you are able to read from and write to the devices using the dd operating system utility.

  To test the readability of a device:

  ```
  dd if=<device path> of=/dev/null count=x
  ```

  You should get a result similar to:

  ```
  %dd if=/dev/raw/raw123 of=/dev/null
  count=10
  10+0 records in
  10+0 records out
  ```

  You can use the dd utility to test the writeability of your devices as well. However, you should do this only if there is no data in the devices to preserve.
- Run the ping utility to verify connectivity among all nodes. From each node, attempt to ping the host name or network address of every other node. Do this for each network to be used. For example, if your configuration uses a public and two private networks, verify that ping succeeds for all combinations of node and network address.

  Use the sybcluster 'show cluster config' parameter to determine the private interconnect addresses used by each instance. For example, if your cluster contains nodes node1 and node2, sybcluster displays information similar to this:

```
SYBCE> show cluster config
```

```
** Cluster configuration for "SYBCE" **
    Interface Path "/sybce"
    Trace Flags:
    There were are no trace flags.
    Maximum Instances "4"
    Quorum "/dev/raw/raw23"
    Master Device "/dev/raw/ra24"
      logfile INSTANCE1 /sybce/ASE15_0/install/GATEST_INSTANCE1.log
      run_parameters INSTANCE1
      logfile INSTANCE2 /sybce/ASE-15_0/install/GATEST_INSTANCE2.log
      run_parameters INSTANCE2
    Primary Interconnect "udp"
      Server[1] INSTANCE1 node1_priv 49152 49171
      Server[2] INSTANCE2 node2_priv 49172 4919
```

This cluster includes interconnect network addresses `node1_priv` and `node2_priv`. From node1, execute ping node2_priv to verify that the address of the private network on node2 is accessible from node1. From node2, execute ping node1_priv to verify that the private network on node1 can be reached from node2.

If the ping command fails or error messages indicate a problem with the private network, check:

- The information contained in the */etc/hosts* file
- The condition of the network cables, routers, or switches used by the private networks
- The names or IP addresses specified in the cluster configuration reported by the sybcluster "show cluster config" command

# Restarting the cluster using a *dataserver* binary from an earlier version

The Cluster Edition does not start, and writes this message to the error log "`Cluster is running with message version` ***y***`. This version of ASE requires message version` ***x***," if:

- You attempt to use different versions of the dataserver binary in the same running cluster.
- You apply an EBF or ESD without first gracefully shutting down a cluster using a previous version of the dataserver binary.

Resolving the problem

Start an instance of the cluster using the old version of the dataserver binary, and then issue `shutdown cluster`.

Restart the cluster using the new version of the dataserver binary.

If the previous version of the dataserver binary is not available, you can resolve this error message by re-creating the quorum device.

**Warning!** These steps bypass Adaptive Server's safety checks, and you must make sure no instances are running while you perform them.

1 Verify that all members of the cluster are shut down.

2 Extract the cluster input file information using qrmutil:

```
$SYBASE/ASE-15_0/bin/qrmutil --quorum_dev=path_to_quorum
--extract_config=quorum.out
```

3 Start the Cluster Edition using the new dataserver binary, and rebuild the quorum device:

```
dataserver --quorum_dev=path_to_quorum --instance=instance_name
--buildquorum=force --cluster_input=quorum.out
```

4 Shut down the cluster

5 Start the cluster using normal procedures.

**Warning!** The --buildquorum or --cluster_input dataserver parameters are used only for these steps. Do not use them during subsequent cluster or instance restarts.

# Errors accessing disk devices

All disk devices used by the Cluster Edition must be configured to be accessible from all the nodes in the cluster. The paths to these devices must be the same on all nodes, and the account used to start the cluster must have permission to read and write to all of the disk devices.

If the Cluster Edition reports that any of the devices cannot be accessed, verify from each node in the cluster that,

- The device paths specified when configuring the cluster are accessible from all nodes in the cluster.
- The account used to start the cluster has permission to read and write to these devices.
- The device paths are the same on all nodes in the cluster.
- Any symbolic links used to refer to the devices are correct.

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

See “Verifying the cluster environment” on page 176.

## Verifying the cluster is down

The sybcluster utility may not be able to determine whether or not the cluster is running if the cluster has crashed and left some status information in the quorum device in an inconsistent state. If you are uncertain about the status of the cluster, analyze each instance in the cluster to determine the cluster’s status.

1 Use isql to log in to each instance in the cluster.

2 Use sybcluster 'show cluster status' to view the status of each instance in the cluster.

3 If sybcluster does not report that the state of all instances is `'Up'` and that the `heartbeat` is `'Yes'`, the instance may be down.

4 Use the UNIX 'ps' command on each node in the cluster to determine whether processes representing the dataserver program are running for each instance configured for that node.

5 After starting the first instance, issue sybcluster start cluster to start all remaining instances in the cluster.

If you determine the instances are not running, unlock the cluster and restart:

```
start instance instance_name unlock
```

# Creating cluster using *sybcluster* fails with error -131

The sybcluster create cluster command may fail with error -131 and issue a message stating that the parent directory in which the raw devices are defined cannot be accessed:

```
INFO - Choosing the first instance to be created using the connected agent...
ERROR - Parent directory access error. The parent directory /dev/rdsk for the
device can not be accessed.  Please change the protection on the device and try
again.
INFO - Create cluster error: -131
```

You may have incorrectly spelled the name of the raw device. Check the complete name of each raw device and verify that it is correct.

# Cluster creation fails leaving files in $SYBASE directory

If the sybcluster create cluster command or the Sybase Central ASE plug-in Create Cluster Wizard terminates with an error condition, some files may be left in the cluster installation directory. Remove these files before attempting to create the cluster again:

1. Remove any entries for the cluster in the interfaces file in the installation directory.
2. Remove Unified Agent plug-in directories for each node from these locations:

```
$SYBASE/UAF-2_5/nodes/node_name/plugins/cluster_name
```

3. Remove resource files with names ending in **.res* from the *$SYBASE* directory.
4. Remove cluster definition files with names ending in **.inp* from the *$SYBASE* directory.
5. Remove the intermediate *RUN_instance_name* file from *$SYBASE/ASE-15_0/install*.
6. Remove any error log files from *$SYBASE/ASE-15_0/install*.
7. Remove any instance configuration files from *$SYBASE*.

Only steps 1 and 2 are required before you can initiate another cluster creation operation. However, perform the other steps to reduce the number of files in the installation area.

## Unified Agent starts but *sybcluster connect* fails

The sybcluster 'connect' command may fail because of incorrect network configuration information during the Unified Agent startup or when executing the sybcluster command.

- Examine the Unified Agent log files for each agent in the cluster to determine whether any errors are reported opening the RMI listener for the agent. The agent log files are:

  *$SYBASE/UAF-2_5/nodes/node_name/log/agent.log*

- Specify the correct node names and listening port numbers for the -F parameter when starting sybcluster:

  ```
  sybcluster -U uafadmin -P -C MYCLUSTER
  -Fnode1:1234,node2:1234
  ```

## Disk devices in use

On Linux, some devices are, by default, bound to a disk. If you attempt to create a cluster and inadvertently specify one of these devices, disk init fails and the Create Cluster wizard or sybcluster cannot create the cluster.

Before you start the Create Cluster wizard or a sybcluster session, check to make sure the disk device is available.

See “Verifying the cluster environment” on page 176.

# Instances fail to join the cluster

Instances may fail to join the cluster because of problems with the private networks used to support interconnect communication. If one or more instances in the cluster fail to start:

- Examine the error logs for those instances that could not join to see if there are any messages indicating the network communication has failed. The error logs are located in *$SYBASE/$SYBASE_ASE/install/instance_log*
- Verify that all private networks used on the nodes used by the cluster are accessible from all nodes in the cluster. Use the ping utility to do this. See "Verifying the cluster environment" on page 176.
- Check for errors in the UAF agent logs.

# Private interconnect failure

The cluster does not start if a private interconnect is not configured correctly. Use the operating system ping command to validate the private interconnect is working. If ping does not work, see your system administrator to enable the interconnection communication between various nodes.

See "Verifying the cluster environment" on page 176.

# Client connection failover fails

This error occurs when a client connection failover fails when the client connects to the cluster using IP addresses and the nodes are not included in the local */etc/hosts* or DNS.

Adaptive Server sends the failover instance addresses as they are listed in the cluster's interfaces file. If the cluster's interfaces file lists the instance network addresses as host names, Adaptive Server returns the host names to the client applications. However, the client application uses DNS or the */etc/hosts* file to resolve the names of the hosts for the cluster instances, so if the clients do not have the host names in their */etc/hosts* file or DNS server, failover is unsuccessful.

Verify that all nodes in the cluster are listed in the client systems's DNS server or the client system's */etc/hosts* file.

Try using IP addresses rather than host names in the cluster's interfaces file.

## sybcluster cannot connect if all connections use SSL

If all Adaptive Server listening ports use SSL, Adaptive Server issues this error message:

```
2008-03-20 10:42:46,260 ERROR [Timer-6]
GA1:GA1_1:SQLConnect:270:Login Failure - The user "sa"
and the entered password is not authorized to connect
to the cluster.
```

Verify that there is a non-SSL connection for each instance in the cluster and that this connection is included in the interfaces file Unified Agent uses.

## jConnect sample disables HA

The sample jConnect connection string for the Cluster Edition included in the *What's New in ESD #12* document for Open Client, Open Server, ODBC, jConnect, and ADO.NET for Adaptive Server version 15.0 is incorrect.

The sample connection string in the document is:

```
URL="jdbc:sybase:Tds:server1:port1,server2:port2,...,s
erverN:portN/mydb?JCONNECT_VERSION=6&PACKETSIZE=1024&D
YNAMIC_PREPARE=true&REQUEST_HA_SESSION=true"
```

This string fails because the `JCONNECT_VERSION=6` parameter emulates the high availability companion server functionality. The JDBC driver causes a client Java exception when the client tries to use the jConnect driver's Failover property but does not have a hafailover server specified in the connection string.

This is the correct connection string:

```
URL="jdbc:sybase:Tds:server1:port1,server2:port2,...,s
erverN:portN/mydb?&PACKETSIZE=1024&DYNAMIC_PREPARE=tru
e&REQUEST_HA_SESSION=true"
```

# PC-Client installation – java.lang.NoClassDefFound Error

Occurs on the Windows platform when you unpack the PC Client tar file using the MKS tar utility rather than the Winzip utility. On Windows platforms, the MKS tar utility truncates full path names, resulting in missing files.

This problem occurs on UNIX platforms if you do not use the GNU tar utility to untar the Adaptive Server installer.

For Windows platforms, always use the Winzip utility to unpack the installer. For UNIX platforms, always use GNU tar utility to untar the installer.

# The cluster entry "*name*" did not contain any servers

If the quorum device is not accessible to the UAF agents, the following errors may occur when you issue the sybcluster start cluster and start instance commands:

```
start cluster
ERROR - The cluster entry SDCDEMO did not contain any servers
start instance INSTANCE1
ERROR - The cluster entry SDCDEMO did not contain any servers
```

- The UAF agent may not have permission to read the quorum device. Verify that you started the UAF agent under the proper user account and that this account has permission to both read and write the quorum device.
- sybcluster and the UAF agent are using an incorrect path to the quorum device on one or all nodes in the cluster. If you used symbolic links, verify that all links are correct on all nodes.
- Another process is preventing the UAF agent from reading the quorum device. This could be caused by functional problems or configuration errors in the disk storage system.

# After password change, sybcluster cannot manage the cluster

sybcluster uses the UAF agent to connect to and perform operations on the cluster (for example, shutdown cluster). The UAF agent must log in to the cluster to do this. To log in, the UAF agent must use the correct sa login and password.

The sa login and password are stored in encrypted form in the UAF plug-in *configure* file for the cluster. Use sybcluster set cluster login to change the login and password that UAF uses to connect to the cluster.

Any time you change, Adaptive Server password, you must also change the UAF agent login.

Perform one of the following to change the login and password that UAF uses to connect to the cluster:

- Use sybcluster set cluster login to set the UAF login, password, or both. Connect to the cluster before issuing this command

  The syntax is:

  set cluster login sa-login [password sa-password

- Use the ASE plug-in to modify the UAF plug-in:

  a Connect to the cluster.

  b From “Server Instances,” left-click an instance listed in the tree view pane.

  c Select “Agents Attributes” from the right pane.

  d Click the “password” row.

  e Enter a new password for the agent managing the current instance.

  f Repeat steps c – e for each instance in the cluster

- Edit the UAF configuration files.

  **Note** You must shut down all cluster UAF agents before editing the *agent-plugin.xml* file.

  The UAF Adaptive Server login and password are stored in:

  ```
  $SYBASE/UAF-2_5/nodes/<node name>/plugins/
  <cluster name>/agent-plugin.xml
  ```

For example:

```
<set-property property="ase.user" value="sa" /><set-property
property="ase.password"
value="REVTe1NZVUFGfWNvbS5zdW4uY3J5cHRvLnByb3ZpZGVyLlN1bkpDRXtTWVVBRn1j
UEZPSkJoTTZ2QT0=" />
```

**Note** If the password tag is missing, add following steps a – f, above.

Enter the password in clear text. passencrypt generates an encrypted text string. Enter this *entire* string between the quotation marks in the `password` tag in the *agent-plugin.xml* file

Generate the encrypted password:

```
$SYBASE/UAF-2_5/bin/passencrypt
```

# Agent “cannot be found”

sybcluster show agents does not display all agents.

Often, host systems names in a cluster are similar. If you use sybcluster -F to specify the connections to each of the agents in your cluster, verify that each host system is spelled correctly, that you have specified port numbers, and that the port numbers are correct

# Sybase Central cannot register the AMCP plug-in

If you get this message when you attempt to register the AMCP plug-in (*amcplugin.jpr*):

```
Could not read manifest.file
```

The installer has renamed the *amcplugin.jar* file to *amcplugin.jar.installed* to avoid overwriting existing files.

When you attempt to run *registerAMCP* from the command line, you see:

```
Error: Unable to find the AMC Plugin binary.
Please check that $SYBROOT has been set correctly and
that the file
```

```
/AMCP/lib/amcplugin.jar' exists.\n
```

Rename *amcplugin.jar.installed* to *amcplugin.jar*.

# UAF plug-in register error

You may see this error because of an existing cluster parameter value that is inconsistent with a newer configuration parameter:

```
2008-06-16 14:05:16,051 ERROR [main] Failed to register plugin
com.sybase.ase.cluster_15.0.1. Class
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin not found. Ignored.
java.lang.ClassNotFoundException:
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin
2008-06-16 14:05:16,052 INFO [main] Finished loading primordial services.
2008-06-16 14:05:16,063 WARN [main] Bootstrap completed with 1 error(s):
2008-06-16 14:05:16,064 WARN [main] Failed to register plugin
com.sybase.ase.cluster_15.0.1. Class
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin not found. Ignored.
java.lang.ClassNotFoundException:
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin
```

If you get this error, it is likely that:

- The UAF agent will not start, and the *agent.log* file displays `Bootstrap completed with x error(s):` near the end of the file.
- You have created, dropped, and reconfigured the same cluster a number of times.

This error message appears in the *agent.log* file or in the terminal window if you did not execute the *uafstartup.sh* as a background process. Either:

- Clean up the older parameters that may be introducing the error, or

- Remove the *cluster* directory in *$SYBASE_UA/nodes/<node_hostname>/plugins*. You can permanently delete the folder or relocate it outside of the directory structure.

  **Note** Do not remove the **snmp* and **sysam* folders.

After you have performed either of these steps, restart the UAF agent. You may have to manually redeploy the plug-in.

**Note** If the steps described above removed the cluster's *plugin* diretory from one of the nodes, use deploy plugin to re-create this directory. The cluster requires that each node of the cluster include a *plugin* directory. See "deploy plugin" on page 331.

# Data on disk unavailable: problems affecting database creation

Disk labels are stored in block 0 of every disk. If raw data slices used for device creation start on block 0, they may overwrite the disk label and make data on the disk inaccessible.

Do not create raw disk slices that start on block 0.

# Access permission to devices is denied after enabling I/O fencing

On Solaris systems, the UNIX user starting the Cluster Edition must have SYS_DEVICES inheritable privileges to access the raw devices (*/dev/raw/raw#* or */dev/rdsk/c#t#d#s#*) used for databases and quorums.

The SYS_DEVICES privilege grants Adaptive Server the ability to execute SCSI-3 PGR commands used in I/O fencing. The UNIX user can be granted temporary or permanent SYS_DEVICES privileges:

- To grant temporary SYS_DEVICES privileges, add SYS_DEVICES privileges to the inheritable privilege set of the current user's shell process:

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

- To grant permanent SYS_DEVICES privileges, use the usermod UNIX command to add SYS_DEVICES to the inheritable privilege set of any user:

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

# sybcluster cannot find interfaces file

sybcluster may report an error message, stating that it cannot find the interfaces file in a nonexistent directory:

```
ERROR - The interfaces file /remote/ase_cluster/sdclinux/UAF-2_5/bin/interfaces
could not be found.
```

This error generally occurs when one or more UAF agents cannot access the quorum device. This may occur because:

- The quorum device does not exist at the configured location.
- The quorum device's permissions do not allow the UAF agent to read or write data.

To resolve this issue, verify that:

1 You sourced the *SYBASE.sh* or *SYBASE.csh* files before starting the UAF agent on each node in the cluster.

2 The quorum device exists and that the login that was used to start the UAF agent has read and write permissions on it.

# IBM errors

This section describes errors you may encounter when running the Cluster Edition on the IBM AIX operating system.

## Asynchronous I/O not enabled

If asynchronous I/O is not enabled on IBM AIX when you attempt to start the Cluster Edition, the dataserver binary issues this error message and the Cluster Edition does not start:

```
exec(): 0509-036 Cannot load program dataserver because of the following errors:
0509-130 Symbol resolution failed for /usr/ccs/lib/libc.a[aio_64.o] because:
0509-136   Symbol kaio_rdwr64 (number 1) is not exported from dependent module
/unix.
0509-136   Symbol listio64 (number 2) is not exported from dependent module
/unix.
0509-136   Symbol acancel64 (number 3) is not exported from dependent module
/unix.
0509-136   Symbol iosuspend64 (number 4) is not exported from dependent module
/unix.
0509-136   Symbol aio_nwait (number 5) is not exported from dependent module
/unix.
0509-136   Symbol aio_nwait64 (number 6) is not exported from dependent module
/unix.
0509-136   Symbol aio_nwait_timeout (number 7) is not exported from dependent
module /unix.
0509-136   Symbol aio_nwait_timeout64 (number 8) is not exported from dependent
module /unix.
0509-026 System error: Error 0
0509-192 Examine .loader section symbols with the 'dump -Tv' command.
```

See your IBM AIX operating system documentation for information about enabling asynchronous I/O.

## Incorrect permissions on device

If you do not have permission to manage raw devices on IBM AIX, the operating system issues this message when you attempt to start the Cluster Edition and the cluster does not start:

```
dopen: open '/dev/device_name', Not owner
```

- Grant the correct permissions (one of: PV_ROOT, PV_SU_, or PV_KER_RAS):

  ```
  setsecattr -p iprivs=+PV_KER_RAS $$
  ```

  These permissions are inherited by the Cluster Edition process when it starts.

- Grant a non-networked (NIS) user permission to run processes that manage devices:

  a Create the user:

  ```
  mkuser sybase
  ```

  b Create the role:

  ```
  mkrole authorizations=aix.device.manage.change
  role_disk_access
  ```

  c Assign the role to a user:

  ```
  chuser roles=role_disk_access
  default_roles=role_disk_access sybase
  ```

## Another machine using device

If the Cluster Edition does not have permission to access a device because a process on another machine is using the device, and the device can be used only by a single machine at a time, the operating system issues:

```
The IBM AIX SDC dataserver may fail to run with one of the following errors:
Quorum library error 1: Failed to open quorum device '/dev/disk_name'.  OS error
16, 'Device busy'
```

Or:

```
dopen: open '/dev/disk_name', Device busy
```

The device must allow multiple servers simultaneous access. The database devices must have shared reservations where the reservation key is the instance ID defined in the cluster configuration file (for example, where ID = 1):

Change the device access restrictions for the quorum device on each machine in the cluster:

```
chdev -l device_name -a reserve_policy=no_reserve
```

For example, if a disk device is named */dev/hdisk1*:

```
chdev -l /dev/hdisk1 -a reserve_policy=no_reserve
```

Run this command for each database device on each instance in the cluster to change the device access restrictions for a database device:

```
chdev -l device_name -a PR_key_value=instance_ID -a reserve_policy=PR_shared
```

For example, to change the device access restrictions for */dev/rhdisk2* on instance 1:

```
chdev -l hdisk2 -a PR_key_value=1 -a reserve_policy=PR_shared
```

## Error running chdev

You may see this error when running the chdev command:

```
Method error (/usr/lib/methods/chgdisk):
          0514-047 Cannot access a device.
```

The device is currently in use. Shut down all processes accessing the device.

CHAPTER 11

# Administering Clusters with the Adaptive Server Plug-in

The Adaptive Server plug-in for Sybase Central allows you to perform the administrative tasks for the cluster. For example, creating a cluster, adding an instance, starting and stopping a cluster or instance, creating or modifying logical clusters, administering workload management.

The ASE plug-in is used instead of the command line method.



For more information about using Sybase Central and the ASE plug-in, see Chapter 4, “Introduction to the ASE plug-in for Sybase Central,” in the *System Administration Guide*.

## Managing a shared-disk cluster

The ASE plug-in allows you to manage the shared-disk cluster environment from within Sybase Central.

You must have a Unified Agent running with a cluster agent ASE plug-in for all of the cluster management functionality to be available through the ASE plug-in. See “Enabling Unified Agent functions in the ASE plug-in” on page 195 for more information.

## Connecting to a cluster

When you start 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 next to the cluster name.

If the cluster managed by ASE Plug-in is not running, a red square appears in the server icon. If neither triangle nor square indicator is displayed, see the instructions for configuring ASE Plug-in and Unified Agent at "Starting a cluster" on page 200.

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 this 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 the Find button. 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, see "Changing server discovery settings" on page 195 for more information.

❖ **Connecting to a cluster**

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 for Linux, Solaris, IBM AIX, and HP-UX, and from *sql.ini* from Windows), or type in the host and port of a cluster node.

5 (Optional) Specify the host and port of an instance within the cluster.

6 Click OK.

❖ **Shortcuts**

- Right-click the cluster icon and select Connect.

Once you connect to the cluster, the cluster icon changes from grey to blue.

## Disconnecting from a cluster with the toolbar

1. Select the icon of the cluster from which you want to disconnect.
2. Select File | Disconnect.

Shortcuts

- Right-click the shortcut icon for the cluster and select Disconnect.
- Select the cluster from which you want to disconnect. Select Disconnect from the toolbar.

## Enabling Unified Agent functions in the ASE plug-in

1. Select Tools | Adaptive Server Enterprise | Preferences. On the Preferences tab, select: “Enable Unified Agent features,” “Check Server Status,” and “Use Agent Port Number.” The default UAF port is 9999. If you need to check UAF agents on a different port, enter this value here

   You can change the UAF port number; the default is 9999.
2. Click OK. If the cluster is monitored by the UAF agent, a red square appears on the cluster icon if the cluster is not running. A green triangle appears on the cluster icon if the cluster is running.

## Changing server discovery settings

You need not select a single discovery method. Server discovery searches all specified discovery methods.

1. Right-click the cluster’s name.
2. Select Connect.
3. From the Connect dialog box, select Settings.
4. Select the Server Discovery tab.
5. Select the discovery method.
   - 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.

See Chapter 2, "Installing and Configuring Unified Agent and Agent Management Console," in *Unified Agent and Agent Management Console Version 2.0 for Windows and UNIX*.

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

6 Click Add.

7 If you selected JINI in the previous step:

- Select the host of the JINI server
- Select either the default host and port or enter new ones

8 Click OK.

9 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, Build Date.

d Select the condition: contains, does not contain, is, is not, starts with, ends with.

e Enter the condition string you want filtered.

f Click OK.

10 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.

11 If you are using LDAP servers, select the LDAP pane:

a Select the LDAP server name.

b Use the gauge to set the search timeout period.

c Enter the user name and password to log in to the nodes.

d Select the nodes you want to log in to in the Select Cluster Nodes box.

e Click OK.

## Displaying cluster properties

To view the cluster properties, right-click the cluster name and select Properties. The ASE plug-in displays the Server Properties dialog box, which includes the General, Configuration, Log Space, Job Scheduler Server, Agent, Server Log, and Localization tabs.

### General properties tab

The General Properties tab displays this information about the Cluster Edition:

- Type – the edition of Adaptive Server
- Version – version of the software.
- Release Type – whether the release is a Beta or Production version of the software.
- Platform – machine that is running the node.
- Operating System – operating system running on the node.
- Build option – options specific to the currently running version of Adaptive Server.
- Build Date – date the dataserver binary was built.
- Edition – currently running edition of Adaptive Server.
- License – current status of the Adaptive Server license. Select Details for more information.
- Character Set – currently configured default character set.
- Language – currently configured language.
- Sort Order – currently configured sort order.
- Status – status of the server, running or down.
- ASE home – location of the release directory.

- ASE log file – location of the log file.

## Configuration tab

In the Cluster Creation Wizard, each instance uses the same server configuration file (*server_name.cfg*) to determine its configuration. By default, all instances in the cluster use the *cluster_name.cfg* file for cluster configuration. However, you can specify a different configuration file for when you configure the instance, allowing you to set different configuration values for different instances.

You can use the Configuration panel for clusters and instances.

To view the current configuration settings, right-click the cluster or instance name and select Configure. You can also select Properties and then select the Configuration panel.

For details about configuration parameters, see the *System Administration Guide*. For a discussion of configuration issues to consider when determining optimal settings, see the *Performance and Tuning Guide: Basics*.The following rules govern who can set configuration parameters:

- Logins assigned the system security officer ("sso_role") role can reset:
  - allow updates
  - audit queue size
- The default character set id parameter is automatically set during the cluster installation and cannot be reset from within Sybase Central.
- Logins assigned the system administration ("sa_role") role can reset all other parameters.

### Parameters that require restart

Some configuration parameter values are dynamic, which means the parameter takes effect as soon as you reset the value. Others do not change until you restart the cluster (these are called static parameters). The ASE plug-in indicates whether the parameter requires a restart when you select the parameter name.

#### Dropping instance-specific configuration parameters

The Configuration panel for a clustered instance includes a Drop button, which is enabled only for instance-level configuration values, and does not appear on the cluster configuration property tab. Dropping a parameter means you drop it as an instance-specific setting independent of the cluster-wide value.

Drop is disabled until you change a value and select Apply. The next time you select this configuration parameter, Drop is enabled, and you can select it to drop the configuration parameter.

#### Setting configuration parameters

The Cluster Edition includes global and instance configuration parameters. Global configuration parameters affect the entire cluster, while instance configuration parameters affect only the instance on which they are set. To set global configuration parameters, open the Configuration tab for the cluster and select the name of the cluster, then select the configuration parameters to reset.

By default, instances use the global configuration values unless an instance setting overrides them.

❖ **Setting configuration parameters for an instance**

1 Right-click the instance name you want to configure.

2 Select Configure (or select File | Properties, then click the Configuration tab).

3 Select the functional group to display, or select “All.”

4 Select the parameter you want to update. For a brief description of the selected parameter, read the Explanation box.

5 Enter the new value in the Value column of the table.

6 Click OK (or Apply if you are changing multiple configuration values).

- If the parameter takes effect immediately, it is listed in the Value column.
- If the parameter requires you to restart Adaptive Server, it is listed in the Pending Value column.

## Log Space tab

The Log Space panel displays this information about the current log space for the cluster:

- Database (Instance) – name of the log space. If the log is specific for an instance, the instance name appears in parentheses. If there are no parentheses, the log is for the cluster.
- Total (MB) – total amount of log space available, in megabytes.
- Used (MB) – amount of log space currently used, in megabytes.
- Free (MB) – amount of free space available for the log, in megabytes.
- Used – percentage of the total space currently used.

### Job Scheduler Server tab

The server you designate as a Job Scheduler server must have Job Scheduler installed on it. See the Job Scheduler *User's Guide* for more information on installing Job Scheduler.

### Localization tab

The Localization tab displays the current values for default language, charset, and sort order. You can change the default values and add or remove languages.

## Starting a cluster

The ASE plug-in must have the unified agent features enabled to start the cluster. Clusters that are administered by an agent display a red square (if the cluster is not running) or green triangle (if the cluster is running) on the server icon by the cluster name. To start a cluster that is not running:

1 In the left pane of the tree view, right-click the cluster shortcut icon and select Start.

2 Enter the Unified Agent login and password with administrative access to start the cluster.

3 A messages log window opens when the agent starts the server. The OK button is enabled when the cluster start process is completed.

4 The red square on the cluster icon turns to a green triangle, indicating the cluster is running.

## Shutting down a cluster

Shutting down a cluster that is not connected:

1 Right-click the cluster icon to shut down.

2 Select Shut Down.

3 Click Yes to confirm the shut down.

Shutting down a cluster that is connected:

1 In the left pane of the tree view, right-click the cluster icon and select Shut Down.

2 Check the boxes to:

- Shut down the cluster after processes have finished, and be notified if shutting down takes longer than one, five, or ten minutes, or
- Shut down the cluster immediately.

3 Click Yes to shut down the cluster.

## Dropping a cluster

Dropping a cluster allows a user to undo all the steps he or she performed to create a cluster. Dropping a cluster is different from removing a cluster from a server group. You can only drop clusters that are shut down.

---

**Note** You remove a cluster completely when you drop it. It is no longer available for restart.

---

To drop the cluster:

- Right-click the name of the cluster you want to drop.
- Select Drop Cluster from the list.
- Enter the login for the managing agent (usually, uafadmin).
- Select OK.

## Removing a server group

Removing a server group removes the server group's cluster entry from the ASE plug-in. The cluster is unaffected.

To remove a cluster from the default group:

1 Right-click the cluster name and select Remove from Default.

2 Confirm the deletion by selecting the cluster name from the Confirm Delete dialog box and selecting Yes.

## Displaying the status of a cluster

If the cluster is managed by a unified agent, the status is provided by the agent whether or not you are connected, but a connected cluster shows more detailed information.

In the tree view, click the Server Instances folder to view the instances in the right pane. Status details include:

- Instance – the name of the instances within the cluster.
- ID – the numerical order of the instance in the cluster.
- State – the current state of the cluster, online or offline.
- Address – the address of the instance in the cluster.
- Start Time – the time the cluster started.
- Connections Active – the number of connections.
- Engines Online – the number of engines.

## Managing a clustered instance

This feature enables management of the instances in a cluster.

### Adding an instance to a cluster

Before you can add an instance, the max instances parameter must have room for more 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).

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 Instance 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.

5 Start the instance.

## Dropping an instance from a cluster

You must shut down an instance before dropping it from the cluster. After you drop an instance, all temporary database definitions for that instance are dropped, including bindings and group memberships.

**Note** You cannot drop the last remaining instance of a cluster.

1 In the right pane, right-click the instance to be dropped and select Delete.

2 Click Yes. The instance is dropped from the cluster.

## Starting an instance

Right-click the instance icon and select Start. A bar titled “Start in Progress” appears.

Once the instance is started, the state reads “online.”

**Note** If the instance takes longer to start than anticipated, you may need to manually refresh the folder to update the state of the instance.

## Shutting down an instance

1 Right-click the instance icon and select Shut Down.

2 Select whether to:

- Shut down the instance after all processes have finished, and be notified if shutting down takes longer than one, five, or ten minutes.
- Shut down the instance immediately.

3 Click Yes. When the shut down is complete, the new status of the instance appears.

## Creating shared database devices

A shared database device is accessible to all of the cluster instances. Select the Database Devices folder in the tree view

- Select Add Database Device to start the Add Database Device wizard.
- Follow the instructions in the wizard.
- Click Finish. The device appears in the right pane under the list of devices.

# Managing multiple temporary databases

In the Cluster Edition there are four variations of temporary databases: global system, local system, global user-created, and local user-created.

To view temporary databases, select the Temporary Databases folder:

- Group view – lists the temporary database groups. Only local temporary databases can participate in temporary database groups.
- List View – displays the global temporary databases (not the local temporary databases).

## Managing the local temporary databases

A local temporary database can be accessed only by the local instance. Other server instances within the cluster cannot access this temporary database.

To view the temporary databases:

1 Select the Server Instances folder

2 Select the name of the instance.

3 Select Local Temporary Databases

Right-click the temporary database name to configure and maintain your local temporary database. Select the option to:

- Open Interactive SQL – start a session with Interactive SQL.
- Check Consistency – follow the instructions on the wizard to run the database consistency checker (dbcc) on the temporary database.
- Checkpoint – select Yes to run a checkpoint on this database. Select Preview to view the currently running SQL.
- Display Statistics – follow the instructions on the wizard to run optdiag on the temporary database.
- Generate DDL – select Create Database DDL to view the currently running DDL. Select Exclude DDL to view of list of objects that you can exclude from the DDL. Check the objects to exclude and select OK.
- Delete – select Delete to drop this temporary database. Select Yes to confirm. Because you cannot delete the last system temporary database, the Delete option is disabled if this is the only temporary database left.
- Properties – select the appropriate pane for information about:
  - General – describes the type of database, the database owner (select Change to change the database owner), the date the database was created, the last time the transaction log was dumped, whether the database has a guest user, the type of data cache, the default database location, and whether to resynchronize the proxy tables.
  - Devices – lists the currently configured database devices. Select:
    - Add – to add a device. Select Data or Transaction Log on the Device Size window, and specify the device size. Click OK to confirm.
    - Remove – remove the device.
    - Edit – configure the device. The Device Size window lists the currently configured name, size, unused portion, current allocation, total space allocation, and allows you to add space to the device.
    - Move Log – specify a new location for the log device.
    - Properties – includes four panes describing general information, mirror device status, databases located on this device, and segment information.
  - Usage – details the space the database uses. Select the unit in which to display the information: Pages, KB, MB, or GB.

  - Transaction log – allows you to configure the bindings, log IO size, and segments that use last-chance thresholds.
  - Options – lists the database options you can set for this database. Check or uncheck the options and click OK.
- Active sessions – displays SPID and login information for the sessions assigned to this database.

## System temporary databases

You create the system temporary databases when you create the cluster or instances. You cannot remove global or local system temporary database. However, local system temporary databases are removed automatically when you delete the instance.

## Adding a user-created global temporary database

In the Cluster Edition, you can create global temporary databases (available cluster-wide to all instances) and local temporary database (available to an individual instance).

1 Navigate to Databases | Temporary Databases | List View.

2 Select Add Temporary Database wizard.

3 Follow the instructions outlined in the wizard, clicking Next to go on to the next set of instructions and Back to return to a previous instruction.

4 Click Finish when you are done. The temporary database appears in the right pane under the list of databases.

## Adding a user-created local temporary database

The user-created local temporary database can be accessed only by the owning instance:

1 In the right pane of the tree view, navigate to Server Instances | *Instance_name* | Local Temporary Databases.

2 Select Add Local Temporary Database.

3 Follow the instructions in the wizard to create a local temporary database.

## Adding temporary databases to a group

To view the database groups, select Group View from the Temporary Databases folder.

### Adding a group

Administrators create groups that contain local temporary databases. The default group is created by default. Sybase recommends that you create bindings on groups rather than on individual temporary databases for easier administration.

Select Group View from the Temporary Databases folder to view the Group folder.

1 From the tree view, navigate to Databases | Temporary Databases | Group view

2 Select Add Temporary Database Group

3 The ASE plug-in starts the Add Temporary Database Group wizard. Follow the instructions provided by the wizard to create a local temporary database group

### Group properties

Right-click the group name and select Properties. Sybase Central displays the Bindings and Databases panes.

Databases pane

The Databases pane displays all current temporary databases in the group.

To add a temporary database to the group:

1 Select Add.

2 From the Add Temporary Database screen, select the name of the temporary database you want to add.

3 Click OK.

To remove a temporary database from the group:

1 Select the temporary database name.

2 Select Remove.

3 Click Yes.

Bindings pane

The Bindings pane displays the application and login bindings for the current group.

