Oracle® Solaris Cluster Data Service for Sybase ASE Guide

Part No: E39577
July 2014, E39577–01

ORACLE

Contents

Using This Documentation 7

1 Installing and Configuring HA for Sybase ASE 9
HA for Sybase ASE Overview 9
Overview of Installing and Configuring HA for Sybase ASE 10
Preparing to Install HA for Sybase ASE 10
Installing the Sybase ASE Software 11
▼ How to Prepare the Nodes for the Installation of the Sybase ASE Software 11
▼ How to Install the Sybase ASE Software 12
▼ How to Verify the Sybase ASE Installation 14
Configuring Sybase ASE Database Access and Creating the Sybase ASE Database Environment 15
▼ How to Configure Sybase ASE Database Access With Solaris Volume Manager 15
▼ How to Create the Sybase ASE Database Environment 16
Installing the HA for Sybase ASE Package 18
▼ How to Install the HA for Sybase ASE Package 18
Registering and Configuring HA for Sybase ASE 19
Setting HA for Sybase ASE Extension Properties 19
▼ How to Register and Configure HA for Sybase ASE 20
Verifying the HA for Sybase ASE Installation and Configuration 23
▼ How to Verify the HA for Sybase ASE Installation and Configuration 23
Location of HA for Sybase ASE Log Files 24
HA for Sybase ASE Logging and Security Issues 24
HA for Sybase ASE Logging Issues 25
HA for Sybase ASE Security Issues 25
Tuning the HA for Sybase ASE Fault Monitor 25
Main Fault-Monitor Process 26
Database-Client Fault Probe 27
Obtaining Core Files for Troubleshooting 27

Customizing the HA for Sybase ASE Fault Monitor 27
Defining Custom Behavior for Errors 28
Propagating a Custom Action File to All Nodes in a Cluster 36
Specifying the Custom Action File That a Server Fault Monitor Should Use 36

A HA for Sybase ASE Extension Properties 39

Index 45

Using This Documentation

- **Overview** –Provides procedures to install and configure the Oracle Solaris Cluster HA for Sybase Adaptive Server Enterprise (ASE) (HA for Sybase ASE) data service.
- **Audience** – Experienced system administrators with extensive knowledge of Oracle software and hardware.
- **Required knowledge** – Knowledge of the Oracle Solaris operating system, of Oracle Solaris Cluster software, and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

This document is not to be used as a planning or presales guide.

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E39579.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program web site at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Installing and Configuring HA for Sybase ASE

This chapter explains how to configure and administer HA for Sybase ASE on your Oracle Solaris Cluster nodes.

This chapter contains the following sections.

- "HA for Sybase ASE Overview" on page 9
- "Overview of Installing and Configuring HA for Sybase ASE" on page 10
- "Preparing to Install HA for Sybase ASE" on page 10
- "Installing the Sybase ASE Software" on page 11
- "Configuring Sybase ASE Database Access and Creating the Sybase ASE Database Environment" on page 15
- "Installing the HA for Sybase ASE Package" on page 18
- "Registering and Configuring HA for Sybase ASE" on page 19
- "Verifying the HA for Sybase ASE Installation and Configuration" on page 23
- "HA for Sybase ASE Logging and Security Issues" on page 24
- "Tuning the HA for Sybase ASE Fault Monitor" on page 25
- "Customizing the HA for Sybase ASE Fault Monitor" on page 27

HA for Sybase ASE Overview

HA for Sybase ASE provides fault monitoring and automatic failover for the Sybase ASE application.

You must configure HA for Sybase ASE as a failover data service.

For general information about data services, resource groups, resources, and other related topics, see:

- "Oracle Solaris Cluster Concepts Guide "
- Chapter 1, "Planning for Oracle Solaris Cluster Data Services," in "Oracle Solaris Cluster Data Services Planning and Administration Guide "

Overview of Installing and Configuring HA for Sybase ASE

The following table summarizes the tasks for installing and configuring HA for Sybase ASE and provides cross-references to detailed instructions for performing these tasks. Perform the tasks in the order that they are listed in the table.

TABLE 1-1 Tasks for Installing and Configuring HA for Sybase ASE

Task	Instructions
Prepare to install HA for Sybase ASE	"Preparing to Install HA for Sybase ASE" on page 10
Install the Sybase ASE software	"Installing the Sybase ASE Software" on page 11
Configure Sybase ASE database access and create the Sybase ASE database environment	"Configuring Sybase ASE Database Access and Creating the Sybase ASE Database Environment" on page 15
Install the HA for Sybase ASE package	"Installing the HA for Sybase ASE Package" on page 18
Register HA for Sybase ASE resource types and configure resource groups and resources	"Registering and Configuring HA for Sybase ASE" on page 19
Verify the HA for Sybase ASE installation	"Verifying the HA for Sybase ASE Installation and Configuration" on page 23
Tune the HA for Sybase ASE fault monitor	"Tuning the HA for Sybase ASE Fault Monitor" on page 25

Preparing to Install HA for Sybase ASE

To prepare your nodes for the HA for Sybase ASE installation, select an installation location for the following files.