Binding a new application.

1 Select Bind Application.

2 Enter the application name and Click OK.

Binding a login.

1 Select Bind Login.

2 From the New Login Binding screen, select the login you want to bind.

3 Click OK.

Removing a currently bound application or login.

1 Select the application or login name.

2 Select Unbind.

3 Click Yes to confirm.

You can bind applications or logins to temporary databases or temporary database groups.

Viewing the current bindings.

1 Right-click the temporary database or temporary database group.

2 Select the Bindings tab. The ASE plug-in lists your current bindings.

Unbinding a login or application.

1 Select the login or application from the list.

2 Click Unbind.

3 Select Yes to confirm.

Unbinding all logins and applications.

1 Click Unbind All.

2 Select Yes to confirm.

# Managing the workload

Use the workload manager to view, create, and manipulate logical clusters, workload profiles, load scores, and routes.

## Load profiles

Load profiles allow you to define the operating criteria for a logical cluster. These criteria are typically called "load score metrics," with the value associated for each criteria rolled into a "score" for each instance in the logical cluster that uses the load profile. You can periodically compare load scores for different instances within a logical cluster to detect when the workload is undesirably skewed to one or more instances, or determine if an instance is under-utilized.

Instances included in multiple logical clusters can be impacted by multiple load profiles, so take care when associating instances with multiple logical clusters and when defining, and applying load profiles.

---

**Note** The Cluster Edition includes two system load profiles: sybase_profile_oltp for OLTP environments and sybase_profile_dss for DSS environments. You cannot modify or delete system load profiles. However, you can duplicate them and modify the duplicates to create your own load profiles.

---

The load profile status reports the:

- Name – the name for the load profile configuration.
- Type – the load profile type: system or user.
- Metric weights – relative weight assigned to each metric in the load profile. Metrics include:
    - User connections – displays the weight of users connected to the particular load profile.
    - CPU busy – displays the weight of CPUs that are currently busy.
    - Run queue length – displays the weight of the run queue.
    - IO load – displays the weight of the I/O load.
    - Engine deficit – displays the weight of the engine deficit.
    - Users – displays the weight for a metric the user chooses to measure.

- Thresholds – configured difference (as a percentage) in the load between two instances in a logical cluster at which point the following can occur:
  - Login redirection – used for connection-time load balancing and routing connections to a logical cluster. If necessary, an instance directs a client to stop the current login attempt and try connecting to instances it supplies as a list of available network addresses.
  - Dynamic migration – (also known as the hysteresis value) displays the dynamic migration configuration.
- Minimum load score – load score necessary to trigger login redirection and dynamic migration.

Adding a load profile

1 Select Load Profiles from the Workload Management folder and select Add Load Profile.

2 Enter the name for your profile.

3 Select Next.

4 Adjust the load profile metric weights.

When a load profile is associated with a logical cluster, the workload manager calculates a load score for each instance in the logical cluster. This is calculated using the weight you entered for each metric, the raw value of each metric for the instance, and the workload measurement algorithm. See “Viewing workload status” on page 220.

The metrics measured by the server are:

- User connections – the capacity of an instance to accept a new connection, based on resource availability.
- CPU utilization – the capacity of an instance to accept additional work.
- Run-queue length – the number of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time.
- I/O load – outstanding asynchronous I/Os.
- Engine deficit – the difference in the number of online engines among instances in the cluster.

**Note** Engine deficit is measurable only when instances in the cluster have unequal numbers of engines. Engine deficit adds a metric for maximum relative capacity to the load score.

- User metric – an optional customer-supplied metric specific to the user's environment.

  Make sure the load scores you specify add up to 100. If they do not sum to 100, the workload manager uses the scores to create proportionate values that add to 100.

5 Select Next.

6 Enter values for:

- Minimum Load Score – the load score is not a percentage, but is the minimum score the workload manager requires before it redirects work to other instances. The minimum load score is meaningful when compared to the load scores of other instances in the logical cluster using the load profile.
- Login Redirection (%) – the load threshold for determining how best to distributed incoming connections.
- Dynamic Connection Migration (%) – the load threshold that determines whether to distribute existing connections.

The load threshold is the percentage difference between the load on the current instance and the load on the least-loaded instance currently participating in a logical cluster. That value must be met before the Cluster Edition redirects a login or migrates an existing connection.

**Note** The percentages for Login Redirection and Dynamic Connection Migration are independent percentages, and do not need to add up 100.

7 Select Finish to create the load profile.

## Deleting load profiles

To delete a load profile:

1 From the Workload Management | Load Profile folder, right-click the load profile name.

2 Select Delete.

**Note** You can delete only user-created load profiles.

## Associating a load profile with a logical cluster

1 From the Workload Management | Logical Clusters folder, right-click the logical cluster name.

2 Select Properties.

3 Select the Load Profile tab.

4 Click Change. The ASE plug-in displays a list of available load profiles.

5 (Optional) Select Preview Profiles to display a window that allows you to choose a profile and see how it will influence the weighted metric values for instances within the logical cluster.

   a Highlight the load profile you want to associate with this logical cluster.

   b Select Close.

   c Select OK to exit from the Properties dialog box.

6 Select a load profile.

7 Select OK.

8 Select OK or Apply to associate the new load profile with the logical cluster.

## General tab for load profiles

To view the load profile properties:

1 Select Load Profile from the Workload Management folder.

2 Right-click the load profile name and select Properties.

The General tab describes the load profile including its name and type (whether it is a system or user load profile).

See “Load profiles” on page 209.

### Metric Weights tab

The Metric Weights tab describes the current weights applied to measurement metrics. If the weights do not sum to 100, the workload manager uses proportionate, adjusted values that do sum to 100.

- User connections – the capacity of an instance to accept a new connection, based on resource availability.
- CPU utilization – the capacity of an instance to accept additional work.
- Run-queue length – the number of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time.
- I/O load – outstanding asynchronous I/Os.
- Engine deficit – the difference in the number of online engines among instances in the cluster.

  **Note** Engine deficit is measurable only when instances in the cluster have unequal numbers of engines. Engine deficit adds a metric for maximum relative capacity to the load score.

- User metric – an optional customer-supplied metric specific to the user's environment.

  Make sure the load scores you specify add up to 100.

### Thresholds tab

Use the Thresholds tab to view and change the threshold settings for the load profile. Enter values for:

- Minimum Load Score – the load score is not a percentage, but is the minimum score the workload manager requires before it redirects work to other instances. The minimum load score is meaningful when compared to the load scores of other instances in the logical cluster using the load profile.
- Login Redirection (%) – the load threshold for determining how best to distributed incoming connections.
- Dynamic Connection Migration (%) – the load threshold that determines whether to distribute existing connections.

## Managing logical clusters

The Logical Clusters folder contains a list of defined logical clusters and a wizard to help you add new logical clusters. The right pane shows the following details:

- Cluster ID – displays the associated ID number for each logical cluster.
- Current state of the logical cluster – the connection status of the logical cluster, on- or offline.
- Connections – the number of active connections in the logical cluster.
- Base instances – the number of base instances in the logical cluster.
- Active base instances – displays the number of base instances currently active in the logical cluster.
- Failover instances – the number of instances that are configured for failover.
- Active failover instances – displays the number of active failover instances.
- Down routing mode – displays the down-routing configuration. One of:
  - system – sends unroutable connections to the system logical cluster. system ensures the highest availability as the system logical cluster is always online for every instance. This is the default setting.
  - open – sends unroutable connections to the open logical cluster. If the connection cannot be sent to the open logical cluster, or the connection does not support redirection, the instance applies the down-routing mode of the open logical cluster.
  - disconnect – disconnects unroutable connections. This setting can enforce resource reservation by disconnecting clients that cannot be served by instances in the target logical cluster.
- Failover mode – displays the failover mode configuration. The options are instance or group.
- Startup mode – displays the start-up mode configuration: automatic (the option is unselected) or manual.
- System view – displays the system view configuration, instance or cluster.

- Roles – only the system-created logical cluster has the system role. By default, the open role is assigned to the system logical cluster. When the open role is assigned to a new logical cluster, the open role is removed from the logical cluster that previously owned that role.
- Load profile – displays the name of the load profile used for the cluster.

### Adding a logical cluster

1. Right-click Add Logical Cluster.
2. Select Open.
3. Enter the cluster name and click Next.
4. Click Add to select the server instances you want to participate in the logical cluster. Click Next.
5. Add the failover server instances for the logical cluster. Click Next.
6. Enter the routed applications, logins, and aliases. You can also drop a route by first selecting the route.
7. To select a load profile other than the default, click Change, then click Next.
8. Set options for new logical cluster. Click Next.
9. ASE plug-in displays the summary. To make changes, use the Back button. When you are done click Finish.

### Dropping a logical cluster

To drop a logical cluster, it must be offline.

1. Right-click the logical cluster and select Delete.
2. Click Yes.

## Logical cluster properties

Right-click the logical cluster name and select Properties to view the logical cluster's configuration.

See "Load profiles" on page 209 for information about load profiles.

## General tab

These options appear in the General tab:

- System view – determines how the logical cluster users view Cluster Edition, whether as an entire cluster, or as individual instances. This affects some queries and stored procedures. Select Instance or Cluster.
- Automatically starts logical cluster – select this option if you want this logical cluster to start when the cluster starts.
- Failover mode – determines whether the logical cluster fails over to another instance or a group:
  - Instance – the logical cluster to fail over one instance at a time.
  - Group – specifies that base instances are replaced only when all base instances fail, and that all failover instances then come online. For example, the failover mode for SalesLC is “group.” If base instance “ase1” fails, the cluster continues to run on base instance “ase2”. No failover instances are brought online. However, if both “ase1” and “ase2” fail, then the cluster runs on failover instances “ase3” and “ase4”.
  - Fail-to-any – permits the logical cluster to fail over even though no failover instances are available. The logical cluster fails over to any available physical instances in the cluster, based on load.

    Fail-to-any configures the logical cluster to fail over to any available instance, even if it is not defined as a failover instance within that logical cluster.
- Down routing mode – displays the down-routing configuration. One of:
  - system – sends unroutable connections to the system logical cluster. system ensures the highest availability as the system logical cluster is always online for every instance. This is the default setting.
  - open – sends unroutable connections to the open logical cluster. If the connection cannot be sent to the open logical cluster, or the connection does not support redirection, the instance applies the down-routing mode of the open logical cluster.
  - disconnect – disconnects unroutable connections. This setting can enforce resource reservation by disconnecting clients that cannot be served by instances in the target logical cluster.
- Logical cluster roles – the system role is automatically set to the “system logical cluster.” You cannot change this setting.

By default, the open role is assigned to the system logical cluster. You cannot grant the open role, and it is removed only if another logical cluster assumes that role.

## Base Instances tab

The Base Instances tab lists the currently configured logical cluster instances.

- Select Add to add instances to the logical cluster.
- To remove an instance from the logical cluster:
  a Highlight the instance to remove. This instance must be offline before you can select it.
  b Select Remove.
  c Select Yes to confirm the deletion.
- Select Offline to bring an instance offline. You can specify whether to bring the instance offline immediately, or to bring it offline gradually, but to remind you after the specified period of time.
- Select Online to bring an offline instance online.
- Select Failover to fail over from this instance to another.

### Adding an instance to a logical cluster

To add an instance to the logical cluster:

1 Click Add.

2 Highlight the instance you want to add

3 Click OK. These changes do not take effect until you click Apply or OK.

## Failover Instances tab

Lists information about the currently configured failover instances, including:

- Name
- ID
- State
- Failover Group

#### Adding a failover instance

To add a failover instance:

1 Click Add.

2 Select the failover group you want to configure from the Add Failover Instance to Logical Cluster window.

Failover groups allow you to specify the order in which failover instances become active in the event of a failover. A group can have one or more instances.

3 Highlight the instance you want to configure as a failover instance from this list and click OK

## Load Profile tab

Lists information about the load profiles associated with the logical cluster:

- Name – name of the load profile associated with this logical cluster. Click Change to associate another load profile with this logical cluster.
- Type – displays the load profile type: system or user.
- Minimum Load Score – the minimum load score that activates a login redirection or migration.
- Metrics – includes a variety of statistics about the load profile, including
  - User connections – the capacity of an instance to accept a new connection, based on resource availability.
  - CPU utilization – the capacity of an instance to accept additional work.
  - Run-queue length – the number of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time.
  - I/O load – outstanding asynchronous I/Os.
  - Engine deficit – the difference in the number of online engines among instances in the cluster.
  - User metric – an optional customer-supplied metric specific to the user's environment.

- Weight – indicates how important a metric is to the load score. This is a relative measurement, and is based on a upper limit of 100. A metric with a weight of 0 has no value and is not used in calculating the load score, and a metric with a weight of 50 has one half the influence on the load score. If 5 metrics all have a weight of 20, each metric is equally important when calculating the load score.
- Thresholds – currently configured thresholds associated with this load profile.

## Routes tab

This tab describes the routes for applications, logins, and aliases that are assigned to this logical cluster.

### Adding an application route

1 Click Add Application Route.

2 Enter the application name and click OK.

### Adding a login route

1 Click Add Login Route.

2 Select the login (or logins) for the route and click OK.

### Adding an Alias route

1 Click Add Alias Route.

2 Enter the alias name.

3 Click OK.

### Dropping routes

1 Select the route.

2 Click Drop Route.

3 Select Yes to confirm the deletion.

## Viewing workload status

The Workloads folder displays a cluster-wide view of workload metrics on two tabs: Weighted Scores and Base Metric Values.

Select Workloads from the Workload Management folder in the tree view. The workload scores report the weighted and base scores.

Weighted Scores tab

Each instance tracks a set of load metrics. Load scores and metrics are computed for each combination of instance and logical cluster, and are determined by applying the logical cluster's load profile to an instance's workload statistics. The result is an overall load score and a set of weighted scores that represent the relative impact of specific instance attributes.

The Weighted Score tab displays the load score and weighted metric values for each instance and logical cluster combination. If an instance is associated with two logical clusters, there are two entries for that instance in the details tab.

The Weighted scores tab includes:

- Instance – name of the instance whose workload is represented.
- Logical Cluster – name of the logical cluster associated with the instance.
- Load Profile – the load profile assigned to the logical cluster.
- Load score – a computed value representing the overall load on the instance. Compare this unitless number across instances as a means of comparing workloads.
- User connections – the capacity of an instance to accept a new connection, based on resource availability.
- CPU Busy – - a measurement of how busy the engines are, and provides the same information as sp_sysmon. Determines an instance's capacity to accept additional work.
- Run-queue length – the number of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time.
- IO load – measures outstanding asynchronous IOs, which indicates the relative IO saturation between instances.

- Engine Deficit – measures the difference in online engines between instances. In a cluster where all the instances have the same number of engines there is no engine deficit. However, in a two-instance cluster where, for example, “instance1” has four engines and “instance2” has two engines, “instance1” has no engine deficit but “instance2” has a 50% deficit because it has half as many engines as “instance1.”
- User – the weighted value of a metric you specified in the load profile.

**Note** Because each instance can be included in multiple logical clusters, each instance has one set of metric values for each logical cluster it belongs to.

Base Metric Values tab

The Base Metric Values tab displays all workload values for each instance in the cluster. Since each instance has only one set of values, regardless of how many logical clusters it participates in, one set of values is displayed for each instance.

- Instance – name of the instance.
- % User Connections – percentage of configured user connections in use.
- % CPU busy – percentage of time the instance was busy performing work. This is a one minute, moving average taken from all engines on the system.
- % Run queue length – base percentage of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time
- % IO load – base percentage of outstanding asynchronous IOs
- % Engine deficit – the base percent difference in the number of online engines among instances in the cluster.
- % User – percentage value you provide for the user metric specified in the load profile.

## Viewing workload status of an instance

The workload status is based on the load profile associated with the instance. The status displays raw values that indicate how taxed an instance is with regard to each metric area.

To view the workload status:

1 Click the Server Instances folder to display a list of instances in the right pane.

2 Right-click an instance name and select Workload Status. The workload status displays raw values indicating the amount of work each instance is performing for each metric area.

- User connections – the capacity of an instance to accept a new connection, based on resource availability.
- CPU busy - a measurement of how busy the engines are, and provides the same information as sp_sysmon. Determines an instance's capacity to accept additional work. This is measured as a one-minute moving average.
- Run-queue length – the number of runnable tasks on a system. Run-queue length measures the processing backlog, and is a good indicator of relative response time. This is measured as a one-minute moving average.
- IO load – measures outstanding asynchronous I/Os, which indicates the relative IO saturation between instances. This is measured as a one-minute moving average.
- Engine Deficit – measures the difference in online engines between instances. In a cluster where all the instances have the same number of engines, there is no engine deficit. However, in a two-instance cluster where, for example, "instance1" has four engines and "instance2" has two engines, "instance1" has no engine deficit but "instance2" has a 50% deficit because it has half as many engines as "instance1."
- User – the weighted value of a metric you specified with the workload_metric function.

**Note** Because each instance can be included in multiple logical clusters, each instance will have one set of metric values for each logical cluster it is member to.

# Managing routes

Routes allow you to direct client connections to specific logical clusters.

## Route properties

To view the current routes:

1 From the Workload Management | Logical Clusters folder, right-click the logical cluster name.

2 Select Properties.

3 Select the Routes tab.

The Routes General tab reports the:

- Name of the route.
- Route Type – displays the type of route: application, login, or alias.
- Logical Cluster – name of the logical cluster to which this route is associated.

  To associate this route with another logical cluster, select another logical cluster name and click OK.

### Creating a route

Use the Add Route wizard to create a route.

To start the wizard:

- Select Routes from the Workload Management folder.
- Double click on Add Route.
- Follow the instructions on the wizard to add a route.

CHAPTER 12

# Administering Clusters Using sybcluster

sybcluster is an interactive utility that allows you to perform administrative tasks for the cluster from the command line.



You can use sybcluster to create a cluster, add an instance, start and stop a cluster or instance, display status information, and so on. Alternatively, you can perform these tasks using the Adaptive Server plug-in for Sybase Central as described in Chapter 11, “Administering Clusters with the Adaptive Server Plug-in.”

You cannot manage temporary databases or logical clusters using sybcluster.

- To manage temporary databases, use the Adaptive Server plug-in or follow instructions in Chapter 7, “Using Temporary Databases.”
- To manage logical clusters, use the Adaptive Server plug-in or the sp_cluster logical and sp_cluster profile stored procedures described in Chapter 5, “Managing the Workload.”

For complete syntax and usage information for the sybcluster commands, see Chapter 14, “The sybcluster Utility.”

# Using sybcluster

Some sybcluster commands are available before you connect to a cluster; others are available only after you connect to a cluster, see Table 12-1.

The sybcluster prompt lets you know whether or not sybcluster is connected to a cluster, and the default cluster and instance if either or both of these have been set.

- When sybcluster is not connected to a cluster, the prompt is:

  ```
  >
  ```

- After you have connected to a cluster, the prompt becomes:

  ```
  cluster_name>
  ```

  For example:

  ```
  mycluster>
  ```

- If you declare a default instance, the prompt is:

  ```
  cluster_name instance_name>
  ```

  For example:

  ```
  mycluster ase1>
  ```

***Table 12-1: sybcluster interactive commands***

| Command name | Description | Active before or after connecting to a cluster |
|---|---|---|
| add backupserver | Configures one or more new Backup Servers on nodes in the cluster not currently configured for Backup Server. | After |
| add instance | Adds one new instance to the cluster. | After |
| connect | Connects to an existing cluster. | Before |
| create backupserver | Create Backup Server. | After |
| create cluster | Creates a new cluster. | Before |
| create monitorserver | Creates a Monitor Server. | After |
| create xpserver | Creates an XP Server. | After |
| deploy plugin | Deploys the configuration information for a single instance of the cluster to the Unified Agent. | Before |
| diagnose { cluster \| instance } | Performs a set of checks to ensure the cluster or instance is working properly. | After |
| disconnect | Closes all connections to the current cluster and returns the cluster to the unconnected state. | After |
| drop backupserver | Drops the Backup Server. | After |

| Command name | Description | Active before or after connecting to a cluster |
|---|---|---|
| drop cluster | Removes each instance from the cluster and deletes the cluster definition from the cluster configuration file. | After |
| drop monitorserver | Drops Monitor Server. | After |
| drop xpserver | Drops the XP Server. | After |
| drop instance | Removes an instance from the cluster. | After |
| exit | Exits sybcluster. | Before or after |
| help | Lists the currently available sybcluster interactive commands. | Before or after |
| localize | Displays current values for default language, charset, and sort order. Allows you to change the default values and add or remove languages. | After |
| quit | Exits sybcluster. | Before or after |
| set backupserver | Changes the listening port number for Backup Server on one or more nodes. | After |
| set cluster | Sets properties for the cluster. | After |
| set instance | Sets properties for the instance. | After |
| set monitorserver | Changes the listening port number for Monitor Server on one or more nodes | After |
| set xpserver | Changes the listening port number for XP Server on one or more nodes | After |
| show agents | Displays information about available UAF agents. | Before |
| show backupserver config | Displays the names of the nodes on which Backup Server is configured and the associated listening port number. | After |
| show cluster | Displays configuration, log, and status values for the cluster. | After |
| show instance | Displays information about an instance. | After |
| show monitorserver confi | Displays the names of the instances and nodes on which Monitor Server is configured and the associated listening port number. | After |
| show session | Displays current agent and discovery information | After |
| show xpserver config | Displays the names of the instances and nodes on which XP Server is configured and the associated listening port number. | After |
| shutdown cluster | Shuts down the cluster by executing a Transact-SQL shutdown command for each instance in the cluster. | After |
| shutdown instance | Shuts down the instance by executing a Transact-SQL shutdown command. | After |
| start cluster | Starts all instances in the cluster. | After |
| start instance | Starts an instance in the cluster. | After |

| Command name | Description | Active before or after connecting to a cluster |
|---|---|---|
| upgrade server | Updates Adaptive Server to Adaptive Server Cluster Edition. | Before |
| use | Sets the default instance. | After |

# sybcluster and the Unified Agent Framework

sybcluster uses the Unified Agent Framework (UAF) to provide remote management capabilities, in the form of runtime services for managing distributed Sybase resources. It offers a set of common services and enables Sybase processes, such as sybcluster and the Adaptive Server plug-in, to plug in to an agent to manage server resources and perform various operations.

Unified Agent servers can broadcast themselves on a subnet using UDP, or register themselves with a lookup server such as Jini or LDAP.

See the Users *Guide to the Unified Agent and Management Console* for detailed information about UAF.

# Starting sybcluster

Start sybcluster from the command line. The simplest way is to enter:

```
sybcluster -U uafadmin -P
```

"uafadmin" is the default user name, and the default password is null or blank. You can also use authenticated Adaptive Server database user names and operating system user names. See "Authenticating the user" on page 229.

You can also start sybcluster and, at the same time, specify a cluster, identify a default instance, and connect to the Unified Agents on one or more nodes in the cluster. For example:

```
sybcluster -U uafadmin -P -C mycluster -I ase1
-F "blade1:9999,blade2:9999,blade3:9999"
```

In this example, the -F option identifies the node and listening port for each Unified Agent in the cluster. If you use this command often, you can create a simple alias for it. For example:

```
sybcluster_mycluster
```

# Connecting to the cluster

You can connect to a cluster by specifying the required information when you start sybcluster; you can also start sybcluster and connect to the cluster at a later time using the connect command. You must connect to a cluster before starting it.

When you install and configure the cluster, you also install and configure a Unified Agent on each node in the cluster. When you start sybcluster, you must provide an authenticated user name and password, and identify the Unified Agents on one or more of the cluster's nodes using a direct connect, or discovery method. sybcluster can then plug in to the Unified Agents and perform command, control, and discovery operations.

## Authenticating the user

You can connect to the cluster using any UAF authenticated user name and password. By default, the user name is "uafadmin", and the default password is a null or blank string. For example:

```
sybcluster -U uafadmin -P -C mycluster
```

## Setting the user name and password

Sybase recommends that you change the default user name and change and then encrypt a new password. User name and password information for the UAF is stored in the *csi.properties* files located on each node in the cluster.

❖ **Setting the user name**

- Enter the new user name in the "username" property in the Simple Log Module section of each *$SYBASE/UAF-2_5/nodes/<node_name>/conf/csi.properties* file. For example:

```
# Simple Login Module
...
CSI.loginModule.2.options.username=newusername
CSI.loginModule.2.options.password=
CSI.loginModule.2.options.encrypted=false
CSI.loginModule.2.options.roles=uaAgentAdmin,uaPluginAdmin
```

**Note** Make sure you edit the *csi.properties* file in each node in the cluster.

❖ **Encrypting and setting the password**

1 Run passencrypt, located in *$SYBASE/UAF-2_5UAF-2_5/bin* , to generate the encrypted password.

2 In the Simple Log Module section of each *$SYBASE/UAF-2_5/nodes/<node_name>/conf/csi.properties* file:

   1 Paste the encrypted value into the password property.

   2 Set the encrypted property to “true”. For example:

```
# Simple Login Module
...
CSI.loginModule.2.options.username=newusername
CSI.loginModule.2.options.password=REVTe1NZVUFGfWNvbS5zdW4uY3J5cHRv
LnByb3ZpZGVyLlN1bkpDRXtTWVVBRn1nTUJacVh5R3pnN09RSDJDN1NPUXhBPT0=
CSI.loginModule.2.options.encrypted=true
CSI.loginModule.2.options.roles=uaAgentAdmin,uaPluginAdmin
```

**Note** Make sure you edit the *csi.properties* file in each node in the cluster.

❖ **Activating the new user name and password**

- For the new user name and password to take effect, shut down and restart the Unified Agent on each node in the cluster.

## Identifying the Unified Agents

You can identify the Unified Agents using a direct connect or discovery method either when you start sybcluster or later using the connect to interactive command.

---

**Note** If the cluster is running when you connect to it, you normally need to identify only one Unified Agent on one node in the cluster. To start the cluster or to create Monitor Server or XP Server, however, you must identify the agent on each node in the cluster.

sybcluster always prompts you when additional information is required.

---

If you do not know the Unified Agent specifications for a cluster, use the sybcluster show agents command to discover available Unified Agents and clusters on your subnet.

### Using a direct connect method

You can connect directly to one or more agents by specifying the cluster node and port numbers for the Unified Agents responsible for managing the cluster. Some possible agent specifications are:

- Node name or names of the cluster and optional port numbers – allows you to specify an exact address. If you do not include a port number, sybcluster assumes a default value of 9999. For example, to specify agents on "blade1", "blade2", and "blade3" of "mycluster," enter:

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

  To specify node and port numbers using connect after you have started sybcluster, enter:

  ```
  connect to mycluster -U uafadmin -P
  -F "blade1:1234,blade2:2345,blade3:3456"
  ```

- The domain of the node – lets you specify the exact address through the domain name. For example:

  ```
  sybcluster -U uafadmin -P -C mycluster
  -F "blade1.mydomain.com"
  ```

### Using a discovery method

sybcluster supports these three discovery methods that locate the agents and the discovery order:

- User Datagram Protocol (UDP) – is a broadcast discovery method in which sybcluster broadcasts a request and agents located on the same subnet respond. For example, to use UDP to look up the location of a node in “mycluster”, enter:

  ```
  sybcluster -U uafadmin -P -C mycluster
  -d "udp()"
  ```

  To perform discovery using connect after starting sybcluster, enter:

  ```
  connect to mycluster login uafadmin password " "
  discovery "udp(),
  jini(myjinihost1:5678;myjinihost2:1234)"
  ```

- Jini server technology – provides lookup capabilities. Each agent registers with the Jini server, which stores the location of each node and status information. To look up the location of an agent in “mycluster”, enter:

  ```
  sybcluster -U uafadmin -P -C mycluster
  -d "jini(myjiniserver:4564)"
  ```

  To perform discovery using connect after starting sybcluster, enter:

```
connect to mycluster discovery
"jini(myjiniserver:4564)"
```

- LDAP technology – provides lookup capabilities. Each agent registers with the LDAP server, which stores the location of each node and status information.

To look up the location of Unified Agents for "mycluster" using all three discovery methods, enter:

```
sybcluster -U uafadmin -P -C mycluster
-d "udp(),jini(myjiniserver:4123),
ldap(myldapserver:6123)
```

# Starting the cluster

You must connect to the Unified Agent on each node in the target cluster before you can start the cluster. See "Connecting to the cluster" on page 229. Then, to start the cluster, enter:

```
start cluster
```

sybcluster prints a cluster description as it starts all instances in the cluster.

# Managing the cluster

This section describes how to perform tasks that help you to manage the cluster and its environment.

## Creating a cluster

Create a cluster using either sybcluster or the Adaptive Server plug-in. Either method prompts you for required information and creates the cluster for you. You can also create a cluster manually, performing all the tasks yourself.

See the *Installation Guide* for step-by-step instructions for creating a cluster using each of these methods.

## Verifying the cluster

diagnose cluster performs a set of checks that ensures that the cluster is working properly. Enter:

```
diagnose cluster
```

diagnose cluster displays cluster information and checks to see if:

- A Unified Agent is running on all nodes in the cluster.
- The number of nodes in the cluster does not exceed the maximum instances in the cluster.
- The quorum device exists, and if not, checks to see if the directory has write permission.
- The interfaces file exists for all nodes, and that node names and port numbers do not conflict.
- The primary and secondary protocol specifications do not overlap.
- The Sybase home directories on each node are shared.

## Displaying information about available Unified Agents

Use show agents to identify all the configured Unified Agents on your subnet, or narrow the search to show information about specific Unified Agents.

For example, to identify all Unified Agents, enter:

```
show agents
```

sybcluster displays the direct connect address for each Unified Agent, its node and cluster name, and other relevant information.

You can display information about specific Unified Agents by restricting discovery or identifying the desired agents. For example, to view information about Unified Agents on “blade2” of “mycluster,” enter:

```
show agents agent "blade2:9999"
```

## Displaying cluster information

This section describes how to use sybcluster to display information about the cluster, instances in the cluster, and the cluster environment. For complete syntax and usage information, see Chapter 14, “The sybcluster Utility.”

You can perform the same tasks using the Adaptive Server Plug-in. See Chapter 11, “Administering Clusters with the Adaptive Server Plug-in.”

- To show configuration information, primary and secondary protocol values, trace flags used, and the addresses for the quorum device and the master device, enter:

  ```
  show cluster config
  ```

  To view formatted configuration information for the cluster, enter:

  ```
  show cluster config template
  ```

- To show status and heartbeat information for each instance in the cluster, enter:

  ```
  show cluster status
  ```

  Status values are:

  - Up
  - Down
  - Undefined
  - Invalid
  - Start
  - Init
  - Quiesce

- To show all log information, enter:

  ```
  show cluster log
  ```

  You can limit the output by specifying:

  - The error severity level, for example:

    ```
    show cluster log minseverity 5
    ```

  - A date range for log entries:

    ```
    show cluster log startdate 03:31:08
    enddate 04:30:08
    ```

  - A number of lines to display from the error log, working backwards from the most recent:

    ```
    show cluster log last 25
    ```

- To display all UAF and JDBC connections to the cluster, enter:

```
show cluster connection
```

- To display general information about the cluster and detailed information about agent connections, enter:

```
show session
```

## Changing cluster configuration values

You can change certain configuration values for the cluster using set cluster. Verify the cluster status by issuing show cluster status.

When the cluster is down, you can change:

- The maximum number of instances
- The active trace flags
- The primary or secondary protocol

When the cluster is running, you can change:

- The login name or password the Unified Agent uses to log in to the cluster
- The default language, character set, and sort order for the cluster

For example, to change the maximum number of instances to 4 for "mycluster", enter:

```
set cluster maxinst 4
```

To reset the primary protocol for the cluster to "udp", enter:

```
set cluster primary protocol udp
```

### Changing user names or passwords

sybcluster is a client program that connects to a Unified Agent that is configured to run the cluster. When you start sybcluster, you provide a login and password that enables sybcluster to log in to the Unified Agent. To change the value of the Unified Agent login or password, use the Agent Management Console Sybase Central plug-in. To encrypt the password, see "Setting the user name and password" on page 229.

For some operations, the Unified Agent must log in to the cluster. This occurs when sybcluster or Sybase Central issues a shutdown command or when the Unified Agent performs cluster heartbeat tasks to determine cluster status. For these tasks, the Unified Agent must use a login with sa_role. By default, the Unified Agent uses the "sa" login with no password. To change this password, use the sybcluster set cluster login command.

For example, to change the password for the "sa" login to "newpassword", enter:

```
set cluster login sa password newpassword
```

The cluster must be running to perform this command.

See "set cluster" on page 341 in Chapter 14, "The sybcluster Utility," for complete syntax and usage information.

## Changing localization values

Use the sybcluster localize command to view the current values for language, character set, and sort order. After displaying the current default values, sybcluster localize prompts you to accept each of these values or change them. For example, to view current values without changing them, enter:

```
localize

Current default locale properties are:
Default Language - portuguese
Default Charset - mac
Default SortOrder - Binary ordering, for use with the
Macintosh charcter set(mac).

Options for default Language are:
1. spanish
2. portuguese
3. german
4. us_english
5. thai
6. french
7. japanese
8. chinese
9. korean
10. polish
Enter the number representing the language to be set as
defaults: [2]

Options for default charsets are:
```

```
1. gb18030
2. eucgb
3. uttf8
Enter the number representing the charset to be set as
default: [1]

Options for sort orders are:
1. Binary ordering, for the EUC GB2312-80 character set
(eucgb).
Enter the number representing the sort order to be set
as default [1]

Do you want to install any language? [Y] n
Do you want to remove any language? [N ]
The cluster mycluster was successfully localized with
default language portuguese, charset gb18030, sortorder
bin_eucgb.
```

To ensure consistency throughout the cluster, shut down and restart the cluster after changing any of the localization values.

## Disconnecting from the cluster

To close all connections to the current cluster, enter:

```
disconnect
```

## Shutting the cluster down

You can shut the cluster down gracefully, which allows transactions to complete before Adaptive Server shuts down each instance in the order specified in the cluster configuration file.

```
shutdown cluster
```

To shut down the cluster immediately, without waiting for transactions to complete, enter:

```
shutdown cluster nowait
```

## Dropping a cluster

Before you can drop a cluster, make sure that the cluster is in the Down state and the Unified Agents are running. Then, enter:

```
drop cluster
```

Adaptive Server removes cluster and instance entries from the interfaces file, deletes the cluster configuration file, marks the quorum disk as unused, deletes the log file, and shuts down and removes the cluster's Unified Agent plug-ins. You must confirm each deletion.

# Managing an instance

This section describes how to perform tasks that help you manage instances in the cluster.

## Displaying information about the instance

Similarly to show cluster, show instance displays configuration, status, and log information about an instance.

- To show configuration information, including the name of the host node, primary and secondary network information, and the path to the log file, enter:

  ```
  show instance instance_name config
  ```

- To display status information, enter:

  ```
  show instance instance_name status
  ```

  This command displays status information for the named instance:

  - Up
  - Down
  - Undefined
  - Invalid
  - Start
  - Init

    - Quiesce
- To show all log information, enter:

  ```
  show instance instance_name log
  ```

  You can limit the output by specifying:

  - The error severity level, for example:

    ```
    show instance instance_name log minseverity 5
    ```

  - A date range for log entries:

    ```
    show instance instance_name log startdate
    03:31:08 enddate 04:30:08
    ```

  - A number of lines to display from the error log, working backwards from the most recent:

    ```
    show instance instance_name log last 25
    ```

## Adding an instance

You can add an instance to the cluster either interactively, with sybcluster prompting you for required values, or by using an input file.

---

**Note** add instance creates a local system temporary database for the new instance. Make sure that there is sufficient space on the device.

---

If you add the instance interactively, Adaptive Server prompts for:

- The instance name, if one was not specified in the command statement
- The name of the node hosting the instance
- The port number of the Unified Agent on the node
- The query port number
- The primary and secondary address of the node
- The primary and secondary port specifications

If you add the instance using an input file, make sure the file mirrors the format of the cluster input file (see the *Installation Guide*), although you need to include definitions only for the new instance.

For example, to add an instance using an input file, enter:

```
add instance new_instance file /$SYBASE/myfile
```

To add an instance interactively, enter:

```
add instance new_instance
```

## Verifying the instance

diagnose instance performs a set of checks to insure that the instance is configured properly. For example, to verify the configuration for “ase1”, enter:

```
diagnose instance ase1
```

diagnose instance displays configuration information for the instance and verifies:

- The query port
- That a JDBC connection is available
- That the instance is available on the public network
- The minimum and maximum port numbers
- The primary and secondary protocol port ranges

## Changing the default instance

Use the use command to set or change the default instance specified in the sybcluster command line. After the default instance has been set, you do not need to specify one in the command line for any interactive command. For example, enter:

```
use ase1
```

You can override the default instance by including an instance name in an interactive command. However, doing so does not change the default designation.

To remove the default designation, omit the instance name. Enter:

```
use
```

## Changing instance properties

You can use set instance to change certain properties for the instance. The instance must be in the Down state to use set instance. Verify status by issuing show cluster status.

Instance properties you can change are:

- The log path
- Arguments used for starting the instance
- The primary or secondary address of the instance
- The primary or secondary port range used by the instance

For example, to reset the primary port range from 6123 to 6126, enter:

```
set instance primary port 6123 6126
```

## Shutting an instance down

You can shut an instance down gracefully, which allows transactions to complete. For example, to shut down “ase1”, enter:

```
shutdown instance ase1
```

To shut down the instance immediately, without waiting for transactions to complete, enter:

```
shutdown instance ase1 nowait
```

If you shut down the last instance in a cluster, the cluster status changes to down. If you shut down the instance on the node hosting the cluster coordinator, another node hosts the coordinator.

## Dropping an instance

Before you can drop an instance, make sure the instance is in the down state and the cluster is in the up state. Then, enter:

```
drop instance instance_name
```

Adaptive Server deletes entries for the instance from the interfaces file and the quorum device, and notifies the cluster of the topology change. Confirm each deletion.

**Note** You cannot use drop instance to drop the last instance in a cluster. Rather, use drop cluster.

# Enabling sybcluster after manually creating the cluster

Typically, a cluster is created using sybcluster or the Adaptive Server plug-in. In these cases, Adaptive Server automatically adds configuration information that allows sybcluster or the Adaptive Server Plug-in to connect to the Unified Agent on each node. If you configure the cluster manually (as described in the *Installation Guide*), you must add that configuration information to the Unified Agents before you can use sybcluster or the Adaptive Server Plug-in to manage the cluster.

You must first deploy the plug-in configuration information.

1 If you have not already done so, start the Unified Agents on the cluster. See Chapter 3, "Installing the Server and Starting the Cluster," in the installation guide for your platform.

2 Deploy the plug-in. For example, to deploy the cluster agent plug-in information on the default cluster "mycluster", enter:

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

You can specify the Unified Agents using any of the direct connect or discovery methods described in "Identifying the Unified Agents" on page 231.

After the agents are specified, Adaptive Server prompts you for the paths to:

- The quorum device
- The environment shell script

- The ASE home directory

**Note** You can use sybcluster or Adaptive Server Plug-in to manage the cluster after deploying the plug-in to a single node in the cluster. However, to start the cluster, you must deploy the plug-in to all nodes in the cluster.

You can also use deploy plugin to update the values for an existing plug-in.

# Creating and managing auxiliary servers

Use sybcluster to create, drop, configure port numbers, and display current port numbers for these auxiliary servers:

- Backup Server
- Monitor Server
- XP Server

## Creating auxiliary servers

See the *Installation Guide* for instructions for creating Backup Server, Monitor Server, and XP Server using sybcluster. You can also use the Adaptive Server Plug-in to create auxiliary servers.

You can create a Backup Server, Monitor Server, or XP Server on one or more nodes in the cluster.

See Chapter 14, "The sybcluster Utility," for syntax and usage information.

## Dropping auxiliary servers

Use the drop backupserver, drop monitorserver, or drop xpserver to remove an auxiliary server from the cluster. The Cluster Edition prompts for confirmation before dropping a server.

You can drop the Backup Server from one or more nodes in the cluster. However, when you use drop monitorserver or drop xpserver, you drop all Monitor Servers or all XP Servers from the cluster.

To drop all XP Servers from “mycluster”, enter:

```
drop xpserver
```

```
Are you sure you want to drop the XP Servers from cluster "mycluster"? (Y or
N): [N] y
The XP Servers have been dropped for all instances.
```

To drop the Backup Server from “blade2” of “mycluster” enter:

```
drop backupserver
```

```
Do you want to drop the Backup Server from:
       1. Selected nodes
       2. Cluster
Enter choice: 1
Do you want to drop Backup Server from node "blade1"? [N] n
Do you want to drop Backup Server from node "blade2"? [N] y
Do you want to drop Backup Server from node "blade3"? [N] n
The Backup Server has been dropped.
```

## Displaying listening port information

To display current listing port numbers for Backup Server, Monitor Server, or XP Server, use:

- show backupserver config
- show monitorserver config
- show xpserver config

For example, to display Backup Server listening port information for “mycluster”, enter:

```
show backupserver config

Backup Server is configured on the following nodes:
   1. blade1:5001
   3. blade3: 5003
```

## Changing listening port information

To change a listening port for an auxiliary server, use:

- set backupserver

- set monitorserver
- set xpserver config

For example, to change the listening port number for Monitor Server for instance “ase3” on “blade3”, enter:

```
set monitorserver

Enter the Monitor Server port number for instance "blade1": [6011] <CR>
Enter the Monitor Server port number for instance "blade2": [6012] <CR>
Enter the Monitor Server port number for instance "blade3": [6013] 6666
```

# Upgrading the server

See the installation guide for your platform for instructions for upgrading Adaptive Server to the current version of the Cluster Edition.

CHAPTER 13

# System Changes

This chapter describes stored procedures, commands, configuration parameters, system tables, global variables, functions, and remote procedure calls that are specific to the Cluster Edition.



## Commands

This section describes changes to commands.

### alter table

- A referential integrity constraint cannot reference a column on a local temporary database except from a table on the same local temporary database. alter table fails when it attempts to create a reference to a column on a local temporary database from a table in another database.
- You cannot encrypt a column with an encryption key stored in a local temporary database unless the column's table resides on the same local temporary database. alter table fails if it attempts to encrypt a column with an encryption key on the local temporary database and the table is in another database.

## create database

Description

Creates a new database.

**Note** This reference page contains information specific to the Cluster Edition. See the *Reference Manual* for complete syntax and usage information for create database.

Syntax

```
create [ [ global | system ] temporary ] database database_name
    [ for instance instance_name ]
    [on {default | database_device} [= size]
        [, database_device [= size]]...]
    [log on database_device [= size]
        [, database_device [= size]]...]
    [with {override | default_location = "pathname"}]
    [for {load | proxy_update}]
```

Parameters

- global temporary – indicates that you are creating a global temporary database.
- system temporary – indicates that you are creating a local system temporary database.
- temporary – indicates that you are creating a temporary database.
- *database_name* – is the name of the new database. This name must conform to the rules for identifiers and cannot be a variable.
- for instance *instance_name* – specifies the instance that is to own the local system temporary database or local temporary database you are creating. This parameter is not used when creating global temporary databases.

  **Note** You must create a local user temporary database from the instance that is to own it. You can create a local system temporary database from any instance.

Examples

**Example1** Creates a local user temporary database on "ase1." Execute the following command from the owner instance ("ase1").

```
create temporary database local_tempdb1 for instance
ase1
```

or,

```
create temporary database local_tempdb1
```

**Example 2** Creates a local system temporary database on "ase1." Execute this command from any instance in the cluster.

```
create system temporary database local_systempdb1 for
instance ase1
```

**Example 3** Creates a global temporary database.

```
create global temporary database global_tempdb1
```

## create schema

You cannot include a referential integrity constraint that references a column on a local temporary database unless it is from a table on the same local temporary database. create schema fails when it attempts to create a reference to a column on a local temporary database from a table in another database.

## create table

You cannot include a referential integrity constraint that references a column on a local temporary database unless it is from a table on the same local temporary database. create table fails if it attempts to create a reference to a column on a local temporary database from a table in another database.

You cannot encrypt a column with an encryption key stored in a local temporary database unless the column's table is on the same local temporary database. create table fails when it attempts to encrypt with an encryption key located on the local temporary database and the table is on another database.

## New and changed dbcc commands

This section describes new and changed dbcc commands.

### New dbcc commands

#### dbcc nodetraceon and dbcc nodetraceoff

| | |
|---|---|
| Description | Enables or disables a trace flag on a local instance. |
| Syntax | dbcc nodetraceon(*trace_flag_number*)<br>dbcc nodetraceoff(*trace_flag_number*) |
| Parameters | *trace_flag_number* – is the number of the trace flag you are enabling or disabling. |

Examples

Enables trace flag 3604:

```
dbcc nodetraceoff(3604)
DBCC execution completed. If DBCC printed error
messages, contact a user with System Administrator (SA)
role.
```

Usage

dbcc traceon and dbcc traceoff apply trace flags for the entire cluster, while dbcc nodetraceoff and dbcc nodetraceon apply trace flags locally.

**dbcc set_scope_in_cluster**

Description

Sets the scope of the dbcc command to the cluster or instance and specifies what the scope is currently set to.

Syntax

dbcc set_scope_in_cluster("cluster"|"instance"|"scope")

Parameters

- cluster – sets the dbcc command scope to the cluster. Subsequent dbcc commands have a cluster-wide effect.
- instance – sets the dbcc command scope to the current instance. Subsequent dbcc commands affect only the local instance.
- scope – displays the current scope of the dbcc command, either cluster or instance.

Examples

**Example 1** Sets the dbcc scope to the cluster:

```
dbcc set_scope_in_cluster('cluster')
```

**Example 2** Sets the dbcc scope to the instance:

```
dbcc set_scope_in_cluster('instance')
```

**Example 3** Displays the current scope for dbcc commands:

```
dbcc set_scope_in_cluster('scope')
```

**dbcc quorum**

Description

Displays the portion of the quorum device related to cluster membership, including the cluster view records that describe the state of the cluster.

Syntax

dbcc quorum

Examples

Displays the contents of the quorum disk for the server:

```
dbcc quorum
```

Usage

- Although you issue dbcc quorum from an instance, the output goes to the:
  - Terminal that started ASE by default

- Client session if trace flag 3604 is on
- Client session if error log 3605 is on.

- dbcc quorum accepts an integer parameter for the number of view records to print. For example this prints the 20 most recent view records:

```
dbcc quorum(20)
```

- If you do not include a parameter dbcc quorum prints the 10 most recent view records. Pass -1 to print all view records.
- Issue dbcc quorum (-1) to view all records.

Permissions

You must have the sa_role to run dbcc quorum.

### Changed dbcc command

#### dbcc checkstorage

dbcc checkstorage includes these restrictions for the Cluster Edition:

- You cannot include instance-only named caches with dbcc checkstorage. dbcc checkstorage issues this error message if you include an instance-only named cache:

```
The cache %1! cannot be used because it is an
instance only cache
```

- To run dbcc checkstorage against a local temporary database, you must run the command from the same node that owns the local temporary database.
- Compared to other releases, dbcc checkstorage for the Cluster Edition may report more soft faults when users in multiple nodes update data. For performance reasons, dbcc checkstorage may not query the latest version of a page in the cluster, causing it to report more soft faults.

  For well-partitioned applications where a single node updates a database, dbcc checkstorage behaves the same as earlier Adaptive Server releases.

## disk init

The Cluster Edition includes the instance parameter for disk init:

```
disk init
name = “device_name”
physname = “physical_name”,
[vdevno = virtual_device_number ,]
```

```
size = number_of_blocks
      [, vstart = virtual_address
      , cntrltype = controller_number ]
[, contiguous]
[, dsync = {true | false}]
[ ,directio = [{true | false}]
[, instance = “instance_name”]
```

The instance parameter marks the device as private and sets its owning instance to *instance_name*.

By default, the disk init . . . directio parameter is set to false for nonclustered versions of Adaptive Server.

For the Cluster Edition, the default behavior of the disk init . . . directio parameter is set to true.

The Cluster Edition provides a new option in the disk init command to create a private device:

```
[instance = "instance_name"]
```

For more information on disk init, see “Using Temporary Databases” on page 139.

**Note** The directio parameter is ignored if you use raw devices.

## disk reinit

The Cluster Edition includes the instance parameter for disk reinit:

```
disk reinit
name = “device_name”
physname = “physical_name”,
[vdevno = virtual_device_number ,]
size = number_of_blocks
      [, vstart = virtual_address
      , cntrltype = controller_number ]
[, contiguous]
[, dsync = {true | false}]
[ ,directio = [{true | false}]
[, instance = “instance_name”]
```

By default, the disk reinit . . . directio parameter is set to false for non-clustered versions of Adaptive Server.

For the Cluster Edition, the default behavior of the disk reinit . . . directio parameter is set to true.

The Cluster Edition provides a new option in the disk init command to create a private device:

```
[instance = "instance_name"]
```

For more information on disk init, see “Using Temporary Databases” on page 139.

**Note** The directio parameter is ignored if you use raw devices.

## grant

grant fails if you attempt to grant permissions to user-defined roles in a local temporary database.

## quiesce database

The Cluster Edition supports the quiesce database command.

- If you issue shutdown instance or shutdown cluster, the cluster aborts all quiescedb commands.
- The cluster rejects all quiescedb commands issued by a user if shutdown instance or shutdown cluster commands are in progress.
- The cluster aborts all quiescedb commands if instance failover recovery is in progress.
- The cluster rejects all quiescedb commands issued by a user if instance failover recovery is in progress.
- You cannot add a new instance to the cluster while the master database is part of an ongoing quiesce database hold command.

## revoke

revoke fails if you attempt to revoke permissions from user-defined roles in a local temporary database.

## set system_view

set system_view allows you to determine the system view for a session, and controls the materialization of fake tables, such as sysprocesses and syslocks, which impact the output of stored procedures such as sp_who. The syntax is:

```
set system_view { instance | cluster | clear }
```

- instance – sets the system view for the current instance.
- cluster – sets the system view for the cluster.
- clear – clears any session level setting, reverting to the system_view setting for the logical cluster hosting that spid. For example, enter select *@@system_view* to check the current value.

## shutdown

The Cluster Edition allows you to shut down the cluster or an individual instance with the shutdown command.

```
shutdown {cluster | [instance_name]} [with {wait | nowait}]
```

Where

- cluster – is a keyword to shut down all instances in this cluster.
- *instance_name* – is the name of the instance you are shutting down.

For example, to shut down the cluster named “bigcluster”, enter:

```
shutdown cluster
```

To shut down the instance “ase1” (but leave the cluster running):

```
shutdown ase1
```

This form of shutdown are invalid in a clustered environment:

```
shutdown
go
```

# Stored procedures

This section describes new and changed stored procedures.

## New stored procedures

### sp_cluster connection, migrate

Description — Migrates a connection to a different logical cluster or instance.

Syntax — sp_cluster connection, "migrate", *lc_name*, *instance_name*, "*spid_list*"

Parameters — *lc_name* – is the name of the logical cluster.

*instance_name* – is the name of the instance.

*spid_list* – is the list of spids you are migrating. Separate multiple spids with semicolons.

Examples — Moves the connection with a spid of 73 into the SalesLC cluster:

```
sp_cluster connection, migrate, SalesLC, NULL, '73'
```

Moves the current connection to the "ase3" instance:

```
sp_cluster connection, migrate, NULL, ase3
```

Moves connections with spid values of 73 and 75 into "ase3" instance and the SalesLC cluster:

```
sp_cluster connection, migrate, SalesLC, ase3, '73;75'
```

Usage — To migrate the current spid, omit *spid_list* from sp_cluster connection, migrate.

### sp_cluster connection, migrate_cancel

Description — Determines if previous connection migrations to a new instance are pending, and terminates the migrations if they are.

Syntax — sp_cluster connection, 'migrate_cancel' [, '*spid_list*']

Parameters — migrate_cancel – indicates you are investigating the status of connection migrations.

*spid_list* – is the list of spids you are investigating. Separate multiple spids with semicolons.

Examples — Determines if there is a connection migration occurring on spid 73; if there is, the Cluster Edition cancels the migration:

```
sp_cluster connection, 'migrate_cancel', '73'
```

## sp_cluster connection, migrate_status

Description — Determines if previous connection migrations to a new instance are pending.

Syntax

sp_cluster connection, 'migrate_status' [, '*spid_list*' ]

Parameters

migrate_status – indicates you are investigating the status of connection migrations.

*spid_list* – is the list of spids you are migrating. Separate multiple spids with semicolons.

Examples — Checks the status of migrated connections for connections with a spid value of 73:

```
sp_cluster connection, 'migrate_status', '73'


        SPID  LogicalCluster  Instance
------------   --------------  --------
MigrationLogicalCluster   MigrationInstance   Command
----------------------   -----------------   -----------------
          73          SystemLC      ase1
                SalesLC                 ase3   connection migrate
```

## sp_cluster logical, action

Description — Modifies an outstanding action, such as canceling the action or changing the timing of the action.

Syntax

sp_cluster logical, "action", *lc_name*, {
cancel, *action_handle* |
modify_time, *action_handle*, *wait_option*[, *timeout* ] |
release, *action_handle* }

Parameters

*lc_name* – is the name of the logical cluster.

cancel – specifies an action to be canceled.

*action_handle* – is the action identifier.

modify_time – specifies that the time of the action is to be modified.

*wait_option* – is how the time of the action is to be modified. Values are:

- wait – indicates that existing connections are given a specified amount of time (or an infinite amount of time if no *timeout* is given) to migrate or disconnect.

- nowait – indicates that existing connections are migrated or disconnected immediately.
- until – indicates that existing connections are given until a specific time of day to migrate or disconnect.

*timeout* – is a specific amount of time (when used with wait) or a specific time (when used with until). The format is “hh:mm:ss” according to a 24-hour clock. For example, *timeout* records "11:30 p.m" (or "11:30pm") as "23:30:00".

release – specifies that all resources held by a completed action are to be released.

Examples

**Example 1** Cancels a timed action on the “SalesLC” logical cluster. The action handle is 4390.

```
sp_cluster logical, "action", SalesLC, cancel, "4390"
```

**Example 2** Changes the wait option for existing action 5364 to nowait.

```
sp_cluster logical, "action", SalesLC, modify_time,
"5364", nowait
```

**Example 3** Releases action 3456 for the “SalesLC” logical cluster.

```
sp_cluster logical, "action", SalesLC, release, "3456"
```

Usage

- Retrieve an action handle by querying the monLogicalClusterAction table or executing:

  ```
  sp_cluster logical, "show", NULL, action
  ```

- Any client that does not support migration is disconnected when it completes a SQL batch and has no open transactions, or when the *timeout* period expires, which ever comes first.
- Any client remaining at the end of the *timeout* period is disconnected.
- Cancelling an action does not roll back the action. Additional tasks may be necessary to restore the configuration to the original state.
- Only completed actions can be released. Releasing an action removes the completed action from the system and from the monLogicalClusterAction table.

## sp_cluster logical, add

Description

Adds a resource or one or more routes to the logical cluster.

Syntax

```
sp_cluster logical, "add", lc_name, {
    route, route_type, key_list |
    instance, instance_list |
    failover, instance_list }
```

Parameters

*lc_name* – is the name of a logical cluster.

route – specifies that one or more routes are to be added to the logical cluster.

*route_type* – is the type of route to be added. Values are:

- application – specifies a route for an application name to the logical cluster.
- login – specifies a route for a login name to the logical cluster.
- alias – specifies a route for a server name alias to the logical cluster.

*key_list* – is a list of applications, logins, or aliases, depending on the route type. Elements in the key list are delimited by semicolons.

instance – specifies that one or more base instances are to be added to the logical cluster.

*instance_list* – is the list of instances to be added. Separate multiple instances with semicolons.

failover – specifies that one or more failover instances are to be added to the logical cluster.

Examples

**Example 1** Adds instances “ase1” and “ase2” to the “SalesLC” logical cluster.

```
sp_cluster logical, "add", SalesLC, instance,
"ase1;ase2"
```

**Example 2** Creates one failover group with “ase3” for “SalesLC”.

```
sp_cluster logical, "add", SalesLC, failover, ase3
```

**Example 3** Routes the logins “tom”, “dick”, and “harry” to the “SalesLC” logical cluster

```
sp_cluster logical, "add", SalesLC, route, login,
"tom;dick;harry"
```

**Example 4** Routes the field_sales application to the “SalesLC” logical cluster.

```
sp_cluster logical, "add", SalesLC, route, application,
field_sales
```

Usage

- You cannot add a base instance or a failover resource to the system logical cluster.
- Separate multiple instance, failover resources, or applications with semicolons.
- Create multiple failover groups by enclosing the group in parenthesis, and separating groups with a comma. If you do not specify group, a new group is created and the instances are added to that group. You can specify a group into which the instances are placed (the group number must be quoted).

  For example:

```
1> sp_cluster logical, 'add', tempLC, failover, "asedemo3;asedemo2"
2> go
Added failover instance 'asedemo3' to group 1 for logical cluster
'tempLC'.
Added failover instance 'asedemo2' to group 1 for logical cluster
'tempLC'.
```

  And then add the instances to the group:

```
1> sp_cluster logical, 'add', tempLC, failover, asedemo4, "4"
2> go
Added failover instance 'asedemo4' to group 4 for logical cluster
'tempLC'.
```

## sp_cluster logical, alter

Description

Moves a route from one logical cluster to another.

Syntax

sp_cluster logical, “alter”, *lc_name*, route, *route_type*, *key_list*

Parameters

- *lc_name* – is the name of a logical cluster.
- route – specifies a route is to be altered.
- *route_type* – is the type of route to be altered. Values are:
  - application – specifies a route for an application name to the logical cluster.
  - login – specifies a route for a login name to the logical cluster.
  - alias – specifies a route for a server name alias to the logical cluster.
- *key_list* – is a list of applications, logins, or aliases, depending on the route type. Elements in the key list are delimited by semicolons.

Examples

Creates a route of type alias to logical cluster “lc1” with the alias “SalesLC”. Then, changes the logical cluster association of the route from “lc1” to “lc2”. The route is identified by its route type (alias) and its key (SalesLC).

```
sp_cluster logical, "add", "lc1", "route", "alias",
    "SalesLC"
sp_cluster logical, "alter", "lc2", "route", "alias",
    "SalesLC"
```

## sp_cluster logical, create

Description

Creates a new logical cluster.

Syntax

sp_cluster logical, "create", *lc_name*

Parameters

*lc_name* – is the name of the logical cluster.

Examples

Creates a logical cluster named “SalesLC”.

```
sp_cluster logical, "create", SalesLC
```

## sp_cluster logical, deactivate

Description

Stops the logical cluster on one or more instances or the entire logical cluster, and places the instances or the cluster in the inactive state.

Syntax

sp_cluster logical, "deactivate", *lc_name*, {
"cluster" |
"instance", *instance_list* }
[, *wait_option*[, *timeout*,[, @handle output ]]]

Parameters

*lc_name* – is the name of a logical cluster.

cluster – specifies the entire cluster.

instance – specifies that only certain instances in the logical cluster are to be placed in the inactive state.

*instance_list* – is a list of selected instances in the logical cluster.

*wait_option* – is how the time of the action is to be specified. Values are:

- wait – indicates that existing connections are given a specified amount of time (or an infinite amount of time if no *timeout* is given) to migrate or disconnect.
- nowait – indicates that existing connections are migrated or disconnected immediately.

- until – indicates that existing connections are given until a specific time of day to migrate or disconnect.

*timeout* – is a specific amount of time (when used with wait) or a specific time (when used with until). The format is "hh:mm:ss" according to a 24-hour clock. For example, *timeout* records 11:30 p.m. as 23:30:00.

@handle output – specifies that an action handle is to be retrieved for the action.

Examples

**Example 1** Immediately stops all instances in the "SalesLC" logical cluster, and places "SalesLC" in the inactive state.

```
sp_cluster logical, "deactivate", SalesLC, cluster,
nowait
```

**Example 2** Stops the "ase1" and "ase2" instances, and places "SalesLC" in the inactive state.

```
sp_cluster logical, "deactivate", SalesLC, instance,
"ase1;ase2"
```

Usage

- You cannot use the deactivate command for the system logical cluster.
- offline is identical to the deactivate, except deactivate places stopped instances or clusters in the inactive state and offline places them in the offline state.

## sp_cluster logical, drop

Description

Drops a logical cluster, or one or more resources from the logical cluster.

Syntax

sp_cluster logical, "drop", *lc_name*, {
cluster |
instance, *instance_list* |
failover, *instance_list* |
route, *route_type*, *key_list* }

Parameters

*lc_name* – is the name of a logical cluster.

cluster – specifies that the named cluster is to be dropped.

instance – specifies that one or more base instances are to be dropped from the logical cluster.

*instance_list* – is the list of instances to be dropped. Separate multiple instances with semicolons.

failover – specifies that one or more failover instances are to be dropped from the logical cluster.

route – specifies that one or more routes are to be dropped from the logical cluster.

*route_type* – is the type of route to be dropped. Values are:

- application – specifies a route for an application name to the logical cluster.
- login – specifies a route for a login name to the logical cluster.
- alias – specifies a route for a server name alias to the logical cluster.

*key_list* – is a list of applications, logins, or aliases, depending on the route type. Elements in the key list are delimited by colons.

Examples

**Example 1** Drops the "SalesLC" logical cluster.

```
sp_cluster logical, "drop", SalesLC, cluster
```

**Example 2** Drops the base instances "ase1" and "ase2" from the "SalesLC" logical cluster.

```
sp_cluster logical, "drop", SalesLC, instance,
"ase1;ase2"
```

**Example 3** Drops the routes from the applications field_sales and web_sales from the "SalesLC" logical cluster.

```
sp_cluster logical "drop", SalesLC, route, application,
"field_sales;web_sales"
```

Usage

- You must place an instance or failover resource in the offline state before dropping it.
- Dropping a cluster also drops all routes, resources, and settings associated with the cluster.

## sp_cluster logical, failback

Description

Reverses a manual failover, reinstating the original base instances.

Syntax

sp_cluster logical, "failback", *lc_name*, {
cluster[, *wait_option*[, *timeout*[, @handle output ]]] |
instance, *from_instance_list*, *to_instance_list*[, *wait_option*[,
*timeout*[, @handle output ]]] }

Parameters

*lc_name* – is the name of a logical cluster.

cluster – specifies a failback of the entire logical cluster.

*to_instance_list* – is a list of predefined failover instances. A value of NULL activates the first failover group.

*wait_option* – is how the time of the action is to be recorded. Values are:

- wait – indicates that existing connections are given a specified amount of time (or an infinite amount of time if no *timeout* is given) to migrate or disconnect.
- nowait – indicates that existing connections are migrated or disconnected immediately.
- until – indicates that existing connections are given until a specific time of day to migrate or disconnect.

*timeout* – is a specific amount of time (when used with wait) or a specific time (when used with until). The format is "hh:mm:ss" according to a 24-hour clock. For example, *timeout* records 11:30 pm as 23:30:00.

@handle output – specifies that an action handle is to be retrieved for the failback.

instance – specifies that only selected instances in the logical cluster are to fail back.

*from_instance_list* – is a list of instances that are to be taken offline.

Examples

**Example 1** Fails back the “SalesLC” logical cluster.

```
sp_cluster logical, "failback", SalesLC, cluster
```

**Example 2** “SalesLC” is running on “ase3” and “ase1”. In this example, “ase3” fails back to “ase1”, and “SalesLC” continues to run on “ase2”. The action takes place in two minutes.

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failback", SalesLC, instance,
ase3, ase1, wait, "00:02:00", @handle = @out_handle
output
```

Usage

- To initiate a failback, the logical cluster must first be failed over.

## sp_cluster logical, failover

Description

Initiates a manual failover from base instances to failover instances.

Syntax

sp_cluster logical, “failover”, *lc_name*, {
cluster[, *to_instance_list*[, *wait_option*[, *timeout*[,
@handle output ]]] |
instance, *from_instance_list*, *to_instance_list*[, *wait_option*[,
*timeout*[, @handle output ]]] }

Parameters

*lc_name* – is the name of a logical cluster.

cluster – specifies the failover of the entire logical cluster.

*to_instance_list* – is a list of predefined failover instances. A value of NULL activates the first failover group.

*wait_option* – is how the time of the action is to be recorded. Values are:

- wait – indicates that existing connections are given a specified amount of time (or an infinite amount of time if no *timeout* is given) to migrate or disconnect.
- nowait – indicates that existing connections are migrated or disconnected immediately.
- until – indicates that existing connections are given until a specific time of day to migrate or disconnect.

*timeout* – is a specific amount of time (when used with wait) or a specific time (when used with until). The format is "hh:mm:ss" according to a 24-hour clock. For example, *timeout* records 11:30 pm as 23:30:00.

@handle output – specifies that an action handle is to be retrieved for the failover.

instance – specifies that only selected instances in the logical cluster are to fail over.

*from_instance_list* – is a list of instances that are to be taken offline.

Examples

**Example 1** Fails over the “SalesLC” logical cluster to the first group of predefined failover resources. The failover waits 2 minutes before terminating connections.

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failover", SalesLC, cluster, NULL,
wait, "00:02:00", @handle = @out_handle output
```

```
Action '2' has been issued for the 'failover cluster' command.

Logical Cluster Handle      Action                          From     To
         State            InstancesWaiting ConnectionsRemaining WaitType
         StartTime                 Deadline
         CompleteTime
 --------------- ----------- ------------------------------- -------- --
         ---------------- ---------------- -------------------- --------
         ------------------------- -------------------------
```

```
        -------------------------

 SalesLC                  2 failover cluster                 2, 4     NULL
        complete                        0                    0 wait
               Nov 15 2007  3:23PM        Nov 15 2007  3:25PM
               Nov 15 2007  3:23PM

Remember to issue the `sp_cluster logical, action, <logical cluster name>,
release, <handle>' command for any cancelled or completed actions.
```

**Example 2** "SalesLC" is running on "ase1" and "ase2". In this example, "ase1" fails over to "ase3", and "SalesLC" continues to run on "ase2". No wait option is specified, so it defaults to an indefinite wait.

```
sp_cluster logical, "failover", SalesLC, instance,
ase1, ase3
```

```
Action '1' has been issued for the 'failover instance' command.

 Logical Cluster Handle      Action                             From To

        State            InstancesWaiting ConnectionsRemaining

        WaitType         StartTime                  Deadline

        CompleteTime

 -------------- ----------- ---------------------------------- ---- --

        ---------------- ---------------- --------------------

       ---------------- ------------------------- --------------------------

        --------------------------

 SalesLC                  1 failover instance                  1    4

        complete                        0                   0

        infinite               Nov 15 2007  3:06PM                      NULL

               Nov 15 2007  3:06PM

Remember to issue the `sp_cluster logical, action, <logical cluster name>,
release, <handle>' command for any cancelled or completed actions.
```

## sp_cluster logical, [ gather | set ]

Description — Gathers and migrates a group of connections to a different logical cluster. This can be done manually using "gather" or automatically using "set".

Syntax

```
sp_cluster logical, [ "gather", lc_name |
    "set", lc_name, "gather", [ "automatic" | " manual " ] ]
```

Parameters

gather – indicates you are gathering a set of qualified connections to migrate them to another logical cluster.

*lc_name* – is the name of a logical cluster.

set – indicates that you are setting the migration for a connection to either manual or automatic.

gather – indicates you are combining all connections with similar settings for a migration to another logical cluster

Usage

- Automatic gather occurs when the cluster is brought online manually, routes are added or modified, or the cluster is specified as a new open cluster.
- The logical cluster must be online to gather connections manually.
- The logical cluster must have defined routes to gather connections.

## sp_cluster logical, help

Description

Displays complete syntax for sp_cluster logical.

Syntax

sp_cluster logical, "help"

For more information about sp_cluster logical, see Chapter 5, "Managing the Workload," or individual sp_cluster logical stored procedure descriptions in this chapter.

Examples

Displays syntax for the sp_cluster logical stored procedures.

```
sp_cluster logical, "help"

Usage for sp_cluster 'logical':
sp_cluster 'logical', 'help' [, <module>]

To show the logical cluster configuration:
sp_cluster 'logical', 'show'
sp_cluster 'logical', 'show', <lcname>
sp_cluster 'logical', 'show', <lcname> | NULL, 'action' [, <state>]
sp_cluster 'logical', 'show', <lcname> | NULL, 'route' [, <type [, <key>]]

To create a logical cluster:
sp_cluster 'logical', 'create', <lcname>

To add resources to a logical cluster:
sp_cluster 'logical', 'add', <lcname>, 'failover', <instance_list> [,<group>]
sp_cluster 'logical', 'add', <lcname>, 'instance', <instance_list
sp_cluster 'logical', 'add', <lcname>, 'route', <route_type>, <key_list>
```

```
To drop resources from a logical cluster:
sp_cluster 'logical', 'drop', <lcname>, 'cluster'
sp_cluster 'logical', 'drop', <lcname>, 'failover', <instance_list>
sp_cluster 'logical', 'drop', <lcname>, 'instance', <instance_list>
sp_cluster 'logical', 'drop', <lcname>, 'route', <route_type>, <key_list>

Argument details:
<lcname> is a logical cluster nam
 <instance_list> is a ';' separated list of instance
<route_type> is one of {'user', 'application', 'alias
 <key_list> is a ';' separated list of keys

To set attributes of a logical cluster:
sp_cluster 'logical', 'set', <lcname>, 'open'
sp_cluster 'logical', 'set', <lcname>, 'down_routing', 'disconnect' | 'system'
| 'open'
sp_cluster 'logical', 'set', <lcname>, 'failover', 'instance' | 'group'
sp_cluster 'logical', 'set', <lcname>, 'load_profile', <profile_name>
sp_cluster 'logical', 'set', <lcname>, 'startup', 'automatic' | 'manual'
sp_cluster 'logical', 'set', <lcname>, 'system_view', 'instance' | 'cluster'

To start and stop a logical cluster:
sp_cluster 'logical', 'online', <lcname>[, <instance_list>]
sp_cluster 'logical', 'offline', <lcname>, 'cluster'[, <wait_option>[,<time>[,
@handle output]]]
sp_cluster 'logical', 'offline', <lcname>, 'instance',
<instance_list>[,<wait_option>[, <time>[,
     @handle output]]]

To failover and failback a logical cluster:
sp_cluster 'logical', 'failover', <lcname>, 'cluster'[, <instance_list>[,
<wait_option>[, <time>[,
     @handle output]]]]
sp_cluster 'logical', 'failover', <lcname>, 'instance', <from_instance_list>,
<instance_list>[,
      <wait_option>[,<time>[, @handle output]]]
sp_cluster 'logical', 'failback', <lcname>, 'cluster'[,<instance_list>[,
<wait_option>[, <time>[,
     @handle output]]]]
sp_cluster 'logical', 'failback', <lcname>, 'instance', <from_instance_list>,
<instance_list>[,
      <wait_option>[,<time>[, @handle output]]]

To work with action handles:
sp_cluster 'logical', 'action', <lcname>, 'cancel', <handle>
```

```
sp_cluster 'logical', 'action', <lcname>, 'modify_time', <handle>,
<wait_option>[, <time>]
sp_cluster 'logical', 'action', <lcname>, 'release', <handle>

Argument details:
<wait_option> is one of {'nowait', 'wait', 'until'}
<time> is a time in hh:mm:ss format
<handle> is an action handle
```

## sp_cluster logical, offline

Description

Stops the logical cluster on one or more instances or the entire logical cluster.

Syntax

sp_cluster logical, "offline", *lc_name*, {
cluster |
instance, *instance_list* }
[, *wait_option*[, *timeout*,[, @handle output ]]]

Parameters

*lc_name* – is the name of a logical cluster.

cluster – specifies the entire cluster.

instance – specifies that only certain instances in the logical cluster are to be taken offline.

*instance_list* – is a list of selected instances in the logical cluster.