- **Sybase ASE application files.** These files include Sybase ASE binaries and libraries. You can install these files on either the local file system or the cluster file system.

  For information about the advantages and disadvantages of placing the Sybase ASE binaries on the local file system instead of the cluster file system, see "Configuration Guidelines for Oracle Solaris Cluster Data Services" in "Oracle Solaris Cluster Data Services Planning and Administration Guide ".
- **Sybase ASE configuration files.** These files include the `interfaces` file, `config` file, and environment file. You can install these files on the local file system (with links), the highly available local file system, or on the cluster file system.
- **Database data files.** These files include Sybase device files. You must install these files on the highly available local file system or the cluster file system as either raw devices or regular files.

Installing the Sybase ASE Software

Use the procedures in this section to complete the following tasks.

- Preparing the nodes for the installation of the Sybase ASE Software
- Installing the Sybase ASE software
- Verifying the Sybase ASE installation

Note - Before you configure HA for Sybase ASE, use the procedures that the "Oracle Solaris Cluster Software Installation Guide " describes to configure the Oracle Solaris Cluster software on each node.

- "How to Prepare the Nodes for the Installation of the Sybase ASE Software" on page 11
- "How to Install the Sybase ASE Software" on page 12
- "How to Verify the Sybase ASE Installation" on page 14

Note - The Oracle Solaris Cluster HA for Sybase ASE software can be configured to run in a zone cluster.

▼ How to Prepare the Nodes for the Installation of the Sybase ASE Software

Caution - Perform all of the steps in this procedure on all of the nodes. If you do not perform all of the steps on all of the nodes, the Sybase ASE installation will be incomplete, and HA for Sybase ASE will fail during startup.

Note - Consult the Sybase ASE documentation before you perform this procedure.

---

1. **Assume the `root` role on all of the nodes.**

2. **Configure the `/etc/nsswitch.conf` file as follows so that HA for Sybase ASE starts and stops correctly if a switchover or failover occurs.**
   On each node that can master the logical host that runs HA for Sybase ASE, include the following entries in the `/etc/nsswitch.conf` file.

```
passwd:    files dns
publickey: files dns
project:   files dns
group:     files dns
```

   HA for Sybase ASE uses the `su` *user* command to start and stop the database node.

The network information name service might become unavailable when a cluster node's public network fails. Adding the preceding entries ensures that the `su`(1M) command does not refer to the NIS/NIS+ name services if the network information name service is unavailable.

3. **Configure the cluster file system for HA for Sybase ASE.**

   If raw devices contain the databases, configure the global devices for raw-device access. For information about how to configure global devices, see "Oracle Solaris Cluster Software Installation Guide ".

   If you use the Solaris Volume Manager software, configure the Sybase ASE software to use UNIX file system (UFS) logging on mirrored metadevices or raw-mirrored metadevices. For information about how to configure raw-mirrored metadevices, see "Solaris Volume Manager Administration Guide ".

4. **Prepare the `SYBASE_HOME` directory on a local or multihost disk.**

   Note - If you install the Sybase ASE binaries on a local disk, use a separate disk if possible. Installing the Sybase ASE binaries on a separate disk prevents the binaries from being overwritten during reinstallation of the operating system.

5. **On each node, create an entry for the database administrator (DBA) group in the `/etc/group` file, and add potential users to the group.**

   Verify that the `root` and *sybase* users are members of the *dba* group, and add entries as necessary for other DBA users. Ensure that group IDs are the same on all of the nodes or zones that run HA for Sybase ASE, as the following example illustrates.

   dba`:*:520:root,`*sybase*

   You can create group entries in a network name service. If you create entries this way, also add your entries to the local `/etc/group` file to eliminate dependency on the network name service.

6. **On each node, create an entry for the Sybase system administrator.**

   The following command updates the `/etc/passwd` and `/etc/shadow` files with an entry for the Sybase system administrator.

   `# useradd -u 120 -g` *dba* **`-d /`***Sybase-home* *sybase*

   Ensure that the *sybase* user entry is the same on all of the nodes that run HA for Sybase ASE.

▼ How to Install the Sybase ASE Software

Before You Begin Ensure that the `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.