*wait_option* – is how the time of the action is to be specified. Values are:

- wait – indicates that existing connections are given a specified amount of time (or an infinite amount of time if no *timeout* is given) to migrate or disconnect.
- nowait – indicates that existing connections are migrated or disconnected immediately.
- until – indicates that existing connections are given until a specific time of day to migrate or disconnect.

*timeout* – is a specific amount of time (when used with wait) or a specific time (when used with until). The format is "hh:mm:ss" according to a 24-hour clock. For example, *timeout* records 11:30 pm as 23:30:00.

@handle output – specifies that an action handle is to be retrieved for the action.

Examples

**Example 1** Immediately stops all instances in the “SalesLC”, and places “SalesLC” in the offline state.

```
sp_cluster logical, "offline", SalesLC, cluster,
nowait
```

**Example 2** Stops the “ase1” and “ase2” instances in “SalesLC”, and places “SalesLC” in the offline state.

```
sp_cluster logical, "offline", SalesLC, instance,
"ase1;ase2"
```

Usage

- You cannot use the offline command for the system logical cluster.
- offline is identical to deactivate, except deactivate places stopped instances or clusters in the inactive state.

## sp_cluster logical, online

Description

Starts the default logical cluster on one or more instances.

Syntax

sp_cluster logical, "online", { *lc_name*[, *instance_list* ] }x

Parameters

*lc_name* – is the name of a logical cluster.

*instance_list* – is a list of selected instances in the logical cluster.

Examples

**Example 1** Starts all base instances in the “SalesLC” logical cluster, and brings the cluster online.

```
sp_cluster logical, "online", SalesLC
```

**Example 2** Starts the “ase1” instance in “SalesLC”, and brings the cluster online.

```
sp_cluster logical, "online", SalesLC, ase1
```

Usage

- You cannot use the online command for the system logical cluster.

## sp_cluster logical, set

Description

Sets logical cluster rules: the open logical cluster, the failover mode, the system view, the startup mode, and the load profile.

Syntax

sp_cluster logical, "set", *lc_name*, {
open |
failover, *failover_mode* |
system_view, *view_mode* |
startup, { automatic | manual } |
load_profile, *profile_name* }
login_distribution, { affinity | "round-robin" }

Parameters

*lc_name* – is the name of a logical cluster.

open – sets the open logical cluster. Unrouted connections are sent to the open logical cluster.

failover *failover_mode* – sets the failover mode of the logical cluster. Values for *failover_mode* are:

- instance – specifies a 1:1 failover strategy; every time a base instance fails, a failover resource is brought online.
- group – specifies that failover resources are brought online only after all base instances in the cluster fail.

system_view *view_mode* – sets the default system view for tasks running in the logical cluster. Values for *view_mode* are:

- instance – specifies that monitoring and informational tools such as sp_who, sp_lock, and monitoring tables describe an instance.
- cluster – specifies that monitoring and informational tools such as sp_who, sp_lock, and monitoring tables describe the whole cluster.

startup { automatic | manual }– sets the startup mode of the logical cluster.

- automatic – specifies that the logical cluster is started automatically when the cluster starts.
- manual – specifies that the logical cluster must be started manually.

load_profile *profile_name* – specifies the load profile used to compute load scores for the logical cluster. Values for profile_name are:

- sybase_profile_oltp – is a default profile for OLTP environments supplied by Sybase.
- sybase_profile_dss – is a default profile for primarily read-only, DSS environments supplied by Sybase.
- A user-supplied profile created using sp_cluster profile.

login_distribution – specifies how the Cluster Edition distributes connections when a logical cluster spans multiple instances.

Examples

**Example 1** Sets the load profile for the "SalesLC" logical cluster to the Sybase profile sybase_profile_oltp.

```
sp_cluster logical, "set", SalesLC, load_profile,
sybase_profile_oltp
```

**Example 2** Sets the default system view to cluster.

```
sp_cluster logical, "set", SalesLC, system_view,
cluster
```

Usage

- Only one logical cluster can have the open property. When you set the open property to a new logical cluster, the open property is removed from the previous open logical cluster.

## sp_cluster logical, show

Description

Displays information about a logical cluster.

Syntax

sp_cluster 'logical', "show"[, *lc_name*[, {*action*[, *state*] | route[, *type*[, *key*]]}]]

Parameters

*lc_name* – is the name of the logical cluster. If NULL is entered, summary information for all logical clusters is displayed.

action – specifies information about administrative actions: failover, failback, online, offline, deactivate.

*state* – is one of: cancelled, complete, or active .

route – specifies information about routes.

*type* – is one of: application, alias, or login.

*key* – is a a specific login, alias, or application name.

Examples

**Example 1** Displays summary information for all configured logical clusters.

```
sp_cluster logical, "show", NULL
```

```
ID  Name            State   Online Instances   Connections
--- -------         ------- ------------------ ---------------
1   mycluster       online         4                  1
2   SalesLC         online         2                  0
3   HRLC            online         1                  0
4   CatchallLC      offline        0                  0

Logical cluster 'mycluster' is the system logical cluster.
Logical cluster 'CatchallLC' is the open logical cluster.

Logical Cluster   Instance    State    Type     Connections         Load Score
----------------- --------   ------    ------   --------------      -----------
HRLC               silk      online     base              0           0.01
SalesLC            cotton    offline    failover          0           0.00
SalesLC            linen     online     base              0           0.00
SalesLC            silk      offline    failover          0           0.01
SalesLC            wool      online     base              0           0.01
mycluster          cotton    online     base              0           0.00
mycluster          linen     online     base              0           0.00
```

```
mycluster          silk      online     base              0           0.01
mycluster          wool      online     base              1           0.01
```

**Example 2** Displays a list of all outstanding actions.

```
sp_cluster logical, "show", NULL, action
```

**Example 3** Displays information for the SalesLC logical cluster.

```
sp_cluster logical, "show", SalesLC
ID          Name            State          Online Instances       Connections
----------- --------------- -------------- ---------------------- -----------
2           OrderLC         online         1                      0


Instance          State         Type      Connections Load Score  Failover Gro
---------------   -----------   -------   ----------  ----------  ------------
asedemo1          online        base                0        0.78            NU

Attribute                              Setting
------------------                     -----------------------------------
Down Routing Mode                      system
Failover Mode                          instance with fail_to_any
LC Roles                               none
Load Profile                           sybase_profile_oltp
Login Distribution                     affinity
Startup Mode                           automatic
System View                            cluster

Route Type             Route Key
---------------------- ------------------
application            order_app

Logical cluster 'OrderLC' has no associated actions.
(return status = 0)
```

## sp_cluster profile

Description

Lets you set up and manage the load profile for the logical cluster.

Syntax

```
sp_cluster profile, [ "show" [, profile_name ] |
     "create", profile_name |
     "drop", profile_name |
     "set", profile_name [, weight [, wt_metric [, wt_value ] ] |
          threshold [, thr_metric [, thr_value ] ] ]
```

Parameters

- show – displays configured load profiles and their settings.
- *profile_name* – is the name of a load profile.
- creates – creates a new load profile.
- drop – drops a load profile.
- set – specifies attributes of a load profile. You must set each attribute individually.
- weight – specifies a weight attribute.
- *wt_metric* – specifies an individual weight metric. Values are:
  - user connections – the capacity of an instance to accept a new connection, based on resource availability.
  - cpu utilization – the capacity of an instance to accept a new connection, based on resource availability.
  - run queue – the capacity of an instance to accept a new connection, based on resource availability.
  - io load – outstanding asynchronous I/Os.
  - engine deficit – the difference in the number of online engines among instances in the cluster.

    **Note** engine deficit is measurable only when instances in the cluster have unequal numbers of engines. engine deficit adds a metric for maximum relative capacity to the load score.

  - user metric – an optional, user-supplied metric.
- *wt_value* – specifies a weight value. Valid values are 0 to 255. A weight of zero (0) excludes the metric from calculation.
- threshold – specifies a threshold attribute.
- *thr_metric* – specifies a particular threshold attribute. Values are:
  - dynamic – specifies a threshold for dynamic load distribution.
  - login – specifies a threshold for login redirection
  - hysteresis – specifies a minimum load score for any connection redirection.
- *thr_value* – depends on value of *thr_metric*:

- When *thr_metric* is dynamic or login, *thr_value* is the percentage difference between the the load scores of two instances. Valid values are 0 to 100. A weight of zero (0) disables this form of load distribution.
- When *thr_metric* is hysteresis, *thr_value* is the minimum load score for the target instance that must be met before dynamic load distribution or login redirection can occur.

Usage

- The user metric value must be normalized so that it is compatible with values for metrics provided by Sybase. Consider a user metric that measures response times. If the maximum acceptable response time is 10 seconds and the measured value is 5, the metric value is 50 (5/10 x 100 = 50).
- Threshold metrics let you configure at what point a load imbalance should cause connections to be redirected from one instance to another. The workload manager redirects connections when the load score difference (as a percent) between the target instance and the least loaded instance meets or exceeds the threshold value.

  The hysteresis value guards against redirection when the load score difference meets the threshold value, but the instance load scores (for example, 2 and 8) are so low that redirection is not appropriate.

Examples

**Example 1** Creates the load profile “my_profile”:

```
sp_cluster profile, "create", my_profile
```

**Example 2** Specifies the metric weights for “my_profile.” “user connections” is set to zero, which excludes that metric from the profile.

```
sp_cluster profile, "set", my_profile, weight,
"user connections", '0'
sp_cluster profile, "set", my_profile, weight,
cpu utilization, '20'
sp_cluster profile, "set", my_profile, weight,
runqueue, '30'
sp_cluster profile, "set", my_profile, weight,
io load, '10'
sp_cluster profile, "set", my_profile, weight,
engine deficit, '10'
sp_cluster profile, "set", my_profile, weight,
user metric, '30'
```

For more information, see “Using Monitoring Tables in a Clustered Environment.”

**Example 3** Sets the login redirection threshold to 80 and the hysteresis value to 10 for "my_profile."

```
sp_cluster profile, "set", my_profile, threshold, login, '80'
sp_cluster profile, "set", my_profile, threshold, hysteresis, '10'
```

**Example 4** Displays information about a configured profile.

```
sp_cluster profile, "show", my_profile

ID            Profile                   Type    Connections   CPU  Run     Queue
------------ ------------------------- ------- ------------ --- -------------
--------- -------- ------ -------  ------  ---------
          100 my_profile                user             0     20            30
                 10     10      30      30     0     20

Profile                    Logical Cluster
-------------------------  --------------
my_profile                  SalesLC


Profile                      Logical Cluster Instance

         Load Score                      Connections Score
         CPU Score                       Run Queue Score
         IO Load Score                   User Score
--------------------------------- ------------------- -----------
       ------------------------------------ ------------------------------
       ------------------------------------ ------------------------------
       ------------------------------------ ------------------------------
----------------------------------
my_profile                          SalesLC          ase1
                                    0.028871                 0.000000
                                    0.028871                 0.000000
                                    0.000000                 0.000000
                                    0.000000
my_profile                                           ase2
                                    0.029474                 0.000000
                                    0.029474                 0.000000
                                    0.000000                 0.000000
                                    0.000000
my_profile                                           ase3
                                    0.019503                 0.000000
                                    0.019503                 0.000000
                                    0.000000                 0.000000
                                    0.000000
my_profile                                           ase4
                                    0.582675                 0.000000
```

```
0.290930                0.291745
0.000000                0.000000
 0.000000
```

## sp_tempdb_markdrop

Description

Places a local system temporary database in the drop state.

Syntax

sp_tempdb_markdrop *database_name* [, {'mark' | 'unmark'}]

Parameters

- *database_name* – is the name of the local system temporary database you are dropping.
- mark – marks the specified database for dropping.
- unmark – clears the mark from the database.

Examples

This example marks a local system temporary database named "old_cluster_tempdb1" to be dropped:

```
sp_tempdb_markdrop 'old_cluster_tempdb1', 'mark'
```

This example removes the mark from the local system temporary database "old_cluster_tempdb1":

```
sp_tempdb_markdrop 'old_cluster_tempdb1, 'unmark'
```

Usage

To delete the last local temporary database:

1 Use sp_tempdb_markdrop to place the local system temporary database in the drop state.

2 Shut down and restart the instance that owns the last local temporary database.

**Note** After you mark the local system temporary database to be dropped, the owner instance restarts if there are no other active instances. This instance does not use the marked local system temporary database when it starts.

3 Use drop database to delete the last local system temporary database.

## sp_refit_admin

Description

Provides an interface to perform various disk refit-related actions, such as showing the current status of the disk refit process, resetting the state of the disk refit process, skipping the disk refit process for an instance, and so on.

**Note** See "Using Temporary Databases" on page 139 for information on the two-phase disk refit process.

Syntax

```
sp_refit_admin ['help']
sp_refit_admin 'status'
sp_refit_admin 'reset' [, instance_name]
sp_refit_admin 'skiprefit'[, instance_name]
sp_refit_admin 'removedevice', device_name
```

Parameters

help – displays information on sp_refit_admin syntax and usage.

status – displays the current status of the disk refit process. It lists all the instances and their private devices for which disk refit is still pending. If no such device exists, it prints a message saying so.

reset – resets the state of the disk refit process. It takes an optional parameter *instance_name*.

If *instance_name* is not supplied, this parameter resets the disk refit process back to the beginning of Phase One, so that subsequent disk refit command starts the disk refit process from Phase One and refits all the regular shareable devices, as well as private devices of the instance.

If *instance_name* is supplied, this parameter resets the disk refit process back to the beginning of Phase Two for that instance, so that a subsequent disk refit command on that instance starts the disk refit process from Phase Two for that instance, and refits only the private devices of that instance.

skiprefit – skips running Phase Two of the disk refit process for one or all instances in the cluster without dropping the device. This parameter is meaningful only after the completion of Phase One of the disk refit process. It takes *instance_name* as an optional parameter.

**Note** The refit-pending devices of the skipped instance are not refitted, so the databases on these devices are not available for use.

removedevice – removes a device from the disk refit process. This parameter requires the name of the device that is to be removed, as the input parameter *device_name* or *instance_name*.

*instance_name* – the name of the instance, valid only with reset and skiprefit.

*device_name* – the name of the device, valid only with removedevice.

Examples

**Example 1** Resets the state of the disk refit process to the start of Phase One.

```
sp_refit_admit 'reset'
```

After executing reset, the user must run Phase One and Phase Two of the disk refit process.

**Example 2** Resets the state of the disk refit process on the instance named 'cluster1_instance1' to the start of Phase Two for the instance.

```
sp_refit_admin reset, 'cluster1_instance1'
```

This interface removes sysdatabases entry for all the databases created on the private devices owned by 'cluster1_instance1', and the sysusages entries corresponding to the private devices owned by 'cluster1_instance1'. After executing, you must run Phase Two of disk refit on 'cluster1_instance1'.

**Example 3** Skips the disk refit process of all the refit-pending private devices of instance `'cluster1_instance1'`.

```
sp_refit_admin 'skiprefit', 'cluster1_instance1'
```

This example removes the sysdatabases entry for all the databases that use any of the refit-pending private devices owned by `'cluster1_instance1'`, and removes all the entries in sysusages for all the deleted databases.

To skip the disk refit process on all the refit-pending private devices of all the instances in the cluster, enter:

```
sp_refit_admin 'skiprefit'
```

**Example 4** To remove the device "device1' from the disk refit process:

```
sp_refit_admin 'removedevice', 'device1'
```

This action removes the sysdatabases entry for all databases created on `'device1'`, and all the sysusages entries corresponding to `'device1'`. It also removes `'device1'` from sysdevices.

Usage

- You must follow the instructions in “Troubleshooting” on page 175 after executing skiprefit, to ensure the consistency of the system tables before resuming normal operation.
- Use removedevice only during the disk refit process, to remove the device from the refit process. Do not use it in place of sp_dropdevice
- You can use sp_refit_admin even when the instance is started with the -m option and trace flag 3608 ON.

Permissions — Only a user with system administrator permissions can execute this procedure.

See also — For information on problems encountered with disk refit, see the *Troubleshooting and Error Guide*.

# Changed stored procedures

## Configuration changes

In the Cluster Edition, Sybase supports both cluster-wide and instance-specific configuration. Cluster-wide configuration parameters are applied to all instances in the cluster. Local configuration parameters are applied only to a specified instance.

- Local configuration overrides cluster-wide configuration.
- If an instance-specific configuration has not been applied, the cluster-wide configuration applies.
- Some parameters, such as default character set id, cannot be applied to a specific instance. These parameters can only be used over an entire cluster.

Changes to the configuration file format

The configuration file format now includes an instance-specific configuration block. Parameter settings in the instance-specific block override cluster-wide settings. For example:

```
max online engines = DEFAULT

[Instance:ase1]
max online engines = 5
[Instance:ase2]
max online engines = 3
```

## sp_audit

New cluster option

sp_audit for the Cluster Edition includes the cluster option which enables you to audit cluster commands. The syntax for sp_audit is:

sp_audit ‘cluster’, *login_name*, *object_name* [,*setting*]

Once enabled, the cluster option audits these commands on the cluster:

- sp_cluster 'logical', 'online', *cluster_name*
- sp_cluster 'logical', 'failover', *logical_cluster_name*, 'cluster' [, *instance_list* [, *wait_option* [, *time* [, *@handle output*]]]]

- sp_cluster 'logical', 'failback', *logical_cluster_name*, 'cluster'
- sp_cluster 'logical', 'failover', *logical_cluster_name*, 'instance', *from_instance_list*, *instance_list* [, *wait_option* [, *time* [, @handle output]]]]
- sp_cluster 'logical', 'failback', *logical_cluster_name*, 'instance', *from_instance_list*, *instance_list* [, *wait_option* [, *time* [, *@handle output]*]]
- sp_cluster 'logical', 'offline', *cluster_name*, 'instance', *instance_list*
- sp_cluster 'logical', 'offline', *cluster_name*, 'cluster'
- sp_cluster 'logical', 'action', *logical_cluster_name*, 'cancel', *handle*
- sp_cluster 'logical', 'action', *logical_cluster_name*, '*modify_wait*', *handle*, *wait_option* [, *time*]
- sp_cluster 'logical', 'action', <lcname>, 'release', <handle>
- sp_cluster 'logical', 'drop', *logical_cluster_name*, 'cluster'

Auditing nodes

You must first run the command:

```
sp_audit 'security', 'all', 'all'
```

to audit option changes to nodes.

## sp_addengine

`sp_addengine` includes the *instance_id* parameter to add an engine or engine group to a specific instance. The syntax is:

sp_addengine engine_number, engine_group [, instance_id]

Where *instance_id* is the id of the instance to which you are adding an engine or engine group. For example, this adds engine number 5 to instance id 8:

```
sp_addengine 5, 8
```

If sp_cluster set *system_view* is set to cluster, you can add an engine or engine group to any instance in the cluster. If *system_view* is set to instance, you can add and engine or engine group only to a local instance.

## sp_addexeclass

`sp_`addexeclass includes the *instance_id* parameter which allows you to create or update a user-defined execution class and bind it to a specific instance. The syntax is:

`sp_addexeclass` *`class_name`*`, priority, timeslice,` *`engine_group`* `[,` *`instance_id`*`]]`

where *instance_id* is the id of the instance to which you are binding a user-defined execution class. For example, to define a new execution class called DS with a priority value of LOW and associate it with the engine group DS_GROUP on engine number 8, enter:

```
sp_addexeclass "DS", "LOW", 0, "DS_GROUP", 8
```

If sp_cluster set *system_view* is set to cluster, you can add an execution class on any instance in the cluster. If the *system_view* is set to instance, you can add an execution class only to a local instance.

## sp_addserver

sp_addserver includes the *filter* parameter to add a remote server for remote procedure calls (RPCs). The syntax is:

sp_addserver '*logical_server_name*', ASEnterprise, '*host*:*port*:*filter*'

where filter is:

```
ssl[="CN=common_name"]
```

For example, the following adds the “new_logical_server” server:

```
sp_addserver 'big_logical_server', ASEnterprise, 'maynard:23954:ssl=
"CN=ase1.big server 1.com"'
```

The rules for common names are same as those used for dynamic listeners and the directory service entries.

You can use this format to declare the *host:port* number:

```
ip_address:port
```

## sp_addlogin

sp_addlogin fails if you attempt to include a local temporary database as a default database for any login.

## sp_dbcc_updateconfig

sp_dbcc_updateconfig includes new two options: enable excluded faults inserts and enable dbcc_counter inserts.

- enable excluded faults inserts specifies whether or not excluded faults are inserted in the dbcc_faults and dbcc_fault_params tables during dbcc checkstorage. The default value for enable excluded faults inserts is off (0). The syntax is:

sp_dbcc_updateconfig {*db_name* | null }, "enable excluded faults inserts", ['0' | '1']

Use sp_dbcc_exclusions to specify excluded faults.

- enable dbcc_counter inserts specifies whether or not rows are inserted in the dbcc_counters table when dbcc checkstorage completes. The default value for enable dbcc_counter inserts is off (0). The syntax is:

  sp_dbcc_updateconfig {*db_name* | null }, "enable dbcc_counter inserts", ['0' | '1']

Two new rows have been added to the dbcc_types table to enable these changes. See "dbcc_types" on page 307. See the *Reference Manual: Tables* for a description of the dbcc_counters table.

## sp_configure

Syntax changes for the Cluster Edition:

- You can now specify an instance name to set an instance-specific option. The syntax is:

  sp_configure [*configuration_option* [, *config_value* [,...]]] [, *instance_name*]

- A new option, drop instance, allows you to drop an instance-specific configuration setting. The syntax is:

  sp_configure *configuration_option*, 0, "drop instance", *instance_name*

Usage changes for display:

- If no configuration option or instance name is specified, the information displayed depends on the system_view setting.
- If no configuration option is specified, and the instance name is specified, Adaptive Server displays all instance-specific configuration settings for the specified instance.
- If the configuration option is specified, but the configuration value and instance name are not specified, Adaptive Server displays the current settings for the specified option for all instances under the "cluster" view. If the instance name is specified, Adaptive Server displays configuration information for the specified instance.

Usage changes when setting options:

- If the configuration option and value are specified, but no instance is specified, Adaptive Server configures the cluster-wide setting for the option. If, however, the instance name is specified, Adaptive Server sets the configuration value for the instance only. The syntax is:

  sp_configure *configuration_name*, *config_value*, NULL, *instance_name*

- You cannot set configuration options from inside a local temporary database.
- If an instance already has instance-specific setting for a configuration parameter set, you can reconfigure this parameter for a cluster-wide setting.
- A user can reconfigure only those instances to which they are connected.

## sp_dropengine

`sp_dropengine` includes the *instance_id* parameter to drop an engine or engine group from a specific instance. The syntax is:

sp_dropengine engine_number, engine_group [, instance_id]

Where *instance_id* is the id of the instance from which you are dropping an engine or engine group. For example, this drops engine number 5 from instance id 8:

```
sp_dropengine 5, 8
```

If sp_cluster set *system_view* is set to cluster, you can drop an engine or engine group from any instance in the cluster. If the *system_view* is set to instance, you can drop and engine or engine group only from a local instance.

## sp_encryption

The command, `sp_encryption help, key_name, display_cols`, displays only encrypted columns in the system database and local temporary database that are owned by the instance where the command is run. This command skips the local temporary database owned by a remote instance and displays a message similar to:

```
Local temporary database '<database name>' is skipped
for this operation. The database is accessible from the
owner instance '<instance name>' only. Execute this
procedure on ASE instance '<instance name>' to see any
columns encrypted with key '<keyname>' on database
'<database name>'.
```

You must run the command on the remote instance to display the remote instance's columns encrypted with the specified key on local temporary database.

## sp_helpconfig

Syntax changes to sp_helpconfig:

- A new option, cluster options, displays all strictly cluster-wide configuration options. The syntax is:

  sp_helpconfig "cluster options"

Usage changes:

- If system_view is set to cluster, sp_helpconfig displays configuration information for all instances in the cluster.
- If system_view is set to instance, sp_helpconfig displays configuration information for the current instance.

## sp_helpdb

sp_helpdb does not display device-related information if the specified database is a local temporary database owned by a remote instance.

## sp_modifylogin

sp_modifylogin fails if you attempt to include a local temporary database as a default database for any login.

## sp_sysmon

sp_sysmon report includes these enhancements:

- New categories in the Task Management section. These new rows identify context switches related to different types of lock acquisition. For example:

```
Task Management             per sec   per xact   count   % of total
--------------------        --------  --------   ------  ----------
...

Text Context Switches Due To:
...

Physical Lock Transition      208.9.      0.4    26108        4.6%
Logical Lock Transition      2209.3       4.5   276160       48.5%
...

Interconnect Message Sleeps    612.2       1.2    76527        13.5%
```

- New “Grabbed Locked Buffer” output in the Pool Turnover section. Grabbed Locked Buffer displays the percentage of buffers grabbed with a physical lock. For examplet:

```
Task Management             per sec   per xact   count   % of total
--------------------        --------  --------   ------  ----------
...

Pool Turnover
 4 Kb Pool
LRU Buffer Grab               89.2        0.2    11154      100.0%
Grabbed Locked Buffer          0.0        0.0        0        0.0%
Grabbed Dirty                  0.0        0.0        0        0.0%
---------------             -------    -------   ------
Total Cache Turnover          89.2        0.2    11154
```

## *instance_name* parameter added

These system procedures add the instance_name parameter to their syntax.

### sp_cacheconfig

sp_cacheconfig includes the instance *instance_name* parameter so you can create, drop, or modify the size of a cache on a specific instance.

Syntax

```
sp_cacheconfig "[cachename [,cache_size [P|K|M|G]"
[,logonly | mixed ] [,strict | relaxed ] ]
        [, "cache_partition = [1|2|4|8|16|32|64]"]
[, "instance instance_name"]
```

where *instance_name* is the name of the instance whose cache you are adjusting.

Examples

This example displays the cache for instance blade1:

```
sp_cacheconfig 'instance blade1'
```

This example sets the size of the Sales Cache size on blade1 to 100 megabytes:

```
sp_cacheconfig 'Sales Cache', '100M', 'instance blade1'
```

This example sets the size of the Sales Cache size on blade1 to 0 megabytes, effectively dropping the cache.

```
sp_cacheconfig 'Sales Cache', '0M', 'instance blade1'
```

This example displays the cache configuration for the entire cluster:

```
sp_cacheconfig
```

Usage

- If you do not specify an instance_name, the cache for the cluster is displayed.

**sp_poolconfig**

sp_poolconfig includes the instance *instance_name* parameter so you can create, drop, or modify the size of a buffer pool for a named cache on a specific instance.

Syntax

sp_poolconfig *cache_name* [, "*mem_size* [P\K\M\G]", "*config_pool*K" [, "affected_poolK"]] [, 'instance *instance_name*']

where *instance_name* is the name of the instance whose buffer pool you are adjusting.

Examples

Creates a a 16KB buffer pool of size 25MB in the default data cache on instance blade1:

```
sp_poolconfig 'default data cache', '25M', '16K', 'instance blade1'
```

Displays the buffer pool configuration in the default data cache on instance blade1:

```
sp_poolconfig 'default data cache', 'instance blade1'
```

Displays the buffer pool configuration for named cache c_log on all instances in the cluster:

```
sp_poolconfig c_log
```

**sp_helpcache**

Allows you to display cache information for a specific instance. If you do not specify an *instance_name*, sp_helpcache displays information for all caches.

Syntax

[sp_helpcache[cache_name [,'sizeP|K|M|G']['instance *instance_name*'

where *instance_name* is the name of the instance whose cache you are investigating.

Examples

Displays cache information for all caches:

```
sp_helpcache
```

Displays the overhead for the cache C2 on instance “blade1” for size 10M:

```
sp_helpcache 'C2', '10M', 'instance blade1'
```

## sp_tempdb

Description

sp_tempdb allows users to:

- Create and manage temporary database groups.
- Bind users or applications to the default or other temporary database group or to a specific local temporary database.
- Manage bindings to local temporary databases and temporary database groups.

These bindings are stored in the sysattributes table in master database.

sp_tempdb provides the binding interface for maintaining bindings in sysattributes that are related to the multiple temporary database.

---

**Note** This reference page contains information specific to the Cluster Edition. See the *Reference Manual* for complete information for sp_tempdb.

---

Syntax

sp_tempdb [
[ { “create” | “drop” } , “*groupname*” ] |
[ { “add” | “remove” } , “*tempdbname*”, “*groupname*” ] |
[ { “bind”, ““*objtype*”, “*objname*”, “*bindtype*”, “*bindobj*”
[, “*scope*”, “*hardness*” ] } |
[ { “unbind”, “*objtype*”, “*objname*”
[, “scope” **[, "*instance_name*",]** ] } ] |
[ “unbindall_db”, “*tempdbname*” ] |
**[ "unbindall_gr", "***groupname***" ] |**
[ show [, "all" | "gr" | "db" | "login" | "app" [, “*name*” ] ] |
[ who, “*dbname*” ] |
[ help ] ]

Parameters

- create – creates a temporary database group.
- drop – drops a temporary database group.
- *groupname* – is the name of the temporary database group.

- add – adds a temporary database to a temporary database group.
- remove – removes a temporary database from a temporary database group.
- *tempdbname* – is the name of the temporary database you are adding or removing.
- bind –binds logins and applications to a temporary database or temporary database group.
- unbind – unbinds logins and applications to a temporary database or temporary database group.
- *instance_name* – is the name of the instance owning the local temporary database that is to be unbound. This option is for the Cluster Edition only.
- *objtype* – is the object type. Valid values are:
  - login_name (or LG)
  - application_name (or AP)

  Values are not case-sensitive.
- *objname* – is the name of the object you bind or unbind.
- *bindtype* – is the bind type. Valid values are:
  - group (or GR)
  - database (or DB)

  Values are not case sensitive.
- *bindobj* – is the name of the object being bound, and is either a group or a database depending on the *bindtype*.
- *scope* – NULL
- hardness – is hard, soft, or NULL. The default is soft.
  When you set the value of *hardness* to hard, a failure to assign a temporary database according to the binding results in a failure of the login.

  When you set the value to soft, such a failure results in the assignment of a temporary database from the default group or a local system temporary database.
- unbindall_db – removes all login and application bindings for a given temporary database. It does not remove any database to group memberships. The *tempdbname* variable is required with this option.

  Existing assignments to active sessions are not affected by this operation.

- unbindall_gr *groupname* – removes all login and application bindings for a given temporary database group.
- show –displays information stored in the sysattributes table about the existing groups, group members, login and application bindings, and active sessions that are assigned to a given database. The values are:
  - all or no argument – displays all temporary database groups, all database-to-group memberships, and all login and application bindings.
  - gr – displays all temporary database groups. If you specify *name*, displays all members of the group.
  - db – displays all databases to group memberships. If you provide *name*, then only the database-to-group memberships for the database *name* are printed.
  - login – displays all login bindings where login is not NULL. If you provide *name*, only the bindings for the login *name* are printed.
  - app – displays all bindings where the application is not NULL. If you provide *name*, the bindings for the application *name* are printed.
- who – displays all active sessions assigned to the given temporary database. When using the who parameter, you must use:
  - *dbname* – is the name of a temporary database. If you provide a nontemporary database name for *dbname*, sp_tempdb who executes, but does not report any active sessions bound to it.

    If system_view is set to cluster, all active sessions of the cluster are examined. If system_view is set to instance, sessions that are active on the current instance are examined

    This command may be executed from any instance in the cluster.
- help – displays usage information. Executing sp_tempdb without specifying a command is the same as executing sp_tempdb “help”.

Usage

- For the Cluster Edition, *tempdbname* must be a local user temporary database.

# Configuration parameters

This section describes new and changed configuration parameters.

## New configuration parameters

The Cluster Edition introduces the "shared-disk cluster" configuration group in the server configuration file. The configuration parameters in this group are defined below.

### cluster heartbeat interval

| Summary information | |
|---|---|
| Default value | 10 |
| Valid values | 1– 127 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

cluster heartbeat interval controls the interval that cluster instances use to send and check the heartbeat status.

Using a lower value for cluster heartbeat interval reduces the failure detection time but increases the risk of a false failure because of a transient problem (such as an overloaded CPU). Tuning cluster heartbeat interval to a larger value reduces the risk of a false failure but increases the time needed to detect a failure.

### cluster heartbeat retries

| Summary information | |
|---|---|
| Default value | 1 |
| Valid values | 1– 127 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

cluster heartbeat retries controls the number of times an instance retries a failed cluster heartbeat before entering failure mode.

Tuning cluster heartbeat retries to a lower value reduces the time to detect failure but increases the risk of a false failure because of a transient problem (such as an overloaded CPU). Tuning cluster heartbeat retries to a larger value reduces the risk of a false failure but increases the time needed to detect a failure.

## cluster vote timeout

| Summary information | |
|---|---|
| Default value | 60 |
| Valid values | 1– 127 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

cluster vote timeout controls the maximum amount of time an instance waits for other instances to vote during the voting period. An instance waits only for those instances which it believes are running.

Tuning cluster vote timeout to a lower value can reduce failover time, but increases the risk that an instance that is running is excluded from the new cluster view. Tuning cluster vote timeout to a larger value reduces the risk that an running instance is excluded from the new cluster view, but may increase failover time.

## idle migration timer

| Summary information | |
|---|---|
| Default value | 60 |
| Valid values | 0 – 32767 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

idle migration timer specifies the amount of time after which an idle connection is closed without invalidating the migration request sent to the client, allowing you to stop an instance after a specified period of time without waiting for idle client connections to migrate.

Setting idle migration timer to a high value slows down a graceful shutdown because the instance must wait the specified period of time for all idle connections that issued a migration request without the client having initiated migration.

## session migration timeout

| Summary information | |
|---|---|
| Default value | 600 |
| Valid values | 0 – 32767 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

session migration timeout sets the amount of time an instance stores a client's session id. A high value for session idle timer allows inactive connections to migrate successfully a long time after migration.

## quorum heartbeat interval

| Summary information | |
|---|---|
| Default value | 5 |
| Valid values | 1 – 60 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

quorum heartbeat interval specifies the number of seconds between quorum heartbeats. Setting quorum heartbeat interval to a lower number increases the heartbeat overhead but speeds the detection of a lost disk link, resulting in a quicker termination of an instance for which you have set IO fencing or that has lost its SAN link. Setting quorum heartbeat interval to a high number reduces heartbeat overhead, but delays the detection of a lost disk link.

## quorum heartbeat retries

| Summary information | |
|---|---|
| Default value | 2 |
| Valid values | 0 – 32,768 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

quorum heartbeat retires specifies the number of times an instance retries a failed quorum heartbeat before exiting. A value of 2 indicates that an instance terminates after the third consecutive quorum heartbeat failure. Tuning quorum heartbeat retires to a low number causes a faster failover when access to the quorum device is lost on a given instance, possibly improving an application's recovery time as the risk of failing an instance due to a transient condition is reduced. Setting this parameter to a high number degrades application recovery after an instance loses access to the shared disks, reducing the chances that a transient disk access problem can cause an instance failure.

**Note** Setting quorum heartbeat interval to the default (5) and quorum heartbeat retries to its default (2) implies the time from disk failure detection to instance shutdown is between 10 and 15 seconds, consisting of 0 to 5 seconds for the first heartbeat failure and 5 seconds for each of two retries.

## automatic cluster takeover

| Summary information | |
|---|---|
| Default value | 1 |
| Valid values | 1 (enabled), 0 (disabled) |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

Setting automatic cluster takeover to 1 allows an instance that is starting to automatically recover from an abrupt total cluster failure. If you set automatic cluster takeover to 0, the cluster may not be able to recover from an abrupt cluster failover unless you include the --cluster_takeover parameter.

The Cluster Edition uses quorum heartbeats and a cluster takeover algorithm to determine when cluster takeover should be performed. This algorithm allows an instance that is starting to distinguish between an inability to join the cluster because the cluster has crashed (in which case takeover is appropriate) and an inability to join the cluster because the instance that is starting does not have network connectivity (in which case takeover is not appropriate).

If you disable automatic cluster takeover (set it to 0), The Cluster Edition writes the results of the algorithm to the error log as an advisory message and then exits.

If you enable auotomatic cluster takeover (set it to 1), the Cluster Edition starts as the cluster coordinator and recovers the databases. This is guaranteed to be a safe operation in environments that have I/O fencing enabled.

In environments without I/O fencing, a malfunction of the algorithm could introduce data corruption, so you can set the configuration parameter to 0 to disable this algorithm. However, environments without I/O fencing have a risk of data corruption, and disabling automatic cluster takeover does not mitigate all of those risks.

## enable i/o fencing

| Summary information | |
|---|---|
| Default value | 0 |
| Valid values | 1 (enabled), 0 (disabled) |
| Status | Static |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

Setting enable i/o fencing to 1 enables I/O fencing for each database device that supports the SCSI-3 Persistent Group Reservation (PGR) standard.

## enable backupserver HA

| Summary information | |
|---|---|
| Default value | 1 |
| Valid values | 1 (enabled), 0 (disabled) |
| Status | Dynamic |
| Display level | Comprehensive |

| Summary information | |
|---|---|
| Required role | System administrator |
| Configuration group | Shared disk cluster |

Setting enable backupserver HA to 1 starts the high availability Backup Server for the cluster. Setting enbale backupserver HA to 0 disables the high availability Backup Server on the cluster.

## CIPC large message pool size

| Summary information | |
|---|---|
| Default value | 512 |
| Valid values | 512 – 2147483647 |
| Status | Static |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

CIPC large message pool size specifies the number of large message buffers allocated by CIPC at start-up time.

## CIPC regular message pool size

| Summary information | |
|---|---|
| Default value | 8192 |
| Valid values | 2048 – 2147483647 |
| Status | Static |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

CIPC regular message pool size specifies the number of regular message buffer allocated by CIPC at start-up time.

## DMA object pool size

| Summary information | |
|---|---|
| Default value | 4096 |

| Summary information | |
|---|---|
| Valid values | 2048 – 2147483647 |
| Status | Static |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

DMA object pool size specifies the number of DMA (direct memory access) objects allocated by CIPC at start-up time.

### workload manager cache size

| Summary information | |
|---|---|
| Default value | 80 |
| Valid values | 80 – 2147483647 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Shared disk cluster |

workload manager cache size specifies the maximum amount of memory, in 2K pages, that the workload manager can use. For more information, see Chapter 5, “Managing the Workload.”

## Changed configuration parameters

### number of open databases

In non-clustered versions of Adaptive Server, the minimum value for number of open databases was 5. For the Cluster Edition, the minimum value for number of open databases is 6.

### number of pre-allocated extents

- In earlier versions of Adaptive Server, the maximum value allowed for number of pre-allocated extents was 31. For the Cluster Edition the maximum values for number of pre-allocated extents is 32.

Using a value of 32 for number of pre-allocated extents has a special significance for configuration and impacts the space allocations Adaptive Server performs internally. If you set number of pre-allocated extents to 32, Adaptive Server reserves an entire allocation unit worth of extents for utility operations like BCP-in and select into, both of which use the large-scale allocation scheme of space reservation. This greatly improves the performance of these utilities, particularly when you run them concurrently on multiple nodes. Consequently, using a value of 32 guarantees that each node of a cluster is able to work independently on its own allocation unit without interference from the other nodes.

- In earlier versions of Adaptive Server, the number of pre-allocated extents parameter specified the number of extents reserved in a single allocation call for tables of all sizes.

  With this version of Adaptive Server, the value of number of pre-allocated extents is ignored for large tables with 240 or more pages for these commands only:

  - alter table *table_name* add *column_name* . . .
  - alter table *table_name* modify *column_name* . . .
  - alter table *table_name* drop *column_name* . . .
  - alter table lock . . .
  - reorg rebuild

  When you run these command on tables larger than 240 pages, Adaptive Server reserves an entire allocation unit (32 extents), which greatly improves performance, particularly when you run them concurrently on multiple nodes.

  The value of number of pre-allocated extents continues to be observed for the above commands for tables with fewer than 240 pages, and for all commands (such as select into, bcp, alter table partition) for tables of all sizes.

## runnable process search count

In non-clustered versions of Adaptive Server, the default value for runnable process search count is 2000. For the Cluster Edition, the default value for runnable process search count is 3. The change from 2000 to 3 ensures that Adaptive Server does not overuse the CPU during system idle times.

runnable process search count controls the number of times an Adaptive Server engine loops while looking for a runnable task before releasing the CPU to the operating system. A low value for runnable process search count is useful when other applications (for example, Adaptive Server clients, operating system kernel threads, or non–Adaptive Server applications) run on the same machine as Adaptive Server.

With a runnable process search count value of 3, the Cluster Edition can better share the system CPU with other processes running on the same machine. However, if runnable process search count is 3 and Adaptive Server is running as a stand-alone process, users may experience delays in server response times. In this case, reset runnable process search count to 2000.

# Utilities

This section describes new and changed utilities.

## New utilities

### sybcluster

The sybcluster utility lets you set up, configure, and manage a shared-disk cluster. See the Cluster Edition *Installation Guide* for instructions on how to configure the cluster using sybcluster; see Chapter 14, "The sybcluster Utility," for syntax and usage information.

### qrmutil

A command line utility that allows you to back up, restore, and reconfigure the quorum device. qrmutil is located in *$SYBASE/$SYBASE_ASE/bin*.

Syntax

```
--additional_run_parameters=parameter_list
--buildquorum=[force]--cluster_take_over
--config_file=file_name
--diag={all | boot | toc | nodes | locks | config | cms}
--display={boot | nodes | heartbeat | master | cluster |
          instance | config | state}
--drop_cluster=[force]
--drop_instance=instance_name
--errorlog=file_name
```

--extract_config=*file_name*
-h, --help
-F, --cluster_input=*file_name*
--fence_capable=*device_path*
-s, --instance=*instance_name*
--instance_node=*node_name*
--interfaces_dir=*path_to_interfaces_file*
--max_instances=*number_of_instances*
--master_dev=*master_device*
--primary_address=*interconnect_address*
--primary_port=*port_number*
--primary_protocol=*protocol*
-Q, --quorum_dev=*quorum_device*
--register_node=*node_name*
--secondary_address=*interconnect_address*
--secondary_port=*port_number*
--secondary_protocol=*protocol*
--traceflags=*traceflag_list*
--unregister_node=*node_name*
--verify_node=*node_name*
-v, --version]

Parameters

- --additional_run_parameters=*parameter_list* – parameters that unified agent uses to start the dataserver. Unlike other settings, dataserver does not read the additional run parameters. They are read by the unified agent and passed to the dataserver command line. If you include the --instance parameter, the additional run parameters apply to the specified instance. Otherwise, the additional run parameters apply cluster-wide.
- --buildquorum[=force] – builds a new quorum device. Use =force to overwrite an exiting file or an existing quorum device on a raw partition. You must include the --cluster_input parameter with --buildquorum.
- --config_file=*config_file_name* – if used with -instance, sets this path to the Adaptive Server configuration file for the specified instance. If -instance is not included, sets the path to the cluster-wide configuration file.
- --diag={all | boot | toc | nodes | locks | config | cms} – for internal use only.
- --display={boot | nodes | heartbeat | master | cluster | instance | config | state} – displays the current state of cluster or instance:
  - boot – displays the startup information for the cluster, including the version of the quorum device, any trace flags issued at startup, the boot id of the cluster, and any messages displayed at startup.
  - nodes – displays the registered management nodes.
  - heartbeat – displays heartbeat information for all nodes in the cluster.

  - master – displays master device information.
  - cluster – displays the cluster configuration.
  - instance – displays the instance configuration. You must include --instance=*instance_name* with this parameter.
  - config – displays configuration for the cluster and for all instances in the cluster.
  - state – displays the current state for the cluster and for all instances in the cluster.
- --drop_cluster=[force] – drops a cluster and removes the quorum device. Use =force to force the drop if the quorum indicates the cluster is running.

  **Warning!** --drop_cluster removes the cluster.

- --drop_instance=*instance_name* – Sybase internal use only.

  **Warning!** Use the sybcluster utility to drop an instance from the cluster.

- --errorlog=*log_file_name* – full path to the error log for the specified instance. You must include the -instance_name parameter. Takes effect at next restart of the instance.
- --extract_config=*file_name* – extracts the configuration area of the quorum device to the specified file.
- -h | --help – qrmutil displays its full syntax.
- -F | cluster_input=*file_name* – loads the cluster configuration from the specified cluster input file.
- --fence_capable=*device_path* – tests if specified device can be fenced. Returns either “Device is fence capable” or “Device is not fence capable”.
- -instance=*instance_name* – applies qrmutil parameters to a specified instance.
- --interfaces_dir=*interfaces_path* – the path to a dirctory that contains a file named interfaces. If this parameter is used with --instance, it sets the path to the interfaces file for the specified instance. If --instance is not included, sets the path to the cluster-wide interfaces file.
- --max_instances=*number_of _instances* – sets the maximum number of instances for the cluster configuration.

- --master_dev=*master_device_name* – changes the master device the cluster uses.
- --primary_address=*inteconnect_address* – changes the primary interconnect address for a given instance.
- --primary_port=*port_number* – changes the starting port number for the primary interconnect for a given instance.
- --primary_protocol=*protocol* – changes the protocol used for the primary cluster interconnect.
- -Q | --quorum_dev=*quoum_path* – specifies the full path to the quorum device.
- --register_node=*node_name* – registers a node for quorum management.
- --secondary_address=*inteconnect_address* – changes the secondary interconnect address for a given instance.
- --secondary_port=*port_number* – changes the starting port number for the secondary interconnect for a given instance.
- --secondary_protocol=*protocol* – changes the protocol used for the secondary cluster interconnect.
- --traceflags=*trace_flag*, *trace_flag* – changes the cluster-wide or the instance-specific trace flags for startup. If you do not include a list of trace flags, qrmutil clears the trace flags for the cluster instance.
- --unregister_node=*node_name* – unregisters a node for quorum management.
- --verify_node=*node_name* – indicates that the specified node is registered on the quorum device.
- -v | --version – displays the version information for the qrmutil utility.

Examples

**Example 1** This example changes the path to the error log to */sybase/opt/cluster/ASE-15_0/ase1.log*:

```
qrmutil --quorum_dev=/dev/raw/raw101 --instance=ase1
--errorlog=/sybase/cluster/ASE-15_0/ASE-15_0/ase1.log
```

**Example 2** This example registers the node “blade5” for mycluster:

```
qrmutil --quorum_dev=/dev/raw/raw101 --register_node=blade5
```

**Example 3** This example creates a new quorum device for the cluster “mycluster”:

```
qrmutil --quorum_dev=/dev/raw/raw101 --cluster_input=/sybase/cluster/ase1.inp
```

```
-buildquorum
```

**Example 4** This example backs up the quorum device to */sybase/cluster_bak/quorum.bak*:

```
qrmutil --quorum_dev=/dev/raw/raw101
--extract_config=/sybase/cluster_bak/quorum.bak
```

**Example 5** This example restores the quorum device from the backup created in */sybase/cluster_bak/quorum.bak*:

```
qrmutil --quorum_dev=/dev/raw/raw101 --
cluster_input=/sybase/cluster_bak/quorum.bak --buildquorum=force
```

**Example 6** This example displays the cluster configuration stored on the quorum device:

```
qrmutil --quorum_dev=/dev/raw/raw101 --display=config
```

**Example 7** This example tests whether the named device can be fenced:

```
qrmutil --quorum_dev=/dev/raw/raw101 --fence_capable=/dev/raw/raw106
```

Usage

- qrmutil is primarily a diagnostic utility. Sybase recommends using sybcluster to make configuration changes to the cluster.
- You can pass up to 20 commands to qrmutil. However, you can specify the --instance= parameter only once.
- If you specify --buildquorum, the quorum is built and qrmutil exits without running any commands other than --cluster_input.
- qrmutil exits after it executes the --drop_cluster parameter.
- This is an example of using multiple commands:

```
qrmutil --quorum_dev=/dev/raw/raw101 --display=cluster
--register_node=blade1 --unregister_node=blade2 --verify_node=blade3
```

# Changes to utility programs

## dataserver

For the Cluster Edition, several parameters have been added to the dataserver utility for starting and configuring a cluster.

Syntax

This is the complete syntax for the dataserver command:

```
dataserver
-u, --admin_name=sa/sso_name
    --buildquorum=[force]
-a, --caps_file=filename
-F, --cluster_input=filename
    --cluster_takeover
-L, --conn_config_file=[filename]
    --create_cluster_id
-D, --default_db_size=size_spec
-e, --error_log=[filename]
-G, --event_log_server=logserv_name
-f, --forcebuild
-H, --ha_server
-h, --help=[{0|1|2|3}[,display_width]]
    --instance=instance_name
-y, --key_password=[key_password]
-K, --keytab_file=filename
-N, --license_prop_file=filename
-z, --logical_page_size=page_size
-Z, --master_db_size=size_spec
-d, --master_dev=master_device_name
-b, --master_dev_size=[size_spec]
-r, --master_mirror=filename
-m, --masterrecover
-g, --no_event_logging
-Q, --quorum_dev=quorum_dev
-q, --recover_quiesced
-w, --rewrite_db=database_name
-p, --sa_name=sa_name
-k, --server_principal=s_principal
-M, --shared_mem_dir=directory_name
-X, --sybmon
-T, --trace=trace_flag
-v, --version
```

Parameters

- --quorurm_dev=*path_to_quorum_device* – is the absolute path to the quorum device. It signals Adaptive Server to start as a shared-disk cluster. This is a required parameter for starting the Cluster Edition.

- --cluster_input=*input_file_name* – loads the cluster configuration from the specified input file into the quorum device. Required when you include --buildquorum. You can use this parameter without buildquorum, but only when no other instances are running..
- --instance=*instance_name* – name of the instance to start.
- --create_cluster_id – pairs the master device and the quorum device. This can be used to pair an existing quorum device with an existing master device and may be required in a disaster recovery situation. Use this parameter with caution because it overrides a safety feature that prevents the Cluster Edition from booting with the wrong combination of master and quorum devices.
- --buildquorum[=force] – creates a new quorum device. The force option overwrites an existing quorum device. This parameter requires that you include --cluster_input.
- --cluster_takeover – used to perform a cluster takeover when you set automatic cluster takeover to zero (0). A cluster takeover is necessary when an abrubt total cluster failure leaves stale state information on the quorum device.

See the *Utilities Guide* for a complete description of the dataserver parameters.

Examples

**Example 1** Create a new master and quorum device. The parameters in this example specify a 500 megabyte master device with a four kilobyte logical page size. The location of the master device, interface configuration file, interfaces path, and error log are specified in the cluster input file *mycluster.inp*. The name of the instance creating the quorum and master devices is "myase1":

```
dataserver --quorum_dev=/dev/raw/raw101 --buildquorum --
cluster_input=mycluster.inp --logical_page_size=4K --master_dev_size=500M --
instance=myase1
```

After the master database is built and the server is shutdown, you can restart the instance with this:

```
dataserver --quorum_dev=/dev/raw/raw101 --instance=myase1
```

**Example 2** Starts the instance myase2 as part of the cluster created in Example 1:

```
dataserver --quorum_dev=/dev/raw/raw101 --instance=myase2
```

**Example 3** Recreates the quorum device for mycluster without overwriting the master device (starting the instance "myase1"). The cluster must be shutdown before rebuilding the quorum.:

```
dataserver --quorum_dev=/dev/raw/raw101 --buildquorum --
cluster_input=mycluster.inp  --instance=myase1
```

Reload the cluster configuration from the *mycluster.inp* input file without recreating the quorum device or the master device (the cluster must be shutdown first):

```
dataserver --quorum_dev=/dev/raw/raw101 --cluster_input=mycluster.inp
--instance=myase1
```

Usage

- The Cluster Edition pairs the master device with the quorum device to prevent you from accidentally starting your master device in multiple clusters. You can use --create_cluster_id to establish a relationship between an existing master device and an existing quorum device.
- The dataserver command can read the following start options from the quorum device:

  ```
  -d / --master_dev
  -e / --error_log
  -c / --config_file
  -i / --interface_dir
  -T / --trace
  ```

- Options are honored in the following order:
  - Options that are passed to dataserver.
  - Options in the instance section of the cluster configuration file stored on the quorum device.
  - Options in the cluster section of the cluster configuration file stored on the quorum device.

  In many cases you can start an instance by providing only the --quorum_dev and --instance parameters. However, any traceflags passed to dataserver with the -T or --trace parameters are combined with any traceflags stored on the quorum device.

  dataserver does not honor the "additional run parameters" stored on the quorum device. These must be explicitly passed to dataserver (the values stored on the quorum device are read by the Unifed Agent and passed to dataserver when it is started by the agent).

# System tables

This section describes new and changed system tables for the Cluster Edition.

## timestamp columns

In Adaptive Server, if a table includes a timestamp column, its value is updated when a row is changed. Client applications can use this functionality to detect changes to rows using an access method called “optimistic locking.” The values in the timestamp column are unique in a database. However, in the Cluster Edition, timestamp column values are not guaranteed to be in increasing order in a database across tables, but they are guaranteed to be in increasing order for a particular table.

## Changed identity values

Identity columns in the Cluster Edition behave differently from those in nonclustered editions of Adaptive Server. Although the Cluster Edition guarantees that identity values are unique, for performance reasons the values may not monotonically increase.

In nonclustered Adaptive Server, a set of identity values are burned into memory to reduce disk I/Os as inserts access the next value from memory. In the Cluster Edition, the same size set is burned into memory, but the set is split among the cluster instances. In a two-instance cluster with an identity set size of 250000, the first instance inserts values {1,2,3, and so on}, and the second instance inserts values {125000,125001,125002, and so on}.

The next_identity function reports the next identity value for a table from the instance in which next_identity is executed. For example, next_identity returns 4 for instance 1 and 125003 for instance 2.

The behavior of the identity_burn_max remains the same as for a nonclustered Adaptive Server because the burn size and burn behavior is unchanged in the Cluster Edition.

# Changed system tables

## dbcc_types

The Cluster Edition adds two columns to the dbcc_types system table. The columns are:

***Table 13-1: New dbcc types***

| type_code | type_name | Description |
|---|---|---|
| 11 | enable dbcc_counter inserts | Enables (1) or disables (0) inserts in the dbcc_counters table. |
| 12 | enable excluded faults inserts | Enables (1) or disables (0) inserts of excluded faults. |

See also “sp_dbcc_updateconfig” on page 281.

## sysinstances

The Cluster Edition adds the sysinstances fake table, which reports on the state of the instances. sysinstances includes a row for each instance defined in the cluster configuration.

Although sysinstances is a fake table, it is not impacted by the setting of set system_view, and always returns a row for each instance, regardless of the system_view setting.

***Table 13-2: sysinstances fake table***

| Column name | Datatype | Description |
|---|---|---|
| id | tiny int | Id of the instance |
| name | varchar(30) | Name of the instance |
| state | char(17) | State of the instance (one of `online`, `offline`, `joining`, `leaving`, and `initiating`). |
| starttime | datetime | Date and time the instance started. |
| hostname | varchar(255) | Name of the operating system host running this instance. |
| connections_active | int | Number of active connections on the instance. |
| engines_online | smallint | Number of online engines for this instance. |

## nodeid renamed

The nodeid column has been renamed instanceid in these system tables:

- sysprocesses
- syslocks
- syscurconfigs
- sysmonitors
- sysengines
- syslisteners
- sysaudits_01 – sysaudits_08
- systransactions
- syssessions
- syscoordinations

## Columns added

***Table 13-3: Columns added to system tables***

| System table | Column name | Datatype | Description |
|---|---|---|---|
| sysconfigures | instanceid | tinyint | Indicates the id of the instance |
| sysdatabases | instanceid | tinyint | Indicates the id of the instance |
| sysprocesses | pad | smallint | Column added for alignment purposes |

| System table | Column name | Datatype | Description |
|---|---|---|---|
| sysprocesses | lcid | int | Id of the cluster |
| sysdevices | uuid | varbinary(16) | Reserved for future use |
| sysdevices | instanceid | tinyint | Indicates the id of the instance |
| sysservers | srvstat2 | unsigned int | Bitmap of server options |

### sysservers.srvnetname length change

The maximum length for the srvnetname column of sysservers changed from 32 characters to 255 characters.

### spid columns change from smallint to int

The datatypes for the spid column of these tables have been changed from smallint to int:

- sysprocesses.spid
- sysprocesses.fid
- sysprocesses.blocked
- sysprocesses.fid
- syslocks.spid
- syslocks.fid
- sysaudits_01.spid - sysaudits_08.spid
- syslogshold.spid
- systransactions.spid

Because of this change in spid datatype, Sybase strongly recommends that you archive and truncate audit tables before you upgrade. This reduces the likelihood of a failed upgrade because of insufficient space in the sybsecurity database.

## Controlling fake-table materialization

Certain stored procedures, such as sp_who and sp_lock, read from fake tables such as sysprocesses and syslocks. Because their rows are not stored on disk, fake tables present an exception to the shared-data nature of a shared-disk cluster, and special features apply.

You can control whether a fake-table query returns rows from the local instance or all instances in the cluster by using the set system_view command. set system_view is a session-level command. In addition, set system_view also controls MDA table materialization.

For information about setting the default system view at the logical-cluster level, see “System-view attribute” on page 87.

By default, Adaptive Server retrieves rows only from the local instance.

- To specify that fake-table queries materialize rows for all instances, use the cluster option. For example:

  ```
  set system_view cluster
  ```

- To specify that fake-table queries materialize rows for the local instance, use the instance option. For example:

  ```
  set system_view instance
  ```

To retrieve the current system_view setting, select the *@@system_view* global variable.

Adaptive Server supports cluster-wide materialization for these fake tables:

- sysprocesses
- syslocks
- sysengines
- syslisteners
- sysmonitors
- syssechmechs
- syscurconfigs

**Note** sysinstances is always set for cluster-wide materialization, regardless of the system_view setting.

# Monitor tables

See “Using Monitoring Tables in a Clustered Environment,”for a description of the changes to the monitoring tables for the Cluster Edition.

# Global variables

This section describes changes to global variables.

***Table 13-4: Global variables added for the Cluster Edition***

| Global variable | Definition |
|---|---|
| *@@instanceid* | Returns the id of the instance from which it was executed. |
| *@@active_instances* | Returns the number of active instances in the cluster. |
| *@@clusterboottime* | Returns the date and time the cluster was first started, even if the instance that originally started the cluster start has shut down. |
| *@@clustercoordid* | Returns the instance id of the current cluster coordinator. |
| *@@clustermode* | Returns the string: “shared-disk cluster”. |
| *@@clustername* | Returns the name of the cluster. |
| *@@instancename* | Returns the name of the instance from which it was executed. |
| *@@jsinstanceid* | Id of the instance on which the Job Scheduler is running, or will run once enabled. |
| *@@quorum_physname* | Returns the physical path for the quorum device. |
| *@@system_busy* | Number of ticks during which Adaptive Server was running a system task[1]. |
| *@@sys_tempdbid* | Returns the database id of the executing instance’s effective local system temporary database. |
| *@@system_view* | Returns the session-specific system view setting, either “instance” or “cluster”. |
| *@@user_busy* | Number of ticks during which Adaptive Server was running a user task[1]. |

[1]The value of *@@user_busy* + *@@system_busy* should equal the value of *@@cpu_busy*

Considerations for global variables in a clustered environment

For *@@servername*, the Cluster Edition returns the name of the cluster, not the instance name. Use *@@instancename* to return the name of the instance.

In a nonclustered Adaptive Server environment, the value for *@@identity* changes for every record inserted. If the most recent record inserted contains a column with the IDENTITY property, *@@identity* is set to the value of this column, otherwise it is set to "0" (an invalid value). This variable is session-specific, and takes its value based on the last insert that occurred during this session.

In a clustered environment, multiple nodes perform inserts on tables, so the session-specific behavior is not retained for *@@identity*. In a clustered environment, the value for *@@identity* depends on the last record inserted in the node for the current session and not on the last record inserted in the cluster.

# Functions

This section describes new and changed built-in functions.

## New functions

These are new functions for the Cluster Edition.

### spid_instance_id

Description — Returns the instance id on which the specified process id (spid) is running.

Syntax

```
spid_instance_id(spid_value)
```

Parameters — *spid_value* – the spid number whose instance id is requested.

Example — Returns the id of the instance that is running process id number 27:

```
select spid_instance_id(27)
```

Usage

- If you do not include a spid value, spid_instance_id returns NULL.
- If you enter an invalid or non-existing process id value, spid_instance_id returns NULL.

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can execute spid_instance_id.

### instance_id

Description — Returns the id of the named instance, or the instance from which it is issued if you do not provide a value for *name*.

Syntax

```
instance_id([name])
```

Parameters — *name* – is the name of the instance whose id you are researching.

Example — Returns the id of the local instance:

```
select instance_id()
```

Returns the id of the instance named "myserver1":

```
select instance_id(myserver1)
```

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can execute instance_id.

### instance_name

Description — Returns the name for the Adaptive Server whose id you provide, or the name of the Adaptive Server from which it is issued if you do not provide a value for *id*.

Syntax

```
instance_name([id])
```

Parameters — *id* – id of the Adaptive Server whose name you are researching.

Examples — Returns the name of the instance with an id of 12:

```
select instance_name(12)
```

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can execute instance_name.

### lc_name

Description — Returns the name of the logical cluster whose id you provide, or the current logical cluster if you do not provide an id.

Syntax

```
lc_name([lc_id])
```

Parameters — *lc_id* – id of the logical cluster.

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can execute lc_name.

## lc_id

Description — Returns the id of the logical cluster whose name you provide, or the current logical cluster if you do not provide a name.

Syntax

```
lc_id([lc_name])
```

Parameters — *lc_name* – the name of the logical cluster.

Examples

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can execute lc_id.

## workload_metric

Description — Queries the current workload metric for the instance you specify, or updates the metric for the instance you specify.

Syntax

```
workload_metric( instance_id | instance_name [, new_value ] )
```

Parameters — *instance_id* – the id of the instance.

*instance_name* – the name of the instance.

*new_value* – is a float value representing the new metric.

Examples — **Example 1** To see the user metric on the current instance:

```
select workload_metric()
```

**Example 2** To see the user metric on instance “ase2”:

```
select workload_metric("ase2")
```

**Example 3** To set the value of the user metric on “ase3” to 27.54:

```
select workload_metric("ase3", 27.54)
```

Usage

- A NULL value indicates the current instance.
- If a value is specified for *new_value*, the specified value becomes the current user metric. If a value is not specified for *new_value*, the current workload metric is returned.
- The value of *new_value* must be zero or greater.
- If a value is supplied for *new_value*, workload_metric returns that value if the operation is successful. Otherwise, workload_metric returns -1.

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — The sa_role or ha_role is required to execute workload_metric.

### sys_tempdbid

Description — Returns the id of the effective local system temporary database of the specified instance. Returns the id of the effective local system temporary database of the current instance when *instance_id* is not specified.

Syntax

```
sys_tempdbid(instance_id)
```

Examples — Returns the effective local system temporary database id for the instance with an instance id of 3:

```
select sys_tempdbid(3)
```

Usage — If you do not specify an instance id, sys_tempdbid returns the id of the effective local system temporary database for the current instance.

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can run sys_tempdbid.

### sdc_intempdbconfig

Description — Returns 1 if the system is currently in temporary database configuration mode; if not, returns 0.

Syntax

```
sdc_intempdbconfig()
```

Example

```
select sdc_intempdbconfig()
```

Standards — ANSI SQL – Compliance level: Transact-SQL extension.

Permissions — Any user can run sdc_intempdbconfig.

### db_instanceid

Description — Returns the id of the owning instance of a specified local temporary database. Returns NULL if the specified database is a global temporary database or a nontemporary database.

Syntax

```
db_instanceid(dbid)
db_instanceid(dbname)
```

Usage

- Access to a local temporary database is allowed only from the owning instance. db_instanceid determines whether the specified database is a local temporary database, and the owning instance for the local temporary database.You can then connect to the owning instance and access its local temporary database.
- You cannot run db_instanceid with no argument.

## db_recovery_status

Description

Returns the recovery status of the specified database. Returns the recovery status of the current database if you do not include a value for *database_ID* or *database_name*.

syntax

db_recovery_status([*database_ID* | *database_name*])

Parameters

- *database_ID* – the id of the database whose recovery status you are requesting.
- *database_name* – the name of the database whose recovery status you are requesting.

Examples

This returns the recovery status of the current database:

```
select db_recovery_status()
```

This returns the recovery status of the database with named test:

```
select db_recovery_status("test")
```

This returns the recovery status of a database with a database id of 8:

```
select db_recovery_status(8)
```

Usage

A return value of 0 indicates the database is not in node-failover recovery. A return value of 1 indicates the database is in node-failover recovery.

Permissions

Any user can execute db_recovery_status.

CHAPTER 14

# The sybcluster Utility

This chapter describes the sybcluster utility, which you can use to create, start, manage the cluster, and manage instances in the cluster.

## sybcluster

Description

Manages a Sybase shared-disk cluster. sybcluster lets you create, start, stop, and manage a cluster or any instance in a cluster.

Syntax

sybcluster

[ -C *cluster_name* ]
[ -d *discovery_list* ]
[ -F *agent_connection* ]
[ -h ]
[ -I *instance_name* ]
[ -i *input_file_path* ]
[ -L ]
[ -m *message_level* ]
[ -P [ *password* ]]
[ -U *user_name* ] (the default value is “uafadmin”)
[ -v ]

**Starting sybcluster** The recommended method for starting sybcluster and connecting to a cluster is:

```
sybcluster -U login_name -P password -C cluster_name
-F agent_spec
```

**Example 1** Starts sybcluster using direct connect and default port numbers.

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

**Example 2** Starts sybcluster using direct connect and port numbers.

```
sybcluster -U uafadmin -P -C mycluster
-F "blade1:9100,blade2:9292,blade3:9393"
```

**Example 3** You can also start sybcluster using discovery. See "-d discovery_list" on page 318 for more information.

```
sybcluster -U uafadmin -P -C mycluster
-d "JINI(myjiniserver:4564)"
```

The -C *cluster_name*, -P *password*, -I *instance_name*, -F *agent_connection*, and -d *discovery_list* parameters are default values that can be changed using subsequent sybcluster interactive commands. If you do not specify these values on the sybcluster command line, sybcluster prompts for them as they are required.

You can also start sybcluster and then use the interactive connect command to connect to the cluster. For example:

```
sybcluster
> connect to mycluster login uafadmin password " "
agent "blade1,blade2,blade3"
```

---

**Note** See "sybcluster interactive commands" on page 65 for syntax and usage descriptions of the sybcluster interactive commands.

---

Parameters

-C *cluster_name*
is the unique name of the Sybase shared-disk cluster to be managed. sybcluster looks up the name in the cluster directory or uses agent discovery services.

-d *discovery_list*
specifies the discovery services to be used to discover a shared-disk cluster agent and the discovery order. Discovery services supported for the Cluster Edition are listed in Table 14-1. The format is:

```
"method[(method_specification][,...)"]]
```

For example:

```
-d "udp(),jini(jinihost1;jinihost2)"
```

***Table 14-1: Discovery methods***

| Discovery method | Description |
|---|---|
| UDP() | Performs a UDP broadcast and listens for a response from listening Unified Agents. UDP discovery does not cross subnet boundaries. |
| JINI(*JINI_spec*) | Specifies the JINI servers used to look up the locations of nodes in the cluster. The specification form is: *host-name*[:*port_num*].<br>Indicate multiple JINI servers by placing a semicolon between each specification. By default, sybcluster uses port number 4160 to attach to a JINI server.<br>The JINI server must be running, and the management agents (UAF) must be registered with the JINI server. The locations of the nodes, and status of the instances are stored on the JINI server. |
| LDAP(*LDAP_spec*) | Specifies an LDAP server that will be used to look up the locations of the nodes in the cluster. The specification form is: *host_name*[:*port_num*][?*registry*].<br>Indicate multiple LDAP servers by placing a semicolon between each specification. By default, sybcluster uses port number 389 to attach to an LDAP server and the LDAP directory at "cn=ua-registry,ou=ua,dc=sybase,dc=com". |

-F *agent_connection*

specifies the agent to be used to access the cluster. The format is:

```
host_name[:port_num] [, host_name[:port_num ]]
```

For example:

```
-F "node1,node2,node3,node4:9999"
```

The default port number is 9999.

-h

displays sybcluster syntax and lists supported interactive commands.

-I *instance_name*

specifies the instance to be accessed. If you do not specify the -I option when you execute sybcluster, you may need to specify it when entering certain interactive commands. sybcluster uses this name to discover the remote host, and as a default when executing interactive commands. If an interactive command affects multiple instances, the instance identified by -I, if available, is used as the priority connection.

To override the instance specified by -I, execute the use command in interactive mode.

-i

specifies an operating system file for input to sybcluster. This file contains sybcluster commands, one command per line. The final command in the file should be quit.

-L

creates a *sybcluster.log* file. sybcluster writes all messages to this file irrespective of the message level set by the -m option.

-m *message_level*

specifies which sybcluster and unified agent messages are displayed on the client console. Message levels are:

- 0 – off (no messages to log file or console)
- 1 – fatal
- 2 – error
- 3 – warning
- 4 – information
- 5 – debug

sybcluster displays all messages of the level you choose and all messages of greater severity (with lower numbers). That is, if you select message level 3, sybcluster displays messages of level 3, 2, and 1. The default level is 4.

-P [*password*]
is the management agent password for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is "uafadmin" with no password. This is the Simple Login Module in the Agent configuration. The user and password can be configured to use several different mechanisms for authentication and authorization, including using the running instance and the operating system logins.

If you do not specify the -P option, sybcluster prompts for a password. For a blank or null password, use the -P option without a value or enter a set of quotation marks without content.

You can encrypt the password using the Sybase passencrypt utility. See "Setting the user name and password" on page 229.

-U *user_name*
is the management agent login name. The default login after installation is "uafadmin." See the -P description for more information.

-v
displays the sybcluster version number and other information.

# sybcluster interactive commands

This section describes the sybcluster interactive commands. Some commands are active before you connect to a cluster (see "Commands active before connecting to a cluster" on page 321); others are active only after you connect to a cluster (see "Commands active after connecting to a cluster" on page 322).

The sybcluster command prompt includes the current cluster and the default instance when these values have been set. The prompt is:

- > – when sybcluster is not connected to a cluster.
- *cluster_name*> – when sybcluster is connected to a cluster.
- *cluster_name instance_name*> – when sybcluster is connected to a cluster and a default instance has been set.

## Commands active before connecting to a cluster

These commands are active before you connect to a cluster. They are not available after you connect to a cluster.

Connect
[ to *cluster_name* ]
[ Login *login_name* ]
[ Password [*password* ]]
[ Agent *agent_spec* ]
[ Discovery *discovery_spec* ]

```
Create Cluster [cluster_name ]
    [ Login login_name ]
    [ Password password ]
    [ Agent agent_spec ]
    [ Discovery discovery_spec ]
    [ File input_file ]

Deploy Plugin
        [ Login login_name ]
        [ Password password ]
        [ Agent agent_spec ]
        [ Discovery discovery_spec ]

Exit

Help

Quit

Show agents
      [ login login_name ]
      [ password password ]
      [ agent "agent_spec[, agent_spec[,...]]" ]
      [ discovery "discovery_spec[, discovery_spec[,...]]" ]

Upgrade server server_name
      [ login login_name ]
      [ password password ]
      [ agent agent_spec ]
      [ discovery discovery_spec ]
      [ file input_file_name ]
      [ checkonly ]
```

Commands active after connecting to a cluster

These commands are active only after you connect to a cluster:

```
Add
      Instance instance_name
            file file_name
      Backupserver

Create
      Backupserver
      Monitorserver
      Xpserver

Disconnect

Diagnose
      Cluster
      Instance instance_name

Drop
```

```
        Backupserver
        Cluster
        Instance instance_name
        Monitorserver
        Xpserver

    Localize

    Set
        Cluster
            MaxInst max_num_instances
            Login [ login_name ] [ Password password ]
            TraceFlags trace_flag
            Primary Protocol protocol
            Secondary Protocol protocol
        Instance [ instance_name ]
            LogPath log_file_path
            Primary
                Address ip_address
                Port port_range_start port_range_end
            Secondary
                Address ip_address
                Port port_range_start port_range_end
            StartArgs startup_arguments
        BackupServer
        MonitorServer
        XPServer
            Port

    Show
        Cluster
            Config
                Template
            Log
                [ Errors]
                [ MinSeverity severity_level |
                [ StartDate [date_string ]]
                [ EndDate [date_string ]]
                [ Last num_of_lines ]
            Status

        Instance [instance_name]
            Config
            Log
                [ Errors ]
                [ MinSeverity severity_level ]
                [ StartDate [date_string ]]
                [ EndDate [date_string ]]
                [ Last num_of_lines ]
            Status
        BackupServer
```

Config
MonitorServer
Config
Session
XPServer
Config

Shutdown
Cluster
Instance [ *instance_name* ]

Start
Cluster
Instance *instance_name*
[ Unlock ]

Use [ *instance_name* ]

Help
Add
Diagnose
Disconnect
Drop
Set
Show
Shutdown
Start

# add backupserver

Description — Configures Backup Server for nodes not already configured for Backup Server.