1. **On a cluster node, assume the `root` role or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2. **If you plan to install the Sybase ASE software on the cluster file system, start the Oracle Solaris Cluster software and become the owner of the disk device group.**

   If you plan to install the Sybase ASE software at another location, omit this step.

   For more information about installation locations, see "Preparing to Install HA for Sybase ASE" on page 10.

3. **Create a failover resource group to hold the network and application resources.**

   ```
   # clresourcegroup create [-n nodelist] resource-group
   ```

`-n` *nodelist*	Specifies an optional, comma-separated list of node names or IDs that identify potential masters. The order here determines the order in which the Resource Groups Manager (RGM) considers primary nodes during failover.
resource-group	Specifies the name of the resource group. This name can be your choice but must be unique for resource groups within the cluster.

   Note - Use the `-n` option to specify the order of the node list. If all of the nodes in the cluster are potential masters, you do not need to use the `-n` option.

4. **Verify that you have added all of the network resources that HA for Sybase ASE uses to either the `/etc/inet/hosts` file or to your name service (NIS, NIS+) database.**

5. **Add a network resource (logical host name or shared address) to the failover resource group.**

   ```
   # clreslogicalhostname create -g resource-group -h host_list logical_host
   ```

logical_host	Specifies a resource name of your choice.
host_list	Specifies a comma-separated list of host names that are to be made available by the logical hostname resource.

6. **Run the `clresourcegroup(1CL)` command to complete the following tasks.**

   - Enabling the resource and fault monitoring
   - Moving the resource group into a managed state
   - Bringing the resource group online

   ```
   # clresourcegroup online -M resource-group
   ```

7. **On the node that masters the resource group that you just created, log in as *sybase*.**

   The installation of the Sybase binaries must be performed on the node where the corresponding logical host is running.

8. **Install the Sybase ASE software.**

   Regardless of where you install the Sybase ASE software, modify each node's `/etc/system` files as you would in standard Sybase ASE installation procedures. For instructions about how to install the Sybase ASE software, refer to the Sybase installation and configuration guides.

9. **For every Sybase ASE server, specify the host name that is associated with a network resource.**

   If you do not specify a host name that is associated with a network resource, Sybase ASE starts *only* on the node where the Sybase ASE software was installed.

   Some versions of Sybase ASE prompt you for the host name. Other versions of Sybase ASE use the physical host name. If your version of Sybase ASE uses the physical host name, you must change the physical host name to specify a network resource.

   - If the Sybase ASE installer prompts you for the host name, type the host name in response to the prompt.
   - Otherwise, modify the `interfaces` file to change the physical host name to a host name that is associated with a network resource.

   Note - For a licensed version of Sybase ASE, the host name that you specify must be associated with a valid license file.

Next Steps After you install the Sybase ASE software, go to "How to Verify the Sybase ASE Installation" on page 14.

▼ How to Verify the Sybase ASE Installation

- **Verify that the *sybase* user and the *dba* group own the `$SYBASE_HOME` directory and `$SYBASE_HOME` children directories.**

Next Steps After you verify the Sybase ASE installation, go to "Configuring Sybase ASE Database Access and Creating the Sybase ASE Database Environment" on page 15.

Configuring Sybase ASE Database Access and Creating the Sybase ASE Database Environment

Configuring Sybase ASE database access and creating the Sybase ASE Database Environment involves the following tasks.

1. Configuring Sybase ASE database access with Solaris Volume Manager. See "How to Configure Sybase ASE Database Access With Solaris Volume Manager" on page 15.
2. Creating the Sybase ASE database environment. See "How to Create the Sybase ASE Database Environment" on page 16.

▼ How to Configure Sybase ASE Database Access With Solaris Volume Manager

1. Configure the disk devices for the Solaris Volume Manager software to use.

For information about how to configure Solaris Volume Manager, see "Oracle Solaris Cluster Software Installation Guide ".

2. If you use raw devices to contain the databases, run the following commands to change each raw-mirrored metadevice's owner, group, and mode.

If you do not use raw devices, do not perform this step.

a. If you create raw devices, run the following commands for each device on *each node* that can master the Sybase ASE resource group.

```
# chown sybase /dev/md/metaset/rdsk/dn
# chgrp dba /dev/md/metaset/rdsk/dn
# chmod 600 /dev/md/metaset/rdsk/dn
```

metaset	Specifies the name of the disk set
`/rdsk/dn`	Specifies the name of the raw disk device within the *metaset* disk set

b. Verify that the changes are effective.

```
# ls -lL /dev/md/metaset/rdsk/dn
```

Next Steps Go to "How to Create the Sybase ASE Database Environment" on page 16.