Syntax — add backupserver

Examples — Adds a Backup Server to “mycluster” on nodes “blade3” and “blade4”.

```
add backupserver

Finding nodes for which Backup Server is not configured...
Do you want to configure Backup Server for node "blade3"? [Y]
Please enter the Backup Server port number for node "blade3": 5001
Do you want to configure Backup Server for node "blade4"? [Y]
Please enter the Backup Server port number for node "blade3": 50011
```

Usage

- You can configure Backup Server for one or more nodes in the cluster.
- add backupserver lets you add additional Backup Servers at any time.

# add instance

Description

Adds one new instance to the cluster. The instance can be added interactively, with sybcluster prompting for necessary configuration information, or through an input file. add instance also creates a local system temporary database for the new instance.

Syntax

add instance *instance_name* [ file "*input_file*" ]

Parameters

*instance_name*
is the name of the instance.

file "*input_file*"
specifies a file name that contains the configuration information for adding an instance.

Usage

- add instance creates a local system temporary database for the new instance. Before executing add instance, make sure that a device with sufficient space for the local system database exists.
- The input file for add instance has the same format as the cluster input file. However, the add instance input file may limit the instance definitions to the new instance in the node section.
- add instance may prompt for this information:
  - The instance name, if you did not enter an instance name in the command statement.
  - The node hosting the instance
  - The port number of the UAF agent on the node
  - The query port number
  - The primary and secondary address of the node
  - The primary and secondary port specification
- If Backup Server is configured for the cluster, add instance asks if Backup Server is to be configured for the new instance and, if yes, prompts for the port number for the node. add instance behaves in a similar manner for Monitor Server and XP Server.

# connect

Description

Connects to an existing cluster.

Syntax

connect [ to *cluster_name* ]
[ login *login_name* ]
[ password [*password* ]]
[ agent "*agent_spec* [, *agent_spec* [,...]]" ]
[ discovery " *discovery_spec* [, *discovery_spec* [, ...]]" ]

Parameters

*cluster_name*
is the name of the cluster to which you are connecting.

login *login_name*
is the management agent login for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is "uafadmin" with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

For information about Sybase Common Security, see *Unified Agent and Agent Management Console Version 2.0 for Windows and UNIX*.

password *password*
is the management agent password for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is "uafadmin" with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

agent *agent_spec*
is the agent specification that identifies the nodes in the cluster running a Unified Agent, and the port number that sybcluster uses to connect to the Unified Agent. The format is *node_name*:*port_number*[, *node_name*:*port_number* ][,...]. The default port number is 9999.

This is the preferred method for connecting to a cluster.

discovery *discovery_spec*
is the discovery method used to identify the agents responsible for the requested cluster. The format is method[(*method_specification*)][, ( *method_specification* )[,...]]. See Table 14-1 on page 319 for more information.

Examples

**Example 1** Connects to “mycluster”, when “mycluster” is the default cluster specified in the sybcluster command statement.

```
connect
```

**Example 2** In this example, you connect to “mycluster” using the agent specification and default port numbers.

```
connect to mycluster agent "blade1,blade2,blade3"
```

Usage

- A direct connection is one in which the user identifies the cluster nodes and, optionally, the port numbers for the UAF agents. Sample agent specifications are:
  - myhost – identifies the host node and assumes the default listening port of 9999.
  - myhost.mydomain.com – includes the host domain name.
  - myhost:9999 – identifies the host node and listening port number.

# create backupserver

Description

Creates a Backup Server for the cluster.

Syntax

create backupserver

Examples

Creates the Backup Server “mycluster_BS” for “mycluster”:

```
create backupserver
```

```
Enter the Backup Server name: [mycluster_BS]
Enter the Backup Server log file path: [$SYBASE/ASE-15_0/
    install/mycluster_BS.log]
Do you want to create a Backup Server for node "blade1"? [Y]
Enter the Backup Server port number for node "blade1":
The Backup Server "mycluster_BS" was successfully defined.
```

Usage

- create backupserver prompts for the Backup Server listening port on each node. It copies other necessary configuration information from the cluster configuration file. create backupserver:
  - Creates directory service entries for Backup Server on each node.
  - Creates the Backup Server configuration and log files, and the RUN_<*backup_server*> script.

- Adds the Backup Server name to the cluster's *sysservers* table.
- Enables Backup Server HA.

# create monitorserver

Description — Creates a Monitor Server for each instance in the cluster.

**Note** You must run the installmon isql script, located in *$SYBASE.$SYBASE_ASE/scripts*, before running Monitor Server.

Syntax — create monitorserver

Examples — Creates a Monitor Server for each instance in "mycluster".

```
create monitorserver
```

```
Enter the Monitor Server log file location for instance "ase1".[$SYBASE/
   ASE-15_0/install/mycluster_MS1.log]
Enter the Monitor Server port number for instance "ase1":
Enter the Monitor Server log file location for instance "ase2".[$SYBASE/
   ASE-15_0/install/mycluster_MS2.log]
Enter the Monitor Server port number for instance "ase2":
Enter the Monitor Server log file location for instance "ase3".[$SYBASE/
   ASE-15_0/install/mycluster_MS3.log]
Enter the Monitor Server port number for instance "ase3":
Enter a username: sa
Enter a password:
WARN – mycluster:AseCfg:411:The SQL script "INSTALLMON" must be run before
the Monitor Server can be run.
The Monitor Server has been successfully configured.
```

Usage — create monitorserver prompts for the listening port number for each instance in the cluster, and a user name and password.

# create xpserver

Description — Creates an XP Server for each instance in the cluster.

Syntax — create xpserver

Examples — Creates an XP Server for each instance in "mycluster".

```
create xpserver

Enter the XP Server port number for instance "ase1":
Enter the XP Server port number for instance "ase2":
Enter the XP Server port number for instance "ase3":
The XP Server was successfully defined for each instance.
```

Usage — create xpserver prompts for the XP Server listening port for each node in the cluster. Other information necessary to create the XP Server is read from the cluster configuration file.

# create cluster

Description — Creates an Adaptive Server shared-disk cluster. Enter the necessary configuration information interactively, as responses to a series of prompts, or use an input file.

Syntax — create cluster [*cluster_name* ]
[ login *login_name* ]
[ password *password* ]
[ agent "*agent_spec* [, *agent_spec* [, ...]]" ]
[ discovery " *discovery_spec* [, *discovery_spec* [, ...]]" ]
[ file "*input_file*" ]

Parameters — *cluster_name*
is the name of the cluster.

login *login_name*
is the management agent login for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is "uafadmin" with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

password *password*
is the management agent password for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

agent *agent_spec*
is the agent specification that identifies the nodes in the cluster running a Unified Agent, and the port number that sybcluster uses to connect to the Unified Agent. The format is *node_name*:*port_number* [, *node_name*:*port_number* ] [,...]. The default port number is “9999.”

discovery *discovery_spec*
is the discovery method used to identify the agents responsible for the requested cluster. The format is method[(*method_specification*)] [, ( *method_specification* ) [,...]]. See Table 14-1 on page 319 for more information.

file “*input_file*”
is the operating system input file for creating the cluster.

Examples

**Example 1** Creates a new cluster called “mycluster”; sybcluster prompts you for the information necessary to create the cluster.

```
create cluster mycluster
```

**Example 2** Creates a new cluster called “mycluster1” using configuration information supplied in the mycluster1.xml file.

```
create cluster mycluster1 file mycluster1.xml
```

Usage

- When you create a cluster, sybcluster prompts for:
  - The cluster name, if one has not been provided
  - The number of instances
  - The complete path to the master, quorum, systemdb, sybsysprocs, and temporary database devices
  - The path to the directory containing the interfaces file
  - Trace flags (optional)
  - The complete path to the dataserver configuration file
  - The primary and secondary interconnection protocols

- The instance host name, port number, private address, log file location, and startup arguments

- After you create and confirm the cluster, create cluster prompts for an I/O fencing check, which confirms whether or not the device has I/O fencing capability.

# deploy plugin

Description

Adds the configuration information for a single instance of the cluster to the Unified Agent. Can be used to configure the Unified Agent to manage a cluster if you created the cluster without using the Adaptive Server plug-in or sybcluster utility, or if you need to recreate the Unified Agent configuration for a cluster. The configuration of a cluster instance is performed by deploying a Unified Agent plug-in.

Syntax

deploy plugin
[ login *login_name* ]
[ password *password* ]
[ agent *agent_spec* ]
[ discovery *discovery_spec* ]

Parameters

login *login_name*

is the management agent login for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

password *password*

is the management agent password for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

agent *agent_spec*

is the agent specification that identifies the nodes in the cluster running a Unified Agent, and the port number that sybcluster uses to connect to the Unified Agent. The format is “*node_name:port_number* [, *node_name*:*port_number* ] [,...]]”. The default port number is “9999”.

discovery *discovery_spec*
is the discovery method used to identify the agents responsible for the requested cluster. The format is "*method*[(*method_specification*)] [, ( *method_specification* ) [,...]]". See Table 14-1 for more information about discovery methods.

Examples

**Example 1** Deploys the plug-in using the uaf agent on host "system1501".

```
deploy plugin agent system1501
```

sybcluster prompts for the cluster name, cluster node number, full path to the quorum device, the environment shell script path, and the Adaptive Server home directory.

**Example 2** Deploys the plug-in and uses discovery to identify the agent:

```
deploy plugin discovery udp
```

Usage

- After you execute deploy plugin, sybcluster prompts you for:
  - The path to the quorum device.
  - The path to the Sybase home directory.
  - The location of your Sybase environment script. This must be a shell script that can be loaded using the "*.file_name*" syntax such as "sh" or "bash". An example is *SYBASE.sh*.
  - The location of your Adaptive Server software directory. The default is *<sybase_home_directory>/ASE-15_0*. When entering the location of the Adaptive Server software directory, include the full path. Do not use $SYBASE.
- The dataserver login and password are configured using the login command, which updates all Adaptive Server plug-ins managing the cluster.

# diagnose cluster

Description

Performs a set of checks to ensure that the cluster is working correctly.

Syntax

diagnose cluster

Examples

Checks that “mycluster” is working correctly.

```
diagnose cluster

Cluster name..................... mycluster
Maximum instances................ 4
Cluster node count............... 1
Instances defined................ 4
Is cluster locked................ Yes
JDBC connection available....... 1 ase1 Yes
JDBC connection available....... 2 ase2 Yes
JDBC connection available....... 3 ase3 Yes
JDBC connection available....... 4 ase4 Yes
Instance Public Network......... 1 ase1 on blade1
(10.22.79.39) Reachable: Yes
Instance Public Network......... 2 ase2 on blade1
(10.22.79.39) Reachable: Yes
Instance Public Network......... 3 ase3 on blade1
(10.22.79.39) Reachable: Yes
Instance Public Network......... 4 ase4 on blade1
(10.22.79.39) Reachable: Yes
Has private Primary network..... No
Has private Secondary network... No
Network ports required/instance. 20
Minimum port allowed............ 1025
Maximum port allowed............ 65535

Current port strategy........... Public primary and
secondary unique.
...The ports are sequenced primary followed by the next
instance primary.
...When the primaries are are completed the secondary
ports follow the same pattern.

Recommended port strategy....... Public primary and
secondary unique.
...The ports are sequenced primary followed by the next
instance primary.
...When the primaries are are completed the secondary
ports follow the same pattern.
```

Usage

- diagnose cluster checks that:
  - A Unified Agent is running on each instance in the cluster.
  - The number of instances in the cluster does not exceed the value set for maximum number of instances.
  - The quorum file exists.

- All instances are defined in the interfaces file and that port numbers do not conflict.
- The primary and secondary protocol specifications do not overlap.
- Each of the *$SYBASE* directories are shared.

# diagnose instance

Description — Performs a set of checks to ensure that the instance is configured correctly.

Syntax — diagnose instance [*instance_name*]

Parameters — *instance_name*
is the name of an instance. If no instance name is entered, sybcluster uses the default value.

Examples — Displays and verifies configuration information for “ase1” on “mycluster”.

```
diagnose instance ase1

Cluster name ............mycluster
Instance id ..............1
Instance name ..............  ase1
Node name ..................  blade1
Query port ...................7101

JDBC connection available .....Yes

Instance Public Network ........1 ase1 on blade1
(10.33.108.139) Reachable: Yes

Minimum port allowed.........1025
Maximum port allowed ..........65535

Instance port range ........1 Primary ase1 17100 to
17115 (16) Okay
Instance port range ........  1 Secondary ase1 17165 to
17180 (16) Okay
```

Usage — Use diagnose cluster to ensure the cluster is configured correctly.

# disconnect

| | |
|---|---|
| Description | Closes all connections to the current cluster and returns sybcluster to an unconnected state. |
| Syntax | disconnect |
| Usage | Use connect to reconnect to an existing cluster. |

# drop backupserver

| | |
|---|---|
| Description | Drops Backup Server from a node or from the cluster. |
| Syntax | drop backupserver |
| Examples | **Example 1** Drops a single Backup Server. |

```
drop backupserver

Do you want to drop the Backup Server from:
   1. Selected nodes
   2. Cluster
Enter choice: 1
Do you want to drop Backup Server from node "blade1"? [N] y
Do you want to drop Backup Server from node "blade2"? [N]
The Backup Server has been dropped from selected nodes.
```

**Example 2** Drops the Backup Server from the cluster.

```
drop backupserver

Do you want to drop the Backup Server from:
   1. Selected nodes
   2. Cluster
Enter choice: 2
Are you sure you want to drop Backup Server mycluster_BS from cluster
mycluster? (Y or N): [N] y
The Backup Server has been dropped.
```

| | |
|---|---|
| Usage | Use create backupserver to create a Backup Server for the cluster. |

# drop cluster

Description — Removes each instance from a cluster and then removes the cluster definition from the cluster configuration file. Also, removes regular files associated with the cluster and the cluster agent plug-ins that manage the cluster. The cluster must be Down to use drop cluster.

Syntax — drop cluster

Examples — Drops all instances from the current cluster and deletes the cluster.

```
drop cluster
```

Usage

- sybcluster prompts for confirmation before dropping the cluster.
- drop cluster:
  - Removes cluster and instance entries from the interfaces file, configuration files, and specified data devices.
  - Marks the quorum device as unused.
  - Shuts down and removes the cluster's UAF agent plug-ins.
- Due to certain file system locking, the UAF plug-ins may not be deleted after you use drop cluster. Verify that the *$SYBASE_UA/nodes/*/plugins/<cluster_name>* directory has been deleted. If the directory still exists, delete it.

# drop instance

Description — Removes an instance from the cluster configuration file and updates the Unified Agent Framework (UAF) and discovery services. Also, notifies the cluster that an instance is to be dropped, and removes the instance and interfaces file entries.

Syntax — drop instance [ *instance_name* ]

Parameters — *instance_name*
identifies an instance in a cluster. If an instance name is not specified, sybcluster uses the default specified in the sybcluster command line.

Examples — Removes the "ase3" instance from the current cluster.

```
drop instance ase3
```

Usage

- Before you use drop instance:
  - Start at least one instance in the cluster other than the instance to be dropped.
  - Shut down the instance to be dropped.
  - Manually remove instance-specific information. drop instance automatically removes the local system temporary database.
- sybcluster prompts for confirmation before removing the instance.
- You cannot drop the last instance in the cluster. You must use drop cluster.
- drop instance removes references to the instance in the interfaces file, the instance entry in the quorum device, and notifies the cluster that the instance has been dropped.
- drop instance drops Monitor Server and XP Server if they have been configured for that instance.

## drop monitorserver

Description Drops the Monitor Server for every instance in the cluster.

Syntax drop monitorserver

Examples Drops all Monitor Servers defined for “mycluster”.

```
drop monitorserver

Are you sure you want to drop the Monitor Server
configurations from the cluster mycluster? (Y or N):
    [N] y
The Monitor Servers have been dropped for all instances.
```

Usage Use create monitorserver to create a Monitor Server for the cluster.

## drop xpserver

Description Drops the XP Server for each instance in the cluster.

Syntax drop xpserver

Examples — Drops the XP Servers for “mycluster”.

```
drop xpserver

Are you sure you want to drop the XP Servers from cluster mycluster"? {Y or
N): [N] y
The XP Servers have been dropped for all instances.
```

Usage — Use create xpserver to create an XP Server for the cluster.

# exit

Description — Exits the sybcluster utility.

Syntax — exit

Usage

- exit and quit perform the same task: they exit the sybcluster utility.
- If some agents have been shut down while connected to sybcluster, error messages describing the connections may display. You can ignore these messages.

# help

Description — Lists the currently available sybcluster interactive commands.

Syntax — help

Usage — The list of currently available interactive commands changes depending on whether or not sybcluster is connected to a cluster.

# localize

Description — Displays the current values for default language, charset, and sort order. Allows modification of default values, and addition or removal of languages.

Syntax — localize

Examples

This example displays default localization values, and then prompts for changes. The default character set changes to Chinese, the default charset to eucgb, and the default sort order to bin_eucgb.

```
localize

Current default locale properties are:
Default Language - portuguese
Default Charset - mac
Default SortOrder - Binary ordering, for use with the
Macintosh charcter set(mac).

Options for default Language are:
1. spanish
2. portuguese
3. german
4. us_english
5. thai
6. french
7. japanese
8. chinese
9. korean
10. polish
Enter the number representing the language to be set as
defaults: [2] 8

Options for default charsets are:
1. gb18030
2. eucgb
3. uttf8
Enter the number representing the charset to be set as
default: [1] 2

Options for sort orders are:
1. Binary ordering, for the EUC GB2312-80 character set
(eucgb).
Enter the number representing the sort order to be set
as default [1]

Do you want to install any language? [Y] n
Do you want to remove any language? [N ]
The cluster mycluster was successfully localized with
default language chinese, charset eucgb, sortorder
bin_eucgb
```

Usage

- The current default localization value displays after each prompt. To accept the current value, enter a carriage return instead of a number.
- The options for default languages include all languages present in *$SYBASE_ASE*. If the selected default language is not configured, use localize to configure it or remove it.
- To ensure that new values are consistent for all instances in the cluster, restart the cluster after changing localization values.

# quit

Description — Exits the sybcluster utility.

Syntax — quit

Usage — exit and quit both exit the sybcluster utility.

# set backupserver

Description — Changes the listening port number for Backup Server on specified nodes in a cluster.

Syntax — set backupserver

Examples — Changes the listening port number for Backup Server on “blade1” of “mycluster”.

```
    set backupserver

Backup Server is configured on the following nodes:
   1. blade1: 3001
   2. blade2: 3002
   3. blade3: 3003
Do you want to change the Backup Server port on any node? {Y}
Enter the number representing the node whose port you want to change: 1
Enter the Backup Server port number for node "blade1":4001
Backup Server was successfully modified as per new properties.
```

Usage — When you set a new listening port number, Adaptive Server first checks to see if that port number is already in use.

# set cluster

Description

Changes configuration values for the cluster. The cluster must be down to execute all set cluster commands except set cluster login.

Syntax

```
set cluster {
    maxinst max_num_instances |
    traceflags trace_flag[, trace_flag[,...]] |
    { primary | secondary } protocol udp |
    login login_name [ password password ] }
```

Parameters

maxinst *max_instances*
specifies the maximum number of instances that can run in the cluster.

traceflags *trace_flag*[, *trace_flag*[,...]
specifies trace flags to be set when the cluster starts.

login *login_name* [ password *password* ]
specifies a user name and password that the Unified Agent uses to log in to the cluster and perform shutdown and certain other tasks. This login must have sa_role. By default, the Unified Agent uses the "sa" login with no password. To change this password, use set cluster login. See also, "Changing user names or passwords" on page 236.

---

**Note** set cluster login can only be used to change the login or password that the Unified Agent uses to log in to the cluster. To change the login or password sybcluster uses to log in to the Unified Agent, use the Agent Management Console Sybase Central Plug-in.

---

{ primary | secondary } protocol udp
sets the protocol for the private network for the primary or secondary interface.

Examples

**Example 1** Changes the maximum number of instances to 4 for "mycluster".

```
set cluster maxinst 4
```

**Example 2** Adds the trace flag 15506.

```
set cluster traceflags 15506
```

**Example 3** Changes the password for the "sa" user name.

```
set cluster login sa password abcde
```

Usage

To check that the cluster is Down, enter show cluster status.

# set instance

Description — Sets properties of the instance. The instance must be Down.

Syntax

set instance *instance_name* logpath *path*
set instance *instance_name* startargs *values*
set instance *instance_name* {primary | secondary} port *port_range*
set instance *instance_name* {primary | secondary} address *ip_address*

Parameters

logpath *logfile_path*
specifies the path for the instance log file.

*instance_name*
specifies an instance.

startargs *startup_args*
specifies arguments for starting the instance.

{ primary | secondary } address *ip_address*
specifies the primary or secondary IP address for the instance.

{ primary | secondary } port *port_range*
specifies the primary or secondary port range for the instance. The format for *port_range* is : *start_num end_num*.

Examples — Changes the port range for the primary interface listening port.

```
set instance primary port 7777
```

Usage — To check that the instance is Down, enter show cluster status.

# set monitorserver

Description — Changes the Monitor Server listening port numbers for specified instances in the cluster.

Syntax — set monitorserver

Examples — Changes the Monitor Server listening port number for instance “ase2” on “blade2” of “mycluster” without changing the listening port numbers for other instances in the cluster.

```
set monitorserver
```

```
Enter the Monitor Server Port number for instance "blade1": [5001] <CR>
Enter the Monitor Server port number for instance "blade2": [5002] 5011
Enter the Monitor Server port number for instance "blade3": [5003] <CR>
```

Usage

- set monitorserver prompts you to change the listening port number for each Monitor Server configured in the cluster. You can accept the current value, provided in square brackets, or enter a new port number.
- You can change the Monitor Server listening port number for one or more instances.

## set xpserver port

Description — Changes the listening port number for XP Server on specified nodes of the cluster.

Syntax — set xpserver port

Examples — Changes the listening port for the XP Server for instance “ase1” on “blade1” of “mycluster” without changing the listening ports for “ase2” and “ase3”.

```
set xpserver port

Enter the XP Server port number for instance "ase1" [3002]: 4002
Enter the XP Server port number for instance "ase2" [3002]: <CR>
Enter the XP Server port number for instance "ase3" [3002]: <CR>
```

Usage — You can change the XP Server listening port number on one or more instances.

## show agents

Description — Displays information about available UAF agents.

Syntax — Show agents
[ login *login_name* ]
[ password *password* ]
[ agent “*agent_spec*[, *agent_spec*[,...]]” ]
[ discovery “*discovery_spec*[, *discovery_spec*[,...]]” ]

Parameters

login *login_name*
is the management agent login for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

password *password*
is the management agent password for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

agent *agent_spec*
is the agent specification that identifies the nodes in the cluster running a Unified Agent, and the port number that sybcluster uses to connect to the Unified Agent. The format is “*node_name:port_number* [, *node_name*:*port_number* ] [,...]]”. The default port number is “9999.”

discovery *discovery_spec*
is the discovery method used to identify the agents responsible for the requested cluster. The format is “*method*[(*method_specification*)] [, ( *method_specification* ) [,...]]”. See Table 14-1 for more information about discovery methods.

Examples

Displays UAF agent information:

```
show agents

Agent Information:
service:jmx:rmi:///jndi/rmi://blade1:9985/agent-------
---------------------------------------------

Node Name:       blade1
Agent Port:      9985
Agent Version:   2.5.0
Agent Build:     977

OS Name:         Linux
OS Version:      2.6.9-42.ELsmp
OS Architecture: amd64

Agent Service Info:

Agent Service (Agent)  Build: 977  Status: running
BootstrapService (BootstrapService)  Build:
```

```
<unavailable>  Status: running
Configuration Service (ConfigService)  Build: 977
Status: running
Deployment Service (DeploymentService)  Build:
<unavailable>  Status: running
Environment Service (EnvironmentDiscoveryService)
Build: 977  Status: running
File Transfer Service (FileTransferService)  Build: 977
Status: running
Plugin Registration Service (PluginRegisterService)
Build: 977  Status: running
RMI Service (RMIService)  Build: 977  Status: running
Remote Shell Service (RemoteShellService)  Build: 977
Status: running Security Service (SecurityService)
Build: 977  Status: running Self Discovery Service
(SelfDiscoveryService)  Build: 977  Status: running
Service Registration Service
(ServiceRegistrationService)  Build: 977  Status:
running Session Service (SessionService)  Build: 977
Status: running Sybase Home Service (SybaseHomeService)
Build: 14  Status: running

Agent Plugin Info:

ASE Cluster Agent Plugin (com.sybase.ase.cluster)
Version: 15.1.0 Build: 85 Instance: 1  Status: running
        Cluster Name: marion
   Env Shell:   /job1/miso/betaR1/SYBASE.sh Shell Type:
sh
    Sybase Home:  /job1/miso/betaR1
    ASE Home:     /job1/miso/betaR1/ASE-15_0
   ASE Version:  Adaptive Server Enterprise/15.0.1/EBF
14721 Cluster Edition/B/x86_64/Enterprise
Linux/asecluster3/2360/64-bit/FBO/Fri Jul 20 10:04:16
2007
    ASE Login:    sa
    Update Time:  60 seconds
    Last Update:  2007-09-28 22:09:02 -0700
```

Usage show agents is active before you connect to a cluster.

# show backupserver config

| | |
|---|---|
| Description | Displays the nodes on which Backup Server is configured and the associated listening port numbers. |
| Syntax | show backupserver config |
| Usage | Use the sybcluster create backupserver command to create a Backup Server. |

# show cluster

| | |
|---|---|
| Description | Displays configuration, log, and status information about the cluster. |
| Syntax | Show<br>Cluster<br>Config<br>Template<br>Log<br>[Errors]<br>[MinSeverity *severity_level*]<br>[StartDate [*date_string*]]<br>[EndDate [*date_string*]]<br>[Last *number_of_lines*]<br>Status |
| Parameters | status<br>displays status information for the cluster. Values are:<br>• Up<br>• Down<br>• Undefined<br>• Invalid<br>• Start<br>• Init<br>• Quiesce<br>log<br>displays logs from all instances in the cluster. |

errors [ minseverity *severity_level* ]
display log file entries for errors. Optionally, limit displayed error entries to a severity level and above.

---

**Note** Error *severities_level* is an attribute of Adaptive Server error messages, not sybcluster messages.

---

startdate [ *date_string* ]
display log file entries that occur on and after the date specified. The format for *date_string* is: *mm:dd:yy*.

If you do not specify a startdate or enddate *date_string*, the default is the current date (today).

enddate [ *date_string* ]
display log file entries that occur on or before the date specified.

last *num_lines*
Limits the number of lines displayed, counting backward from the last line in the log files.

config
displays configuration information for the cluster:

- Maximum number of instances
- Primary and secondary protocols
- Trace flags set
- Location and name of the quorum device
- Location and name of the master device

template
displays formatted configuration information for the cluster.

Examples

**Example 1** Displays current configuration and other information about the default cluster.

```
show cluster status

Id    Name     Node    State   Heartbeat
----------------------------------------
1     ase1     blade1  Up      Yes
2     ase2     blade2  Up      Yes
3     ase3     blade3  Down    No
```

**Example 2** Displays configuration information for the default cluster.

```
show cluster config

**Cluster configuration for "mycluster" **
    Interfaces Path "/work2/sybase/ASE-15_0/"
    Trace Flags:
        15556
    Maximum Instances "4"
    Quorum "/dev/raw/raw101"
    Master Device
         "/dev/raw/raw102"
    logfile ase1 /work2/sybase/ASE-15_0/install/
        ase1.log
    run_parameters ase1 null
    logfile ase2 /work2/sybase/ASE-15_0/install/
        ase2.log
    run_parameters ase2 null

Primary Interconnect "udp"
    Server[1]ase1 tigger.sybase.com 26016 26031
    Server[2]ase2 christopher.sybase.com 26032 26047
Secondary Interconnect "udp"
    Server[1]ase1 tigger.sybase.com 26081 26096
    Server[2]ase2 christopher.sybase.com 26097 26112
```

Usage

show cluster status displays the results of a show instance command on each instance in the cluster.

# show instance

Description

Displays information about an instance.

Syntax

show instance [*instance_name* ] {
config |
status |
log
[ [ errors ] minseverity *severity_level* ] |
[ startdate [ *date_string* ]] |
[enddate [ *date_string* ]] |
[ last *num_lines* ] ] }

Parameters

*instance_name*

specifies a unique name for an instance in the cluster.

status
displays status information for the instance. Values are:

- Up
- Down
- Undefined
- Invalid
- Start
- Init
- Quiesce

log
displays the instance log.

errors [ minseverity *severity_level* ]
displays log file entries for errors. Optionally, limit displayed error entries to a severity level and above.

---

**Note** Error *severities_level* is an attribute of Adaptive Server error messages, not sybcluster messages.

---

startdate [ *date_string* ]
displays log file entries that occur on and after the date specified. The format for *date_string* is: mm:dd:yy.

If a startdate or enddate *date_string* is not specified, *date_string* defaults to the current day.

enddate [ *date_string* ]
displays log file entries that occur on or before the date specified. The format is: mm:dd:yy.

last *num_lines*
Limits the number of lines displayed, counting backwards from the last line in the log file.

Examples

**Example 1** Displays information about "ase1."

```
show instance ase1 status

Id    Name     State
------------------------
1     ase1    Down
```

**Example 2** Displays configuration information for “ase1.”

```
show instance ase1 config

Instance: ase1 at blade6:25001

Private Primary Network
    Address: blade1
    Port Range: 2541 - 2556

Private Secondary Network
    Address: blade1
    Port Range: 2557 - 2572

Log Path: /blade1/sybase/
    ASE-15_0/install/mycluster_ase1.log
```

Usage

- show instance status displays one of seven different states for the named instance:
  - Down
  - Init
  - Invalid
  - Quiesce
  - Start
  - Undefined
  - Up

# show monitorserver config

Description — Displays the Monitor Server name, listening port number, node name, and instance name.

Syntax — show monitorserver config

Usage — Use the sybcluster create monitorserver command to create a Monitor Server.

# show session

| | |
|---|---|
| Description | Displays current discovery and agent information. |
| Syntax | show session |
| Examples | Displays agent status information. |

```
show session

Session information
-------------------

Sybase sybcluster Command Line Utility/15.0.1/CE GA
2/S/jdk1.4.2/sybclustermain/129/Mon Aug 13 09:59:51 PDT 2007

Connected Cluster: mycluster

Default Cluster:
  Default Instance:

  Agent Specifications:
     [1]: oddjob:7171

  Discovery Specifications:

  Agent Connections: 1

     Connection[1] URL: rmi://oddjob:7171

        Node Name:       oddjob1
        Agent Port:      7171
        Agent Version:   2.5.0
        Agent Build:     980
        OS Name:         Linux
        OS Version:      2.6.9-42.ELsmp
        OS Architecture: amd64

      Agent Service Info:
         Agent Service (Agent) Build:980 Status:running
        BootstrapService (BootstrapService)  Build:
          <unavailable>  Status: running
        Configuration Service (ConfigService)  Build:
           980  Status: running
        Deployment Service (DeploymentService)  Build:
           19  Status: running
        Environment Service(EnvironmentDiscoveryService)
           Build: 980  Status: running
       File Transfer Service (FileTransferService)
           Build: 980  Status: running
       Plugin Registration Service
          (PluginRegisterService) Build:980 Status:
```

```
   running
RMI Service (RMIService)  Build: 980  Status:
   running
 Remote Shell Service (RemoteShellService) Build:
   980  Status: running
Security Service (SecurityService)  Build: 980
   Status: running
Self Discovery Service (SelfDiscoveryService)
    Build: 980  Status: running
Service Registration Service
    (ServiceRegistrationService)  Build: 980
    Status: running
Session Service (SessionService)  Build: 980
    Status: running

Sybase Home Service (SybaseHomeService)  Build:
   14  Status: running

Agent Plugin Info:

 ASE Cluster Agent Plugin(com.sybase.ase.cluster)
   Version: 15.0.1 Build: 129 Instance: 1
   Status: running
Cluster Name: mycluster
Env Shell: /oddjob1/work2/
   sybase_sybclustermain_mycluster_vu/SYBASE.sh
   Shell Type: sh
Sybase Home: /oddjob1/
   work2/sybase_sybclustermain_mycluster_vu
ASE Home: /oddjob1/work2/
   sybase_sybclustermain_mycluster_vu/ASE-15_0
ASE Version:  Adaptive Server Enterprise/
   15.0.1/EBF 14721 Cluster Edition/B/x86_64/
   Enterprise Linux/asecluster3/2381/64-bit/
   FBO/Mon Nov 12 07:44:23 2007
ASE Login:    sa
 Update time: 300 seconds
 Last Update:  2007-11-13 15:27:39 -0800
```

Usage

Use the sybcluster show cluster command to view information about the current cluster.

## show xpserver

| | |
|---|---|
| Description | Displays the XP Server name and listening port number, node name, and instance name configured on each node. |
| Syntax | show xpserver |
| Usage | Use the sybcluster create xpserver command to create an XP Server. |

## shutdown cluster

| | |
|---|---|
| Description | Shuts down the cluster by executing a Transact-SQL shutdown command for each instance in the cluster's instance list, in the order specified in the cluster configuration file. |
| Syntax | shutdown cluster [ nowait ] |
| Parameters | nowait<br>shuts down the cluster immediately, without waiting for transactions or statements currently executing to conclude. By default, sybcluster waits for all transactions and statements to execute before shutting down the cluster. |
| Examples | Shuts down the current cluster. |

```
shutdown cluster
```

```
INFO - ...
INFO - 01:00:00000:00117:2007/06/02 00:23:53.56 kernel ueshutdown: exiting
INFO - 01:00:00000:00117:2007/06/02 00:23:53.56 kernel SySAM: Checked in
license for 1 ASE_CORE (2007.1031/31-oct-2007/1293 6876 8FE7 E217).
```

| | |
|---|---|
| Usage | sybcluster prompts for confirmation before shutting down the cluster. |

## shutdown instance

| | |
|---|---|
| Description | Shuts down the instance by executing a Transact-SQL shutdown command. |
| Syntax | shutdown instance [ *instance_name* ] [ nowait ] |

Parameters

*instance_name*
is the unique name of an instance in the cluster.

nowait
shuts down the instance immediately, without waiting for currently executing transactions or statements to finish.

Examples

Shuts down the instance "ase1," after waiting for currently executing transactions or statements to finish.

```
shutdown instance ase1
```

```
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel shutdown server ase1
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 Server SHUTDOWN by request.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 ASE is terminating this
process
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 shut down local cluster
server.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel coordinator to be
shutdown, newcoo is 0.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel Single server
cluster.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel cipcnode_down():
Node 1 down event.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 server ASE shutdown by
request.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel ueshutdown: exiting
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel SySAM: Checked in
license for 1 ASE_CORE (2007.1031.31-oct-2007/1293 6876 8FE7 E 217).
```

Usage

- Shutting down the last instance in a cluster also shuts down the cluster.
- sybcluster prompts for confirmation before shutting down the instance.

# start cluster

Description

Starts all instances in the cluster.

Syntax

start cluster

Examples Starts the current cluster.

```
start cluster

INFO - [cluster boot log]
...

INFO - 02:00:00000:00002:2007/06/02 00:21:53.56 server    'ase1' (ID=1).
INFO - 02:00:00000:00002:2007/06/02 00:21:53.56 server  Master device size:
80 megabytes, or 40960 virtual pages.
```

Usage You must connect to the cluster before starting it.

## start instance

Description Starts an instance.

Syntax start instance [ *instance_name* ] [ unlock ]

Parameters *instance_name*
specifies a unique name for an instance in the cluster. If you do not enter a cluster name, sybcluster uses the instance specified in the sybcluster command line or specified with the use command.

unlock
removes the lock from a cluster that has been terminated unexpectedly. The cluster must be Down before using unlock.

**Warning!** Do not use the unlock parameter unless you have verified that all instances in the cluster are shutdown.

Usage The instance must be Down to use start instance unlock.

## upgrade server

Description Upgrades a nonclustered Adaptive Server to the Adaptive Server Cluster Edition version 15.0.1 ESD1, and creates a cluster with a single instance. You can perform the upgrade by answering prompts at the command line or via an input file.

The Cluster Edition supports upgrading these versions of Adaptive Server:

- Adaptive Server versions 15.0 GA through 15.0.1 ESD3
- Adaptive Server versions 12.5 GA through 12.5.3 ESD6

**Note** You cannot upgrade Adaptive Server 12.5.4 to the Cluster Edition.

Syntax

upgrade server *server_name*
[ login *login_name* ]
[ password *password* ]
[ agent *agent_spec* ]
[ discovery *discovery_spec* ]
[ file *input_file_name* ]
[ checkonly ]

Parameters

*server_name*
is the name of the nonclustered Adaptive Server.

login *login_name*
is the management agent login for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

password *password*
is the management agent password for the Sybase Common Security infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

agent *agent_spec*
is the agent specification that identifies the node in the cluster running a Unified Agent, and the port number that sybcluster uses to connect the Unified Agent. When upgrading a nonclustered Adaptive Server, there is only one node. The format for *agent_spec* is “*node_name:port_number*”. The default port number is “9999.”

discovery *discovery_spec*
is the discovery method used to identify the agents responsible for the requested cluster. The format is “*method*[(*method_specification*)] ”. See Table 14-1 for more information about discovery methods.

file *file_name*
is the input file containing values required for upgrading the server.

checkonly
performs a check run of the nonclustered Adaptive Server to determine its readiness for upgrade.

Examples

**Example** Upgrades “myserver” to the Cluster Edition.

```
upgrade server myserver login uafadmin password " "
agent "myserver:8888"
```

Usage

- upgrade server prompts for these values:
  - The Sybase installation directory of the nonclustered Adaptive Server.
  - The ASE home directory of the nonclustered Adaptive Server.
  - The OCS home directory of the nonclustered Adaptive Server.
  - The name of the first instance in the cluster.
  - Other values as required to create a cluster.
- Use of the checkonly option does not perform any upgrade steps; it does check the server’s readiness for upgrade. Any error conditions found by checkonly must be resolved before actually performing the upgrade.

## use

Description

Specifies the default instance.

Syntax

use *instance_name*

Usage

use overrides the instance name specified in the sybcluster command line.

PART 2

# General Configuration Issues

This part of the Cluster User Guide describes general configuration issues for the Cluster Edition.

CHAPTER 15

# Configuring the Operating System

This chapter discusses the operating system configuration settings that you can adjust after installing or upgrading the Cluster Edition. Unless stated otherwise, the information pertains to all supported UNIX platforms.



## Using the *stty* setting

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

```
stty -tostop
```

If you are redirecting all Cluster Edition output to files, you do not have to change the stty setting.

# Restoring correct permissions

Sybase software files and directories are installed with the correct access permissions. If you notice that the permissions are no longer correct, you can restore the correct permissions with the script setperm_all, located in the *$SYBASE/$SYBASE_ASE/install* directory.

# File descriptors and user connections

The number of user connections used by the Cluster Edition cannot exceed the number of file descriptors available to the Cluster Edition on the operating system. When configuring user connections on the Cluster Edition, the System Administrator should take into account the number of file descriptors available per process. Although most of the open file descriptors are available for user connections, a few are used by the Cluster Edition for opening files and devices.

## For Linux

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

## For Sun Solaris

For Sun Solaris, you can set both soft and hard limits for file descriptors. The soft limit can be increased up to the hard limit by the user, but the hard limit can be increased only by someone with “root” permissions. The soft limit determines the number of open file descriptors available to an Cluster Edition engine. The limit is 10,000.

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

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

## For HP-UX

The kernel parameters maxfiles and maxfiles_lim control the number of file descriptors available to any one process. The maximum number of files descriptors is 10,000 for 32-bit HP-UX systems and 60,000 for 64-bit HP-UX systems.

To display the current values for file descriptors, use the Korn or Bourne shell ulimit command:

```
ulimit -n
```

## Displaying current soft and hard limits

To display the current soft limit, for C shells, enter:

```
limit descriptors
```

For Bourne shells, enter:

```
ulimit -n
```

To display the current hard limit for C shells, enter:

```
limit -h descriptors
```

For Bourne shells, enter:

```
ulimit -Hn
```

## Increasing the soft limit

To increase the soft limit for C shells, enter:

```
limit descriptors n
```

For Bourne shells, enter:

```
ulimit -Sn new_value
```

where *n* is the current value for the soft limit, and *new_value* is the value to which you want to increase the soft limit.

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

## Increasing the hard limit

To increase the hard limit, use a program like the sample program shown in "Sample program" on page 365.

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

1. Create *file_name.c* (where *file_name* is the name you give the file), by using an ASCII text editor. Type the text shown in the sample in "Sample program" on page 365.
2. Compile the file:

   ```
   cc file_name.c -o program_name
   ```

   where *file_name* is the name of the source file you created, and *program_name* is the name you want to give the program.
3. Change the program's permissions and ownership so that it will execute as "root":

   ```
   chmod 755 program_name
   chown root program_name
   ```

   where *program_name* is the name of the compiled program.
4. The "root" user can use the program to start the Cluster Edition with increased user connections by typing the following command at the operating system prompt:

   ```
   # program_name dataserver -d master_device_name
   ```

   where *program_name* is the name of the compiled program, and *master_device_name* is the full path of the Cluster Edition's master device. Instead of typing the command at the operating system prompt, you can add *program_name* preceding the dataserver command line in the the Cluster Edition *RUN_server_name* file.

## Sample program

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

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

```
#include <sys/time.h>
 #include <sys/resource.h>
 #include <sys/types.h>
 /*
 ** define MAX_CONNECTIONS to a number less than
 ** 10000. The number defined will then become the maximum
 ** number of connections allowed by an Adaptive Server.
 */
 #define MAX_CONNECTIONS 9999
 extern int errno;

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

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

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

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

# Adjusting the client connection timeout period

The Cluster Edition uses the KEEPALIVE option of the TCP/IP protocol to detect clients that are no longer active. When a connection to a client is inactive for a period of time (the *timeout period*), the operating system sends KEEPALIVE packets at regular intervals. If it does not receive a response from the client machine for any of these packets, the operating system notifies the Cluster Edition that the client is no longer responding. The Cluster Edition then terminates the client's connection.

The KEEPALIVE default timeout period is 2 hours (7,200,000 ms). To display the current time value, use the command for your platform as shown in the following sections.

## For Sun Solaris

To display the timeout value, enter:

```
/usr/sbin/ndd -get /dev/tcp tcp_keepalive_interval
```

To reduce the timeout period to 15 minutes (900,000 ms.), enter:

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 900000
```

## For Linux

To display the timeout value, enter:

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

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

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

## For HP-UX

To display the current timeout period value, enter:

```
/usr/contrib/bin/nettune -l
```

The tcp-keepstart parameter specifies the length of time (in seconds) that an idle connection is kept active before the system checks to see if the connection is no longer viable.

To change the timeout period, use the nettune -s command:

# Checking for hardware errors

The following types of hardware error messages indicate problems that may lead to database corruption:

- Disk read, write, or retry errors
- Timeouts
- System panics
- Memory problems of any type

## For Sun Solaris

Check the */var/adm/messages* file on a regular basis. If any of the types of hardware errors described in the beginning of this section appear, use the Sun Microsystems diagnostic tool, sundiag, to check memory and disks. See the operating system documentation for more information.

## For Linux

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

## For HP-UX

Check the */var/adm/syslog/syslog.log* file on a regular basis. You can view the file directly, or you can use the HP-UX dmesg command. See the operating system documentation for more information.

# Monitoring the use of operating system resources

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

Sun Solaris and Linux supply the following tools to help monitor performance:

- The iostat command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.
- The ps command gives you an accurate snapshot of accumulated CPU time and usage for individual processes. This can be very helpful in determining the dataserver-, engine-, and process-specific loading.
- The time command can be useful in determining the various user, system, and real-time resources used over a complete run.

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

# A sample C shell maintenance script

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

```
#!/bin/csh -f
 if ( -e dbcc_mail.out) then
   rm dbcc_mail.out
 endif
 foreach i (*.dbcc)
 isql -Usa -Ppassword < $i > dbcc_out
 if ( 'grep -c 'Msg 25[0-9][0-9]' dbcc_out' ) then
   echo "There are errors in" $i >> dbcc_mail.out
   cat dbcc_out >> dbcc_mail.out
 else
   echo "Backing up " $i:r >> dbcc_mail.out
   isql -Usa -Ppassword < $i:r.backup
 endif
 end
 mail -s "Backup Report" jjones < dbcc_mail.out
```

The first set of scripts (one for each database with a file name appended with *.dbcc*) runs dbcc checkalloc and dbcc checkdb for each database and sends the messages to an output file called *dbcc_out*.

For example, the script master.dbcc runs dbcc to check the master database:

```
dbcc checkalloc (master)
go
dbcc checkdb (master)
go
```

The C shell script then runs the grep command to find 2500-level error messages in the dbcc output. The results of the grep command go into an output file called *dbcc_mail.out*.

Next, the script invokes an isql backup script for each database for which no 2500-level errors occurred and adds the "Backing up *database_name*" line to *dbcc_mail.out*. For example, the script master.backup backs up the master database:

```
use master
go
dump database master to master_dump
go
```

You may want to add appropriate dump transaction commands to your scripts.

If there are 2500-level error messages, the script does not back up the database. At the end of the script, *dbcc_mail.out* is mailed to the System Administrator "jjones," who then has a record of fatal dbcc errors and successful backups.

You can tailor the sample shell and isql scripts to suit the needs of your installation.

To have the scripts execute automatically, edit the *crontab* file, and add an entry similar to this:

```
00 02 * * * /usr/u/sybase/dbcc_ck 2>&1
```

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

CHAPTER 16

# Customizing Localization for the Cluster Edition

This chapter provides information about Sybase localization support for international installations, including configuring languages, character sets, and sort order. For more information, see the *System Administration Guide*.



## Overview of localization support

Localization is the process of setting up an application to run in a particular language or country environment, including translated system messages and correct formats for date, time, and currency. The Cluster Edition supports localization for international customers and for customers with heterogeneous environments.

This support includes:

- Data processing support – the Cluster Edition comes with character set and sort order definition files it uses to process the characters used in different languages.

  Sybase provides support for the major languages in:

  - Western Europe
  - Eastern Europe
  - Middle East

    - Latin America
    - Asia
- Translated system messages – the Cluster Edition includes language modules for:
    - Brazilian Portuguese
    - Chinese (Simplified)
    - French
    - German
    - Japanese
    - Korean
    - Polish
    - Spanish
    - Thai

## Language modules

The Cluster Edition stores its localized software messages in separate language modules.

When you install a language module, the installation program loads the messages, character set, and sort-order files that support the new language in the correct locations.

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

## Default character sets for servers

The default character set is the character set in which data is encoded and stored on the Cluster Edition databases.

## Changing the default language and character set

**Warning!** Make all changes to the character set and sort order for a new server before creating any user databases or making any changes to the Sybase-supplied databases. Changing the character set and sort order after data or data structures have been added to the Cluster Edition may require additional steps. To change the character set or sort order after you have added data, see the *System Administration Guide*.

sybcluster and the ASE plug-in create an instance with the following defaults:

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

## Changing the default character set for servers

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

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

  For example, if most of your clients use ISO 8859-1, you can minimize the amount of data conversion that has to occur by specifying ISO 8859-1.
- If your server is operating in a heterogeneous language environment, choose a character set that works with all the character sets needed. Often, this is Unicode (UTF-8).

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

## Supported character sets

The following language, scripts and character sets are supported by the Cluster Edition:

- Arabic – see Table 16-1 on page 374.
- Baltic – see Table 16-2 on page 375.
- Chinese, Simplified – see Table 16-3 on page 375.
- Chinese, Traditional – see Table 16-4 on page 375
- Cyrillic – see Table 16-5 on page 375.
- Eastern European – see Table 16-6 on page 376.
- Greek – see Table 16-7 on page 376.
- Hebrew – see Table 16-8 on page 376.
- Japanese – see Table 16-9 on page 376.
- Korean – see Table 16-10 on page 376.
- Thai – see Table 16-11 on page 377.
- Turkish – see Table 16-12 on page 377.
- Unicode (which supports over 650 languages) – see Table 16-13 on page 377.
- Vietnamese – see Table 16-14 on page 377.
- Western European – see Table 16-15 on page 377.

The tables define each character set.

For more information see “Character set conversion” on page 378.

Table 16-1 lists the Arabic character set:

***Table 16-1: Arabic character sets***

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

Table 16-2 lists the Baltic character set:

***Table 16-2: Baltic character sets***

| Character set | Description |
|---|---|
| cp1257 | Microsoft Windows Baltic |

Table 16-3 lists the simplified Chinese character set:

***Table 16-3: Simplified Chinese character sets***

| Character set | Description |
|---|---|
| eucgb | EUC GB encoding = Simplified Chinese character sets |
| cp936 | Microsoft Simplified Chinese character sets |
| gb18030 | PRC 18030 standard |

Table 16-4 lists the traditional Chinese character set:

***Table 16-4: Traditional Chinese character set***

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

Table 16-5 lists the Cyrillic character set:

***Table 16-5: Cyrillic character sets***

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

Table 16-6 lists the Eastern European character set:

**Table 16-6: Eastern European character sets**

| Character set | Description |
|---|---|
| cp852 | PC Eastern Europe |
| cp1250 | Microsoft Windows 3.1 Eastern European |
| iso88592 | ISO 8859-2 Latin-2 |

Table 16-7 lists the Greek character set:

**Table 16-7: Greek character sets**

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

Table 16-8 lists the Hebrew character set:

**Table 16-8: Hebrew character sets**

| Character set | Description |
|---|---|
| cp1255 | Microsoft Windows Hebrew |
| iso88598 | ISO 8859-8 Hebrew |

Table 16-9 lists the Japanese character set:

**Table 16-9: Japanese character sets**

| Character set | Description |
|---|---|
| cp932 | IBM J-DBCS:CP897 + CP301 (Shift-JIS) |
| eucjis | EUC-JIS encoding |
| sjis | Shift-JIS (no extensions) |
| deckanji | DEC Kanji |

Table 16-10 lists the Korean character set:

**Table 16-10: Korean character sets**

| Character set | Description |
|---|---|
| eucksc | EUC KSC Korean encoding = CP949 |
| cp949 | Ms Windows Korean |

Table 16-11 lists the Thai character set:

***Table 16-11: Thai client character sets***

| Character set | Description |
|---|---|
| tis620 | TIS-620 Thai standard |
| cp874 | Microsoft Windows Thai |

Table 16-12 lists the Turkish character set:

***Table 16-12: Turkish character sets***

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

Table 16-13 lists the Unicode character set:

***Table 16-13: Unicode character set***

| Character set | Description |
|---|---|
| utf8 | Unicode UTF-8 encoding |

Table 16-14 lists the Vietnamese character set:

***Table 16-14: Vietnamese character set***

| Character set | Description |
|---|---|
| cp1258 | Microsoft Windows Vietnamese |

Table 16-15 lists the Western European character set:

***Table 16-15: Western European character set***

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

| Character set | Description |
|---|---|
| mac_euro | Macintosh Roman with EURO support |

# Character set conversion

Backup Server passes messages to the Cluster Edition in the client's language and in the Cluster Edition character set. The Cluster Edition then converts the messages and issues them in the client's language and character set. Keep the following requirements in mind when selecting a character set:

- In a heterogeneous environment, the Cluster Edition and Backup Server may need to communicate with clients running on different platforms and using different character sets. To maintain data integrity, the server converts the code between the character sets.
- Unicode conversions exists for all native character sets. When converting between two native character sets, Unicode conversion uses Unicode as an intermediate character set. For example, to convert between the server default character set (CP 437), and the client character set (CP 860), CP 437 is first converted to Unicode; Unicode is then converted to CP 860. By default, Unicode conversion is used for character set conversion.
- Adaptive Server direct conversions support conversions between two native character sets of the same language group. For example, Adaptive Server supports conversion between CP 437 and CP 850, because both belong to the group 1 language group. To use the direct conversion, you must install the character set definition files on the server for all the character sets being used by your clients. To enable direct conversion, you must run `sp_configure 'enable unicode conversion', 0` to disable Unicode conversion.

If either the Cluster Edition or Backup Server does not support a client's language or character set, that server issues a warning message. Errors also occur when the Backup Server character set is not compatible with the Cluster Edition character set.

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

### Conversions between server and client

If the Cluster Edition does not support the client's language or character set, the client can connect with the server, but no character conversions occur.

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

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

## Sort orders

Each character set comes with one or more sort orders (collating sequences), which are either located in the sort-order definition files (*.srt* files) or installed in the system if it is a Unicode sort order. These files accompany the character set definition files and can be found in the same directory.

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

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

## Available sort orders

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

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

**Note** Available sort orders vary according to the character set installed on the Cluster Edition.

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

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

To view available Unicode sort orders, please run sp_helpsort. For more information about localization files, see “Localization directories” on page 384.

Table 16-16 describes available sort orders.

***Table 16-16: Sort orders available in the Cluster Edition***

| Sort order name | Description |
|---|---|
| Binary order | Sorts all data according to numeric byte values for that character set. Binary order sorts all ASCII uppercase letters before lowercase letters. Accented or ideographic (multibyte) characters sort in their respective standards order, which may be arbitrary.<br>All character sets have binary order as the default. If binary order does not meet your needs, you can specify one of the other sort orders either at installation or at a later time by, using the charset utility. |
| Dictionary order, case sensitive, accent sensitive | Case sensitive. Sorts each uppercase letter before its lowercase counterpart, including accented characters. Recognizes the various accented forms of a letter and sorts them after the associated unaccented letter. |
| Dictionary order, case insensitive, accent sensitive | Case-insensitive dictionary sort order. Uppercase letters are equivalent to their lowercase counterparts and are intermingled in sorting results. |
| Dictionary order, case insensitive, accent insensitive | Case-insensitive dictionary sort order. Diacritical marks are ignored. |

| Sort order name | Description |
|---|---|
| Dictionary order, case insensitive with preference | Case-insensitive dictionary sort order, with case preference for collating purposes. A word written with uppercase letters is equivalent to the same word written with lowercase letters.<br>Uppercase and lowercase letters are distinguished only when you use an order by clause. The order by clause sorts uppercase letters before it sorts lowercase.<br>**Note** Do not select this sort order unless your installation requires that uppercase letters be sorted before lowercase letters in otherwise equivalent strings for order by clauses. Using this sort order may reduce performance in large tables when the columns specified in an order by clause match the key of the table's clustered index. |
| Alternate dictionary order, case sensitive | Case-sensitive alternate dictionary sort order with lowercase variants sorted before uppercase.<br>Use with several of the Western European languages. |
| Alternate dictionary order, case insensitive, accent insensitive | Case-insensitive and accent-insensitive alternate dictionary sort order.<br>Use with several of the Western European languages. |
| Alternate dictionary order, case insensitive, uppercase preference | Case-insensitive alternate dictionary sort order with uppercase preference.<br>Use with several of the Western European languages. |
| Spanish dictionary order, case sensitive | Case-sensitive Spanish dictionary sort order.<br>Use with Spanish and for most Latin American locales. |
| Spanish dictionary order, case insensitive | Spanish case-insensitive dictionary sort order.<br>Use with Spanish and for most Latin American locales. |
| Spanish dictionary order case insensitive, accent insensitive | Spanish case-insensitive and accent-insensitive dictionary sort order.<br>Use with Spanish and for most Latin American locales. |
| Scandinavian dictionary order, case sensitive | Case-sensitive dictionary sort order.<br>Use with Scandinavian languages. |
| Scandinavian dictionary order, case insensitive, uppercase preference | Case-insensitive and accent-insensitive dictionary sorting, with uppercase preference.<br>Use with Scandinavian languages. |

To see the sort orders that are available, use the charset utility to display the sort orders for the character sets you plan to use. For mo7re information on unicode sort orders for UTF-8, see "Configuring Character Sets, Sort Orders, and Languages" in the *System Administration Guide*.

Table 16-17 lists the available default Unicode sort orders.

***Table 16-17: Default Unicode sort orders***

| Name | ID | Description |
|---|---|---|
| defaultml | 20 | Default Unicode multi-lingual ordering |
| thaidict | 21 | Thai dictionary ordering |
| iso14651 | 22 | Ordering as per ISO14651 standard |
| utf8bin | 24 | Ordering for UTF-16 that matches the UTF-8 binary |
| binary | 25 | Binary sort |
| altnoacc | 39 | Alternate accent-insensitive |
| altdict | 45 | Alternate dictionary ordering |
| altnocsp | 46 | Alternate case-insensitive with preference |
| scandict | 47 | Scandinavian dictionary ordering |
| scannocp | 48 | Scandinavian case-insensitive with preference |
| bin_utf8 | 50 | UTF-8 binary sort order |
| dict | 51 | General-purpose dictionary ordering |
| nocase | 52 | General-purpose case-insensitive dictionary ordering |
| nocasep | 53 | General-purpose case-insensitive with preference |
| noaccent | 54 | General-purpose accent-insensitive dictionary ordering |
| espdict | 55 | Spanish dictionary ordering |
| espnocs | 56 | Spanish case-insensitive dictionary ordering |
| espnoac | 57 | Spanish accent-insinuative dictionary ordering |
| rusnocs | 59 | Russian case-insensitive dictionary ordering |
| cyrnocs | 64 | Cyrillic case-insensitive dictionary ordering |
| elldict | 65 | Greek dictionary ordering |
| hundict | 69 | Hungarian dictionary ordering |
| hunnoac | 70 | Hungarian accent-insensitive dictionary ordering |
| hunnocs | 71 | Hungarian case-insensitive dictionary ordering |
| turknoac | 73 | Turkish accent-insensitive dictionary ordering |
| cp932bin | 129 | CP932 binary sort order |
| dynix | 130 | GB pinyin sortorder |
| gb2312bn | 137 | GB2312 binary sort order |
| cyrdict | 140 | Cyrillic dictionary sort order |
| turdict | 155 | Turkish dictionary sort order |
| euckscbn | 161 | EUCKSC binary sort order |
| gbpinyin | 163 | GB pinyin sort order |
| rusdict | 165 | Russian dictionary sort order |
| sjisbin | 179 | SJIS binary sort order |

| Name | ID | Description |
|---|---|---|
| eucjisbn | 192 | EUCJIS binary sort order |
| big5bin | 194 | BIG5 binary sort order |

# Language modules

If you want the Cluster Edition error messages to be displayed in a language other than U.S. English (us_english), you must install the appropriate language module.

When you install a new language module, installation automatically loads the language into the Sybase installation directory to support the new language. For information about directories, see “Localization directories” on page 384.

## Installing a new language module

A full server installation includes all the language components automatically. If you did not select a full install, you must install additional language modules manually.

To install a new language module:

1 Load the language module software from the distribution media. You must load this software into the same directory in which you loaded the Cluster Edition.

2 Reconfigure the language and, if necessary, the character set and sort order for the Cluster Edition. For instructions, see “Changing the localization configuration” on page 388.

## Message languages

For messages, U.S. English is installed as the default language in the Cluster Edition. The following rules apply to language modules:

- During the Cluster Edition installation or reconfiguration, you can specify a default language other than U.S. English. However, you must have installed the language module for the language you specify.

- If your clients require the Cluster Edition messages in a language other than U.S. English, you must load the language module for those languages. Then, you can configure the Cluster Edition to the language used by your clients.
- If the Cluster Edition does not support messages in a client's language, these clients receive messages in the server default language.

  For example, if your client's language is Latin, the Spanish language module is installed, and Spanish is specified as the Cluster Edition default language, the client receives messages in Spanish.

# Localization

By default, the Cluster Edition and Backup Server configurations use the English locale settings, which include:

- Character set definition files for Western European character sets
- Sort-order definition files for Western European character sets
- U.S. English system message files

During the installation process or through reconfiguration, you can specify a different language, character set, and sort order.

## Localization directories

Sybase localization configuration involves the following directories:

- *locales*
- *charsets*

The table below illustrates the structure of the localization files. It does not show a complete list of all the files.

| | | |
|---|---|---|
| *$SYBASE/charsets* | *charset_name* | **.srt* files |
| | *charset_name...* | *charset.loc* |
| | *unicode* | **.uct* files |

| | | |
|---|---|---|
| *$SYBASE/$SYBASE_ASE/locales* | *language_name* | *charset_name* |
| | *language_name...* | *charset_name...* |
| *%SYBASE/locales* | *locales.dat* | |
| | *message* | *language_name* |
| | | *language_name...* |

## About the directory

The *$SYBASE/locales* and *$SYBASE/SYBASE_ASE/locales* directory contains a subdirectory for each available language. Each language subdirectory contains a subdirectory for each character set available with that language.

- The *.loc* files in these subdirectories enable the Cluster Edition or Backup Server to report errors in a specific language, encoded in a specific character set.

  There are a variety of *.loc* files in each subdirectory. Most of these files contain translated error messages for a specific product or utility.
- The *common.loc* file in each subdirectory contains localized information, such as local date, time, and currency formatting, that is used by all products.
- The *locales.dat* file contains entries that associate platform-specific locale names with Sybase language and character set combinations.

## About the *charsets* directory

The files in *$SYBASE/charsets/charset_name* contain information related to each particular character set, such as the definition of the character set and any sort orders available for that character set.

## About the *locales.dat* file

You can edit the *locales.dat* file to:

- Change the default language or character set for a platform, or
- Add new associations between platform locale names and Sybase language and character set names.

## Format of *locales.dat* file entries

Each entry in the *locales.dat* file links a platform-specific locale definition to a Sybase language and character set combination. Each entry has the following format:

```
locale = platform_locale, syb_language, syb_charset
```

where:

- *platform_locale* is the platform-specific keyword for a locale. For acceptable values, see your operating system documentation.

  When the locale being defined as the default for the site, *platform_locale* is "default."

- *syb_language* is the name of the language directory to be used from within *$SYBASE/locales/language_name*.
- *syb_charset* is the character set name that determines the character set conversion method and identifies the directory location of the message files for clients from within *$SYBASE/locales/language_name/charset_name*.

For example, the following entry specifies that the default locale uses us_english for the language and iso_1 for the character set:

```
locale = default, us_english, iso_1
```

## How client applications use *locales.dat*

Client applications use the *locales.dat* file to identify the language and character set to use. The connection process follows these steps:

1 When a client application starts, it checks the operating system locale setting and then checks the *locales.dat* file to see if that setting is appropriate for the Cluster Edition. For example, a locale entry for French can look like the following:

```
locale = fr_FR, french, iso_1
```

2 When the client connects to the Cluster Edition, the language and character set information is passed to the Cluster Edition in the login record.

3 the Cluster Edition then uses:

- The character set information, for example, iso_1, to identify the client's character set and verify whether it can convert character data to this character set

- The language (in the preceding example, French) and character set information to see if it has messages in the client's language

**Note** The Cluster Edition software includes some locale entries already defined in the *locales.dat* file. If these entries do not meet your needs, you can either modify them or add new locale entries.

## Editing the *locales.dat* file

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

To edit the *locales.dat* file:

1 Open the *locales.dat* file copy in a text editor.

2 Find the section enclosed in brackets:

- For Sun Solaris, *[sun_svr4]*
- For HP, *[hp ux]*
- For IBM, *[aix]*

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

**Note** The value for *platform_locale* must match the value required by your operating system. If the locales definitions in your system configuration files do not match the Sybase locale definitions, your applications will not run properly.

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

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

4 Add the required entry or modify an existing entry.

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

# Changing the localization configuration

By default, the Cluster Edition and Backup Server configurations uses the English locale settings localization, which include:

- Character set definition files for Western European character sets
- Sort order definition files for Western European character sets
- us_english system message files

During the installation process and through reconfiguration, you can specify a different language, character set, and sort order.

## Cluster Edition localization

Each language uses about 2MB of database space per module. If necessary, use the alter database command to increase the size of the master database before adding another language.

---

**Note** If you want to install more than one language on the Cluster Edition, and the master database is not large enough to manage more than one language, the transaction log may become too full. You can expand the master database only on the master device. For more information, see the *System Administration Guide*.

---

1 Source *SYBASE.csh* or *SYBASE.sh* if you have not set up the Sybase environment variables.

2 Use the langinstall utility to configure localization for the Cluster Edition:

```
$SYBASE/$SYBASE_ASE/bin/langinstall
```

This is the syntax for langinstall:

```
langinstall [-S server_name] [-U user_name] [-P password]
[-R release_number] [-I path_to_interfaces] [-v] language character_set
```

For example, to install the French language with the iso_1 default character set:

```
langinstall -Usa -P -Sserver_name french iso_1
```

## Backup Server localization

You can change the Backup server language and character set by modifying the *RUN_<backup_server_name>* file. See the *Utility Guide* for more information on the backupserver command arguments.

### Configuring the Cluster Edition for other character sets

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

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

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

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

Replace *sort_order_file* with the name of the sort order file. See Table 16-18 on page 390. Replace *character_set* with the Sybase name for your character set. See Table 16-19 on page 391.

2 Use charset utility to load any additional character sets. See “charset utility” on page 393 for more about this utility.

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

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

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

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

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

Replace *sort_order_id* with the ID for your sort order. See Table 16-18 on page 390. Replace *character_set* with the Sybase name for your character set. See Table 16-19 on page 391.

5 Shut down the cluster. You can use sybcluster, the Adaptive Server Plug-in, or, if you configured the cluster manually, a command line option.

6 Restart any instance in the cluster using sybcluster or the Adaptive Server Plug-in.

**Note** If you configured the cluster manually, use your normal process on UNIX systems to restart an instance, usually by invoking one of the *RUN_xxx* scripts from *$SYBASE/$SYBASE_ASE/install*.

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

8 Check the cluster log file to verify that the charset and sortorder changes have completed successfully.

## Sort orders

Table 16-18 describes the available sort orders. If your language does not appear, then there is no language-specific sort order for your language—use a binary sort order.

***Table 16-18: Available sort orders***

| Language or script | Sort orders | File name | ID |
|---|---|---|---|
| All languages | Binary order | *binary.srt* | 50 |
| Central European Czech, Slovak | Dictionary order, case sensitive, accent sensitive | *czedit.srt* | 80 |
| These sort orders work only with | Dictionary order, case insensitive, accent sensitive | *czeocs.srt* | 82 |
| CP 852, CP 1250, and ISO 8859-2 | | | |
| | Dictionary order, case insensitive, accent insensitive | *czenoac.srt* | 81 |
| Cyrillic | Dictionary order, case sensitive, accent sensitive | *cyrdict.srt* | 63 |
| | Dictionary order, case sensitive, accent sensitive | *cyrnocs.srt* | 64 |
| English | Dictionary order, case sensitive, accent sensitive | *dictiona.srt* | 51 |
| French | Dictionary order, case insensitive, accent sensitive | *nocase.srt* | 52 |
| German | Dictionary order, case insensitive, accent sensitive, with preference | *nocasepr.srt* | 53 |
| These sort orders work with all Western European character sets. | Dictionary order, case insensitive, accent insensitive | *noaccent.srt* | 54 |

| Language or script | Sort orders | File name | ID |
|---|---|---|---|
| English<br>French<br>German<br>These sort orders work only with CP 850. | Alternate dictionary order, case sensitive | *altdict.srt* | 45 |
| | Alternate dictionary order, case sensitive, accent insensitive | *altnoacc.srt* | 39 |
| | Alternate dictionary order, case sensitive, with preference | *altnocsp.srt* | 46 |
| Greek<br>This sort order works only with ISO 8859-7. | Dictionary order, case sensitive, accent sensitive | *elldict.srt* | 65 |
| Hungarian<br>These sort orders work only with ISO 8859-2. | Dictionary order, case sensitive, accent sensitive | *hundict.srt* | 69 |
| | Dictionary order, case insensitive, accent sensitive | *hunnoac.srt* | 70 |
| | Dictionary order, case insensitive, accent insensitive | *hunnocs.srt* | 71 |
| Russian<br>This sort order works with all Cyrillic character sets except for CP 855. | Dictionary order, case sensitive, accent sensitive | *rusdict.srt* | 58 |
| | Dictionary order, case insensitive, accent sensitive | *rusnocs.srt* | 59 |
| Scandinavian<br>These sort orders work only with CP 850 and CP858 | Dictionary order, case sensitive, accent sensitive | *scandict.srt* | 47 |
| | Dictionary order, case insensitive, with preference | *scannocp.srt* | 48 |
| Spanish | Dictionary order, case sensitive, accent sensitive | *espdict.srt* | 55 |
| | Dictionary order, case insensitive, accent sensitive | *espnocs.srt* | 56 |
| | Dictionary order, case insensitive, accent insensitive | *espnoac.srt* | 57 |
| Thai | Dictionary order | *dictionary.srt* | 51 |
| Turkish<br>These sort orders work only with ISO 8859-9. | Dictionary order, case sensitive, accent sensitive | *turdict.srt* | 72 |
| | Dictionary order, case insensitive, accent insensitive | *turnoac.srt* | 73 |
| | Dictionary order, case insensitive, accent sensitive | *turnocs.srt* | 74 |

## Character sets

Table 16-19 lists the supported character sets and their Sybase name.

***Table 16-19: Sybase character set names***

| Character sets | Sybase name |
|---|---|
| ASCII 8 | acsii_8 |
| Big 5 | big5 |
| Big 5HK | big5hk |
| CP 437 | cp437 |
| CP 850 | cp850 |

| Character sets | Sybase name |
|---|---|
| CP 852 | cp852 |
| CP 855 | cp855 |
| CP 857 | cp857 |
| CP 858 | cp858 |
| CP 860 | cp860 |
| CP 864 | cp864 |
| CP 866 | cp866 |
| CP 869 | cp869 |
| CP 874 | cp874 |
| CP 932 | cp932 |
| CP 936 | cp936 |
| CP 949 | cp 949 |
| CP 950 | cp950 |
| CP 1250 | cp1250 |
| CP 1251 | cp1251 |
| CP 1252 | cp1252 |
| CP 1253 | cp1253 |
| CP 1254 | cp1254 |
| CP 1255 | cp1255 |
| CP 1256 | cp1256 |
| CP 1257 | cp1257 |
| CP 1258 | cp1258 |
| DEC Kanji | deckanji |
| EUC-CNS | euccns |
| EUC-GB | eucgb |
| EUC-JIS | eucjis |
| EUC-KSC | eucksc |
| GB 18030 | gb18030 |
| GREEK8 | greek8 |
| ISO 8859-1 | iso_1 |
| ISO 8859-2 | iso88592 |
| ISO 8859-5 | iso88595 |
| ISO 8859-6 | iso88596 |
| ISO 8859-7 | iso88597 |
| ISO 8859-8 | iso88598 |
| ISO 8859-9 | iso88599 |
| ISO 8859-15 | iso15 |

| Character sets | Sybase name |
|---|---|
| Koi8 | koi8 |
| MAC | mac |
| MAC_CYR | mac_cyr |
| MAC_EE | mac_ee |
| MAC_EURO | mac_euro |
| MACGRK2 | macgrk2 |
| MACTURK | macturk |
| ROMAN8 | roman8 |
| ROMAN9 | roman9 |
| Shift-JIS | sjis |
| TIS 620 | tis620 |
| TURKISH8 | turkish8 |
| UTF-8 | utf8 |

## *charset* utility

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

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

Syntax

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

***Table 16-20: Keywords and options for charsets***

| Keywords and options | Description |
|---|---|
| -U | If you are not already logged in to your operating system as “sa”, you must specify “-Usa” or “/username = sa” in the command line. |
| -P | Specifies the “sa” password on the command line. If not specified, the user is prompted for the “sa” password. |

| Keywords and options | Description |
|---|---|
| -S | Specifies the name of the server. If not specified, charset uses the DSQUERY environment variable to identify the server name. If there is no DSQUERY environment variable, charset attempts to connect to a server named "SYBASE." |
| -I | Specifies the interfaces file to use. If not specified, charset uses the interfaces file in the SYBASE directory. |
| -v | Causes the Sybase version string to be printed, then exits. Use with no other options specified. |
| *sort_order* | When charset is used to load the default character set and sort order, *sort_order* is a mandatory parameter specifying the name of the sort order file to be used by the Cluster Edition. When loading additional character sets, use *charset.loc* to indicate the name of the character set files. |
| *charset* | Specifies the directory of the character set to be used by the Cluster Edition. |

CHAPTER 17

# Adding Optional Functionality to the Cluster Edition

This chapter provides instructions for adding optional functionality to the Cluster Edition:



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

## Adding auditing

Auditing is an important part of security in a database management system. Security-related system activity is recorded in an audit trail, which can be used to detect penetration of the system and misuse of resources. By examining the audit trail, the System Security Officer can inspect patterns of access to objects in databases and can monitor the activity of specific users. Audit records can be traced to specific users, enabling the audit system to act as a deterrent to users who are attempting to misuse the system.

A System Security Officer manages the audit system and is the only user who can start and stop auditing, set up auditing options, and process audit data.

### Audit system devices and databases

The audit system includes several components. The main components are:

- The sybsecurity device and the sybsecurity database, which stores audit information

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

**The sybsecurity device and database**

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

**Tables and devices for the audit trail**

The Cluster Edition stores the audit trail in system tables, named sysaudits_01 through sysaudits_08. At any given time, only *one* of the audit tables is *current*. The Cluster Edition writes all audit data to the current audit table. A System Security Officer can use sp_configure to set or change which audit table is current.

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

**Device for syslogs systems table**

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

## Running auditinit with the Cluster Edition

If you use auditinit with a nonclustered versions of Adaptive Server, auditinit starts the server if it is not already running. However the Cluster Edition requires that an instance is running before you use the auditinit utility. auditinit does not start the instance if it has not already started. If you attempt to log into a Cluster Edition with auditinit that is not started, auditinit displays this warning message:

```
Can not login to server because it is not running.
Please manually start the server and retry
```

## Pre-installation tasks for auditing devices

Determine the location of the raw devices for the sybsecurity, syslogs, and sysaudits table devices. You will need to provide this information later.

Sybase recommends that you:

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

## Installing auditing

❖ **Configuring the Cluster Edition for auditing**

1. Source *SYBASE.csh* or *SYBASE.sh* file if you have not setup the Sybase environment variables.
2. Start auditinit at the UNIX prompt (the auditinit utility is located in *$SYBASE/$SYBASE_ASE/install*):

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

   auditinit displays the following menu:

```
AUDITINIT
1. Release directory:  /usr/u/sybase
2. Configure a Server product
```

3 Select Configure a Server Product.

4 Select Adaptive Server.

5 Select Configure an Existing Sybase Server.

6 Select the server to configure.

7 Provide the SA password for the server you selected.

8 From the Sybase Server Configuration screen, select Configure Auditing.

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

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

List of devices for the audit tables:
Logical name    Physical name    Segment name    Table name    Size

Device for the audit datbase transaction log:
Logical name    Physical name    Segment name    Table name    Size
```

9 From the Configure Auditing screen, select Configure Auditing.

auditinit redisplays the Configure Auditing menu with the value “yes” displayed for Configure Auditing.

10 Restart the Cluster Edition for the changes to take effect.

❖ **Creating a device for an audit table**

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

auditinit displays the following menu:

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

```
4. Device size for auditing:
```

2 Select Sybsecurity Physical Device Name.

To create a device for an audit table:

1 Enter the *full path* of the physical device (raw partition) that you located in "Pre-installation tasks for auditing devices" on page 397, where *path_to_partition* is the path to the raw partition for the device.

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

/dev/path_to_partition
```

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

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

2 Press Return to acknowledge the warning.

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

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

3 Proceed through the remaining items on this menu.

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

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

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

```
List of devices for the audit tables:
Logical name   Physical name    Segment name         Table name    Size

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

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

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

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

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

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

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

❖ **Creating a device for the audit database transaction log**

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

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

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

2 Select Sybsecurity Physical Device Name.

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

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

```
/dev/path_to_partition
```

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

3 Enter the full path name of a physical device.

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

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

4 Press Return to acknowledge this warning.

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

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

5 Proceed through the remaining items on this menu. As you do so, be aware of the following:

- Sybase recommends a minimum size of 2MB for the size of the transaction log.
- auditinit displays the size in both Size of the Device and in Device Size for Auditing in the Add/Change a New Device for Auditing menu.
- The Device Size for Auditing default value is equal to the size of the device, based on the assumption that you may want to devote the entire device to log for the auditing task. If you want to use only a subset of the device, you can edit the Size of the Device value.

6 Press Ctrl+A to accept the settings displayed in the Add/Change a New Device for Auditing menu.

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

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

```
List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

6. Audit_01'   /secret1/sybase_  dr/install/aud1.dat’ sysaudits_01   5
7. Audit_02'   /secret1/sybase_  dr/install/aud2.dat' sysaudits_02   5
8. auditlog    /secret1/.../auditlog.dat logsegment   syslogs        2
```

7 When you are ready to execute the audit configuration, press Ctrl+A. auditinit returns you to the Sybase Server Configuration screen.

8 Press Ctrl+A again. auditinit prompts with:

```
Execute the Sybase Server Configuration now?
```

9 Enter “y” (yes).

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

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

## Enabling auditing

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

### ❖ Deleting a device entry

1 Select Delete a Device Entry from the Configure Auditing menu.

2 Enter the number of the device to delete.

3 Press return.

### ❖ Changing a device entry

1 Select Change a Device Entry from the Configure Auditing menu.

2 Enter the number of the device to change.

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

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
      /secret1/sybase_dr/install/audlog
```

```
2. Logical name of the device: aud.log
3. size of the new device (Meg): 5
4. Device size for auditing:5
```

3 Select each remaining entry you want to change.

4 Press Ctrl+A to save the new entries.

# Installing online help for Transact-SQL syntax

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

## Online syntax help: *sp_syntax*

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

The *scripts* directory contains one or more of the sp_syntax scripts shown in Table 17-1, depending on which Sybase products are included with your server:

***Table 17-1: sp_syntax installation scripts***

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

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

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

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

## Default device for the *sybsyntax* database

The sybsyntax database requires 3MB on your database device. By default, the sybsyntax installation scripts install the sybsyntax database on the device that is designated as the default database device.

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

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

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

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

## Installing *sybsyntax*

For each sybsyntax installation script you want to execute:

1 Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device where you plan to store the sybsyntax database. You will need to provide this information later.

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

3 Use a text editor to edit the script, if necessary, to change the default device from the master device to the device created in step 1. For information on the default device, see "Default device for the sybsyntax database" on page 404.

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

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

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

```
create database sybsyntax on device_name
```

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

4 Execute the script with a command like the following:

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

where *sa* is the user ID of the System Administrator, *password* is the System Administrator's password, and *servername* is the Cluster Edition where you plan to install the database.

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

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

```
isql -Usa -Ppassword -Sservername

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

The Cluster Edition displays a list of commands that contain the word or word fragment “select.”

CHAPTER 18

# Logging Error Messages and Events

This chapter describes how to use the error logging features of the Cluster Edition.



## Cluster Edition error logging

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

*$SYBASE/$SYBASE_ASE/install/instance_name.log*

This file:

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

The Cluster Edition includes the instance ID to the front of the error log header with this format:

```
instance id: engine number : family id: process id: date time
```

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

## Enabling and disabling error logging

Logging to the Cluster Edition error log is always enabled. However, when you create or modify a specific user-defined message, you can set it to be omitted from the log. See “Logging user-defined messages” on page 410.

# Setting error log paths

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

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

- Cluster Edition: *$SYBASE/$SYBASE_ASE/install/instance_name.log*
- Backup Server: *$SYBASE/$SYBASE_ASE/install/instance_name_back.log*
- Monitor Server: *$SYBASE/$SYBASE-ASE/install/instance_name_ms.log*

At start-up, you can reset the name and location of the Cluster Edition error log file from the command line. Use the -e start-up parameter and value in the dataserver command to start the Cluster Edition. However, if you are using sybcluster to manage the cluster, you must use the sybcluster ‘set instance logpath’ parameter to change the location of the error log file for each instance.

**Note** Multiple instances cannot share the same error log.

## Setting the Cluster Edition error log path

**Note** The Cluster Edition installer does not create *RUN_server* files. However, if you create a *RUN_server* file, you can add a new location for the error log.

If you do not provide a path to the error log, the Cluster Edition adds an error log according to the:

- Cluster input file errorlog location. This information is stored in the quorum device and used by the dataserver command.
- Default location for error logs. If you do not supply an error log path, a log file named *instance_name.log* is created in the current working directory for dataserver.

You can change the log location stored on the quorum device using the qrmutil --errorlog parameter or with sybcluster 'set instance errorlog'.

If you are starting the cluster instances using shell scripts, change the value of the dataserver -e parameter, which overrides other settings for the error log location.

# Managing messages

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

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

  For the complete syntax for sp_addmessage and sp_altermessage, see the *Reference Manual*.
- Use configuration parameters to specify whether auditing events are logged. Auditing events pertain to a user's success, log audit logon success, or failure, log audit logon failure, in logging in to the Cluster Edition.

## Logging user-defined messages

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

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

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

### New messages

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

### Existing messages

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

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

## Logging auditing events

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

Possible parameters and values are:

- log audit logon success at 1 – to enable logging of successful Cluster Edition logins:

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

- log audit logon failure at 1 – to enable logging of unsuccessful Cluster Edition logins:

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

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

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

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

CHAPTER 19

# Setting Up Communications Across the Network

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



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

In the Sybase client/server environment, a client can connect with an instance if it knows where the server resides on the network and if the server supports the client's language or character set. When a client initiates a connection, it looks in its directory services for the network location of the target server.

Directory services list the name and address of every server, including Backup Server, Monitor Server, and XP Server. When you are using a client program, and you want to connect with a particular server, the client program looks up the server name in the directory services and connects to that server.

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

Table 19-1 shows where to find more information on server and client interfaces file tasks and topics.

***Table 19-1: Where to find interfaces file tasks and topics***

| Type of interfaces file | Task or topic | See |
|---|---|---|
| UNIX server or client | Adding entries for multiple Cluster Edition installations | *Adaptive Server Configuration Guide* |
| | Creating a master interfaces file for multiple installations | "Creating a master interfaces file" on page 422 |
| | Configuring for multiple networks | "Configuring interfaces files for multiple networks" on page 423. |
| | Reference information | "Understanding the format of the interfaces file" on page 419. |
| PC-client | Configuring a client | *Installation Guide* for your platform |
| | Reference information and instructions for advanced tasks | *Open Client and Open Server Programmer's Supplement* for your PC-client platform, or the appropriate Open Client documentation |
| Client platforms not listed | Configuring, reference information, and instructions for advanced tasks | *Open Client and Open Server Programmer's Supplement* for your PC-client platform, or the appropriate Open Client documentation |

# How the Cluster Edition determines which directory service entry to use

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

1 It looks for the server name supplied in the command line -s option. If the server name is not supplied in the command line:

2 It determines its own name by checking the value of the DSLISTEN environment variable. If the DSLISTEN environment variable is not set, then it assumes that the server name is SYBASE.

3 Looks in directory services for an entry whose name matches the name found in the steps above.

4 It uses the network information provided by the directory services entry it has found to listen for client connections.

## How a client uses directory services

When a client connects to a server it:

- Determines the name of the server either programmatically or by referring to the DSQUERY environment variable. If the application user has not set DSQUERY, the runtime value for the server name defaults to SYBASE.
- Looks in directory services for an entry whose name matches the name of the server.
- Uses the network information provided by the directory services entry to connect to the server. If the client cannot connect the first time, it makes additional attempts according to the delay and retry numbers indicated in directory services. If no matching entry is found, an error message is written to the client's standard error file. If multiple networks are supported, the client attempts to connect using the information in the second network address entry for the server.

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

## Creating a directory services entry

The sybcluster utility creates a directory services entry for the cluster and for each instance when you create the cluster. You can also use the following Sybase utilities to edit the network information in directory services:

- dsedit – a GUI utility.
- dscp – a UNIX command line utility.

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

# Supported directory drivers

There are three supported drivers for UNIX:

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

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

# Contents of an interfaces file

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

The network information in the file includes the server name, network name or address of the host machine, and the port, object, or socket number (depending on the network protocol) on which the server listens for queries. See "Understanding the format of the interfaces file" on page 419 for the specific makeup of the interfaces file entry.

Each entry in an interfaces file can contain two types of lines:

- Master lines, which are used by server applications to listen for queries over the network. This information is called a *listener service*.
- Query lines, which are used by client applications to connect to servers over the network. This information is called a *query service.*

The network information contained in the master and query lines for a server may be identical because a server listens for connection requests on the same port that clients use to request connections.

A server needs master lines in its *interfaces* file. When servers act as clients to other servers, query lines are required for those servers.

A client's interfaces file does not need a master line. It functions correctly with only a query line.

If your site has multiple installations

If you have more than one Adaptive Server or Cluster Edition installation, each server's interfaces file should contain information about all servers on the network.

If all of your server products are running on the same platform, you can create one master *interfaces* file and copy that file to each machine. For more information, see "Creating a master interfaces file" on page 422.

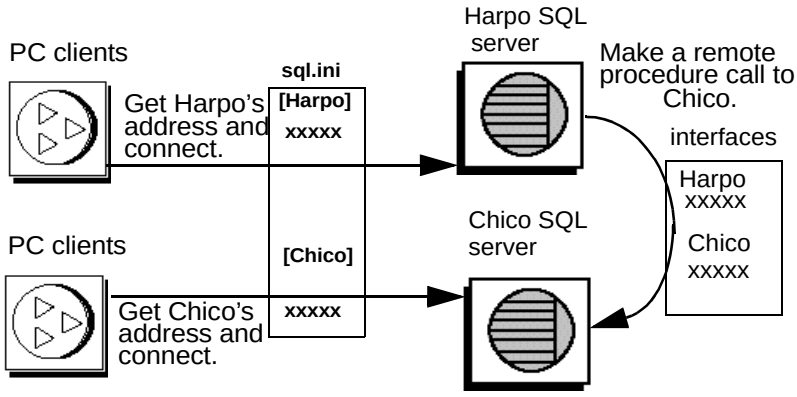
If the host machine supports multiple networks, see "Configuring interfaces files for multiple networks" on page 423.

## Heterogeneous and homogeneous environments

You can run the Cluster Edition and clients on the same platform or on different platforms.

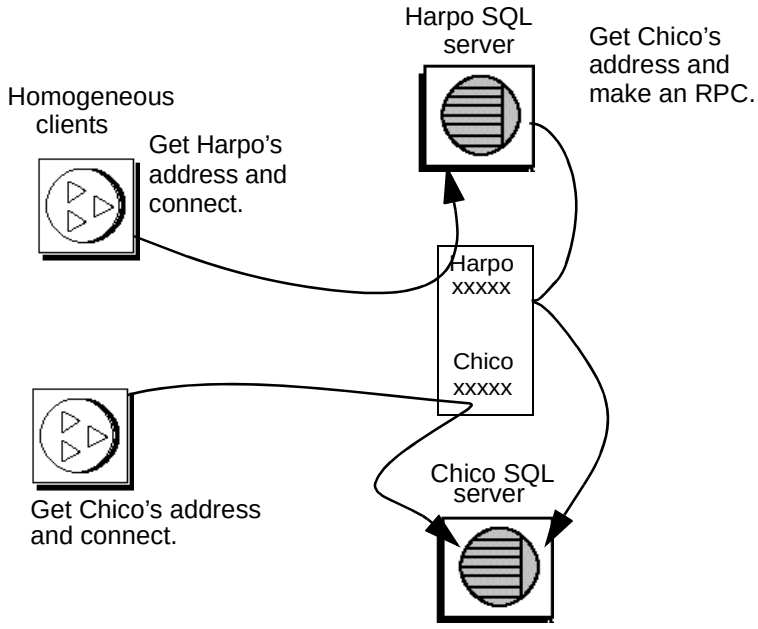
If the platforms are different, each platform may require a different format and configuration for its *interfaces* file. Figure 19-1 illustrates how a PC client uses network information in its interfaces file (*sql.ini*) to connect to an instance running on UNIX, and how an instance uses its *interfaces* file to connect to another server during a remote procedure call.

***Figure 19-1: Establishing network connections in a heterogeneous environment***




If both a client and a server are running under UNIX, the same interfaces file is valid for both. Figure 19-2 illustrates how clients and instances running in a homogeneous environment can use copies of the interfaces file to establish connections. Because the two instances are running under the same operating system, they can use the same interfaces file or exact copies of the same file.

***Figure 19-2: Establishing network connections in a homogeneous environment***




# Understanding the format of the *interfaces* file

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

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

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

A TLI style entry looks like:

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

A TCP style entry looks like:

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

## Components of an interfaces file entry

Table 19-2 describes the components of an interfaces file entry.

***Table 19-2: Components of an interfaces file***

| Component | Value |
|---|---|
| *servername* | Name of instance or Backup Server. Requirements for a server name are:<br>• The name cannot be more than 30 characters long.<br>• The first character must be a letter (ASCII a through z, A through Z).<br>• The characters that follow must be letters, numbers, or underscores (_). |
| *retry_attempts* (optional) | Number of times you want the client to try to connect to a server after initial failure. Default is 0. |
| *delay_interval* (optional) | Number of seconds between connection attempts. Default is 0. |
| *service_type* | Type of service defined by entry. Must be one of the following:<br>• master<br>• query |
| *api* | Application programming interface available to the network.The supported value is tli. |
| *protocol* | Name of the network protocol. Protocol types available are:<br>• TCP/IP, represented by the letters “tcp” |
| *network* | Name of the network; not currently used by the Cluster Edition. sybcluster enters “ether” as a placeholder. |
| *host* | Network name or address of server’s host machine.<br>• For TCP/IP, use either the host name or Internet address. Maximum size of entry is 32 bytes.<br>To determine the host name of a machine, log in to that machine and enter:<br>`/bin/hostname` |

| Component | Value |
|---|---|
| *machine* | Network name or address of server's host machine.<br>You can use either the host name or Internet address. Maximum size of entry is 32 bytes.<br>To determine the host name of a machine, log in to that machine and enter:<br>`/bin/hostname` |
| *device* | The network device endpoint.<br>For TCP networks, the device varies according to the vendor supplying the networking software. Check the vendor-specific documentation for the name of the device. Your network may provide multiple stream devices corresponding to different protocols in the TCP protocol suite. Choose the TCP streams device. Common TCP streams devices is */dev/tcp*. |
| *address* for a TLI protocol entry | Address consists of:<br>• Address prefix, "\x" for TLI.<br>• Network type, always 0002.<br>• Port number converted to four digits, hexadecimal. Must be a unique number between 1025 and 65535. Check the */etc/services* file on each machine on the network to see what port numbers are in use. Enter the instance port number in a new section of */etc/services* labeled "Sybase specific services." You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using the port number.<br>• IP network node address of the host machine converted to 8 digits, hexadecimal.<br>• Trailing zeros, optional, 16 digits. |
| *port* | A unique port number between 1025 and 65535. Check the /etc/services file on each machine on the network to see what port numbers are in use. Enter the instance port number in a new section of /etc/services labeled "Sybase specific services." You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using that port number. |
| *ha_failover* | An entry created in directory services or the interfaces file for high availability. |
| *filter* | The Cluster Edition supports Secure Socket Layers (SSL) as a filter which is appended to the master and query lines of the directory services. SSL is the standard for securing the transmission of sensitive information. For more information about SSL, see "Using SSL in a clustered environment" on page 37. |

# Creating a master *interfaces* file

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

Distributing copies of one interfaces file (a master file) with entries for all instances is the easiest way to maintain consistency in the interfaces files in a homogeneous environment on a network.

You can make all changes to one version of the file and then copy the updated master file to all appropriate Sybase directories.

You can make a master file in one of two ways:

- Using dsedit or dscp
- Using a text editor

## Using dsedit or dscp to create a master interfaces file

You can use the dsedit or dscp utility to create a master interfaces file, which you can then distribute to all servers. If you are not an experienced Sybase user, you may find that using dsedit or dscp is easier than using a text editor. Using dsedit or dscp also ensures that your interfaces file is consistent in format.

To create a master interfaces file with dsedit or dscp:

1 Select the interfaces file that contains the most complete, up-to-date information.

2 Begin a dsedit or dscp session in your latest Sybase installation to edit this interfaces file.

3 Add entries for any instances or Backup Servers that are not listed in this file.

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

## Using a text editor to create a master interfaces file

To construct a single master *interfaces* file from several individual interfaces files:

1 Concatenate the individual interfaces files.

2 Make a copy of the file.

3 Use an ASCII text editor to modify the copy of the concatenated file.

**Note** When you manually edit an *interfaces* file, be sure that, for each entry, each line following the first line begins with a <tab> character.

The following elements must be correct and unique in the resulting file:

- *servername* – each server entry in the *interfaces* file must be unique. The server name entries for each cluster, and for each instance within the cluster, must be unique within the interfaces file.
- A combination of the host machine's network name or address and instance's port or object number.
- If the original *interfaces* file was created when there was only one machine on the network, its entries may have the word "loghost" in place of the machine name (address). If *loghost* is present, replace it with the machine name.

# Configuring interfaces files for multiple networks

On some platforms, the Cluster Edition can accommodate multiple networks. This allows instances to listen for clients over multiple network interfaces. You must add an entry for each network interface to the interfaces file.

## Configuring the server for multiple network handlers

To configure multiple network listeners:

1 Define a unique host name for each network interface in your operating system's host database.

2 In your interfaces file, use a text editor to add copies of the "master" line for your instance; one for each additional interface you want the server to listen on.

3 Include a unique host name on each line to configure a network handler for each network interface.

4 Port numbers within the interface need not be the same, but they can be. They fall under the same rules for naming and numeric range as the primary network interface.

### Sample interfaces files for multiple network handlers

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

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

When the instance restarts, it spawns a network handler process for each master line in the entry that corresponds to the server's DSLISTEN value. Connections made on each interface are handled equally, as peers.

## Configuring the client connections

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

- Use the same DSQUERY name for all clients. The interfaces files on the different machines contain different network names.
- Use different DSQUERY names for the clients. The interfaces files on all the machines are the same, but they contain multiple DSQUERY names.

### Using one network-independent DSQUERY name

If uniform client DSQUERY naming is important, you can make the necessary changes in the network addressing of the clients in the interfaces file. You can install separate Sybase installation directories and distinct interfaces files on client file servers on each network to allow users to connect to the correct network address. Instead of altering the DSQUERY name the clients use, you maintain one DSQUERY name for all clients, on all networks, and alter each network's interfaces file accordingly.

This method assumes that:

- You have complete control over what the Sybase installation clients see on each network.
- The interfaces file (at the very least) is *not* shared or copied among Sybase installations on different networks.

The interfaces file looks like the following example on the "engineering" network:

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_ENGNET 5470
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

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

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

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

The full "master" entry is present in both files. This is allowed because only the instance will use the "master" lines. Assuming that the server host machine can see both networks (both host names are interchangeable), it does not matter which interfaces file is used for instance start-up.

### Using different DSQUERY names

To use different DSQUERY names for each network listener:

1 Choose an additional server name.

  You can concatenate the original server name and the network name. For example, if your server is named PRODUCTION, you could choose the names PRODUCTION_network1 and PRODUCTION_network2.

2 Do one of the following:

  - For PC clients, use sqledit to create multiple *sql.ini* file entries for the server, one for each network. In the following example, you create one entry for PRODUCTION_network 1 and one for PRODUCTION_network2. For more information, see the Open Client documentation for your client platform.

- For UNIX clients, you can edit the *interfaces* files with an ASCII text editor. From the server's *interfaces* files, copy the server name line and the "master" line for each network into the client interfaces file. Add the appropriate server name for each entry, and change "master" to "query."

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

```
# Client entry for PRODUCTION on network1
PRODUCTION_network1<tab>3<tab>3<newline>
<tab>query tcp ether serv_corpnet 4559
# Client entry for PRODUCTION on network2
PRODUCTION_network2<tab>3<tab>3<newline>
<tab>query tcp ether serv_engnet 5479
```

## Configuring for query port backup

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

To configure an instance for query port backup:

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

   The following example shows how to configure a backup network that will be used only if the normal connection fails. The primary network is "corporate network" and backup is "engineering network."

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

3 Configure PC-client interfaces files with the appropriate multiple "query" entries, as described in the Open Client documentation. For client interfaces files in a homogeneous environment, you can copy the entire interfaces file entry for the instance into the client interfaces file.

4 A connection on the secondary port occurs only if the corporate network is disabled, or if the corporate network interface on the host machine fails or is shut down due to a network-related failure.

# IPv6 support

The Cluster Edition supports IPv6 technology.

**Note** The HPIA64 platform does not support IPv6 for Adaptive Server CE 15.0.1.ESD4.

## Understanding IPv6

IPv6 addressing terminology:

- Link-local address – an IPv6 address that is usable only over a single link.
- Site-local address – an IPv6 address that can be used within a single-site.
- Global address – an IPv6 address that can be used across the global Internet.

IPv6 application types:

- IPv6-unaware – an application that cannot handle IPv6 addresses.
- IPv6-aware – an application that can communicate with nodes that do not have IPv4 addresses. In some cases, this might be transparent to the application, for instance when the API hides the content and format of the actual addresses.
- IPv6-enabled – an application that, in addition to being IPv6-aware, takes advantage of some IPv6 features.
- IPv6-required – an application that requires some IPv6 features and cannot operate over IPv4.

IPv6 Infrastructure:

## IPv6 infrastructure

Dual Stack infrastructure implements both IPv4 and IPv6. This is the recommended infrastructure implementation for using the Cluster Edition as an IPv6-aware server.

Sybase applications are IPv6-aware. All code to turn the Cluster Edition and the Open Client/Server components IPv6-aware was done using the IETF designed primitives, see "Creating or converting for IPv6-aware applications." The following matrix lists the platform run-time requirements and the specific product and its release version:

***Table 19-3: IPv6 support***

| Platform | the Cluster Edition IPv6 awareness | Open Client/Server IPv6 awareness |
|---|---|---|
| Sun Solaris 8 32- and 64-bit | 12.5.3a and 15.0 | 12.5 and 15.0 |
| HP-UX 11i(v1) 32- and 64-bit | 12.5.3a and 15.0 | 12.5 and 15.0 |
| Microsoft Server 2003 | 12.5.3a and 15.0 | 12.5 and 15.0 |
| Linux RHEL 3.0 | 15.0 | 12.5 and 15.0 |

Many Sybase products that are Open Client/Server based like XP Server, Backup Server, Replication Server and Open Switch became automatically IPv6-aware due to the layered Open Client Transport Control Layer (CTlib->NETlib) which is IPv6-aware for network-socket operations. An important note is that any DBlib based Open Client product is not IPv6-aware.

For the Cluster Edition, being IPv6-aware is a complex issue because some components within the server are 3rd party components and are not yet IPv6-aware. To understand how this impacts the Cluster Edition, the following list shows all functional mechanisms of the Cluster Edition that are IPv6-aware with respect to the platform / release matrix above:

- Connection Handler
- RPC mechanisms
- Job Scheduler Task / Agent session connection
- Network Host API
- UDP Message support for sybsendmsg

- Component Integration Services connectivity
- Host / name resolving
- XML URL connection handler
- Auditing for client address data

The following functional mechanisms in the Cluster Edition do not support IPv6. These mechanisms in the Cluster Edition are IPv6-unaware:

- Java support
- License Management Server
- LDAP driver
- Private interconnect among various instances in cluster.

## Starting the Cluster Edition as IPv6-aware

The Cluster Edition is IPv6-unaware, by default. You must start the Cluster Edition with trace flag 7841 to make it IPv6-aware. This causes the Cluster Edition to determine IPv6 availability and makes the Cluster Edition IPv6-aware.

See your Network or IT specialist to configure your platforms and Network Infrastructure correctly for IPv6 support.

A second trace flag, 7815 can be set when you start the Cluster Edition which captures and logs address connection requests and host / name lookups.

The IPv6 Cluster Edition traceflags:

- T7841 – Enable the Cluster Edition IPv6-awareness
- T7815 – Report all the Cluster Edition IPv4 & IPv6 Client address connect requests

Before starting the Cluster Edition for IPv6-aware operations, make sure that your infrastructure is correctly set up. Once your operating system is correctly configured, an IPv6 connection handler can be configured and enabled. Configuring and enabling the IPv6 connection handler requires adding an additional DCL entry. A single Cluster Edition configuration can typically carry up to 32 connection handler assignments within the DCL.

For example if you have a Site-local setup with two domains administrated under the nameserver setup:

```
sybase.com - being responsible for all IPv4 networking applications
v6.sybase.com - being responsible for all IPv6 networking applications
```

The DCL entry for the Cluster Edition to start named “SYBASE” on the host “revival” for port 17100 would typically look like:

```
SYBASE
master tcp ether revival.sybase.com 17100
query tcp ether revival.sybase.com 17100
master tcp ether revival.v6.sybase.com 17100
query tcp ether revival.v6.sybase.com 17100
```

In the above example, when the Cluster Edition is started with IPv6-awareness it creates two connection handlers. One listens on port 17100 for incoming IPv4 Clients connection requests, and the other listens on port 17100 for incoming IPv6 Clients connection requests.

# Troubleshooting

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

## Server fails to start

If a server fails to start with the following message, the port number specified in the interfaces file may be in use:

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

❖ **Investigating the port assignment**

1 Look in the interfaces file to identify the port number assigned to the server.

2 Determine whether another process is using the same port number by entering:

```
netstat -a
```

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

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

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

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

❖ **If a stale server process is retaining use of the port number**

1 Do one of the following:

- Use the operating system kill command to terminate the process.
- Use another port number for the server by modifying the interfaces file.

2 Start the server manually to confirm that the port number is available.

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

## Error when executing an ESP

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

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

XP Server cannot start because the port number may be in use by another process. Use the netstat command described in the previous section to determine if the port number specified for XP Server is in use.

If you find no processes using the same port number, execute the ESP you attempted earlier. XP Server should start automatically.

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

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

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

# Glossary

**action** — A user-executed command that stops one or more instances at a specified time to initiate a planned failover, downtime, or other administrative task. An action changes the state of an instance. See Chapter 5, “Managing the Workload.”

**base instance** — An instance assigned to a logical cluster on which a logical cluster normally runs. See Chapter 5, “Managing the Workload.”

**client migration** — The movement of an established client connection from one instance to another. The client is migrated from the old to the new instance without the client application being aware of the migration. Client migration is used for dynamic load distribution and for administrative actions such as logical cluster failback. See Chapter 2, “Client Applications and Client/Server Interaction,” for a complete description.

**cluster** — A collection of homogeneous nodes in a network that operate as a single system. Each node has its own CPU and memory. All nodes communicate with each other through private and high-speed communication pathways.

**cluster lock manager (CLM)** — The server module that provides distributed locking services for the cluster. The CLM enables sharing of buffers, global objects, and metadata among the instances.

**clusterware** — Sybase software included in Adaptive Server that enables the shared-disk cluster.

**dynamic load distribution** — The migration of an established client connection to a different instance in an attempt to balance the workload within a logical cluster.

**extended high-availability (HA) failover** — Adaptive Server provides a list of failover addresses to high-availability-aware clients when they connect. This allows multiple clients to fail over eliminates the need for the “HAFAILOVER” entry in the directory services or interfaces file.

**failover** — The ability to switch automatically to another instance upon the failure or abnormal termination of a previously active node.

**failover group** — A set of failover instances defined for a logical cluster. Failover groups let you specify preference and order for failover instances. See Chapter 5, “Managing the Workload.”

**failover instance** — An instance on which a logical cluster can run if one or more of its base instances fail. See Chapter 5, “Managing the Workload.”

**instance** — An Adaptive Server that participates in a shared-disk cluster.

**instance number** — A number that uniquely identifies a named instance in the Adaptive Server shared-disk cluster.

**instance state** — The state of an instance in a logical cluster as it is perceived by a logical cluster. Thus, an instance can be physically online, but offline to a given logical cluster. See Chapter 5, “Managing the Workload.”

**load profile** — A set of weighted metrics used to determine the relative workload on an instance in a logical cluster. You can create your own load profiles or use one of the profiles provided by Sybase. See Chapter 5, “Managing the Workload.”

**load score** — A computed value of the overall load on an instance; a unitless number that can be used to compare relative workloads on different instances in a logical cluster, or on the same instance at different times. See Chapter 5, “Managing the Workload.”

**local system temporary database** — Space for temporary tables and worktables. Each instance in the cluster has a local system temporary database that it alone can access.

**logical cluster** — A method of abstracting the physical cluster so that multiple application services can be established. A logical cluster supports fine-tuned management of the workload within the cluster by enabling application- or user-specific service level agreements, resource assignments, and failover rules. Applications connect directly to a logical cluster. See Chapter 5, “Managing the Workload,” for more information.

**login redirection** — The mechanism by which an instance can direct an incoming client connection to a different instance in the cluster. Login redirection is used to route inbound connections to instances in a logical cluster and for load balancing. See Chapter 2, “Client Applications and Client/ Server Interaction,” for a complete description.

**node** — A machine (hardware) that hosts an Adaptive Server instance.

**open logical cluster** A logical cluster that accepts connections that have no defined route. By default, the system logical cluster has the open property, but you can grant the open property to another logical cluster. Only one logical cluster can have the open property at a time. See Chapter 5, “Managing the Workload,” for more information.

**physical cluster** The shared-disk cluster, with a specific quorum disk, member instances, and interconnection information. All instances in the physical cluster have direct access to a single installation of the databases and are monitored and managed by the cluster membership service.

**quorum device** This device provides important information that defines the cluster, including the name of the cluster, the names of the instances in the cluster, the number of nodes, and their names. In addition, the quorum device holds state information about the instances in the cluster and defines cluster membership

**resource reservation** The practice of setting aside an instance for a specific logical cluster and only allowing clients routed to that logical cluster to connect to it. To practice resource reservation, you must assign the open property to a logical cluster other than the system logical cluster. See Chapter 5, “Managing the Workload,” for more information.

**shared-disk cluster** A cluster configuration where all instances have direct access to all data on all shared disks. In the Cluster Edition, all instances have direct access to database devices and jointly manage the single installation of the databases.

**symmetric multiprocessing (SMP) system** Is comprised of multiple CPUs and a single RAM memory with a single operating system. The CPUs symmetrically serve and run all functionality of the operating system and applications. This is the nonclustered Adaptive Server environment.

**system logical cluster** A logical representation of the physical cluster. The system logical cluster is automatically created when the physical cluster is created, and it has the same name as the physical cluster. All background tasks run on the system logical cluster. See Chapter 5, “Managing the Workload,” for more information.

**workload manager** An Adaptive Server module that provides application-level management of resource allocation, availability, and load distribution.

# Index

## Symbols



## A

## B



## C

## D

## E



## F

## G



## H



## I

## J



## K



## L

## M

## N



## O



## P

## Q



## R



## S

## T



## U

## V



## W