How to Create the Sybase ASE Database Environment

The Sybase ASE database environment consists of the following servers:

- Data server
- Backup server
- Monitor server
- Text server
- XP server

Creating the Sybase ASE database environment involves configuring these servers.

Before You Begin Ensure that you have completed the following tasks.

- Establish a highly available IP address and name, that is, a network resource that operates at installation time.
- Locate device paths for all of the Sybase ASE devices—including the master device and system devices—in the highly available local file system or cluster file system. Configure device paths as one of the following file types.
  - Regular files
  - Raw devices
  - Files that the Solaris Volume Manager software manages
- Locate the Sybase ASE server logs in either the cluster file system or the local file system.
- Ensure that the password for the Sybase ASE system administrator account is correctly specified.

  HA for Sybase ASE must be able to start and monitor the monitor server. To meet this requirement, ensure that the password for the Sybase ASE system administrator account (`sa`) is specified in the file `RUN_`*monitor-server*, where *monitor-server* is the name of the Sybase ASE monitor server. This name is defined during the Sybase ASE installation. For more information, see your Sybase ASE documentation.

  If you do not set the required password in the `RUN_`*monitor-server* file, the HA for Sybase ASE data service cannot start the monitor service. If no password is set for the `sa` account, you do not need to modify the `RUN_`*monitor-server* file.
- Create an `interfaces` file for the cluster.

  The entire cluster must contain only one copy of the `interfaces` file. The `$SYBASE` directory contains the `interfaces` file. If you plan to maintain per-node file copies, ensure the file contents are identical.

  All of the clients that connect to Sybase ASE servers connect with Sybase OpenClient libraries and utilities. When you configure the Sybase ASE software, in the `interfaces` file,

enter information about the network resource and various ports. All of the clients use this connection information to connect to the Sybase ASE servers.

1. **Run the GUI-based utility `srvbuild` to create the Sybase ASE database.**

   This utility is contained in the `$SYBASE/$SYBASE_ASE/bin` directory.

   For information about the `srvbuild` utility, see your Sybase ASE installation documentation for Oracle Solaris.

2. **To verify successful database installation, ensure that all of the servers start correctly.**

   Run the `ps` command to verify the operation of all of the servers. Sybase ASE server logs indicate any errors that have occurred.

3. **Set the password for the Sybase ASE system administrator account.**

   For details about changing the `sa` login password, see your Sybase ASE system administration documentation.

4. **Create a new Sybase ASE account for fault monitoring.**

   This account enables the fault monitor to perform the following tasks.

   - Supporting queries to system tables
   - Creating and updating user tables

   Note - Do not use the `sa` account for these purposes.

   The following example shows how to create a new Sybase ASE account for fault monitoring.

```
# isql -Usa -Psybase -Sasedb
1> use master
2> go
1> create database sc3xdb
2>go
1> sp_addlogin dbmon, dbmonp, sc3xdb
2> go
1> use sc3xdb
2> go
1> sp_changedbowner dbmon
2> go
1> sp_modifylogin dbmon, defdb, sc3xdb
2> go
1> exit
```

   For more information, see “Tuning the HA for Sybase ASE Fault Monitor” on page 25.

5. **Update the stop file with the `sa` password.**

Because the stop file contains the `sa` password, protect the file with the appropriate permissions, and place the file in a directory that the system administrator chooses. Enable only the *sybase* user to read, write, and execute the stop file.

For more information about the stop file, see "HA for Sybase ASE Security Issues" on page 25.

Next Steps After you create the Sybase ASE database environment, go to "Installing the HA for Sybase ASE Package" on page 18.

Installing the HA for Sybase ASE Package

If you did not install the HA for Sybase ASE package during your initial Oracle Solaris Cluster installation, perform this procedure to install the package.

▼ How to Install the HA for Sybase ASE Package

Perform this procedure on each cluster node where you want the HA for Sybase ASE software to run.

1. **On the cluster node where you are installing the data service package, assume the `root` role.**

2. **Ensure that the data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.**

```
# pkg list -a ha-cluster/data-service/sybase
# pkg publisher
PUBLISHER                             TYPE     STATUS   P  LOCATION
solaris                               origin   online   F  solaris-repository
ha-cluster                            origin   online   F  ha-cluster-repository
```

For information about setting the `solaris` publisher, see "Adding, Modifying, or Removing Package Publishers" in "Adding and Updating Software in Oracle Solaris 11.2".

Tip - Use the `-nv` options whenever you install or update to see what changes will be made, such as which versions of which packages will be installed or updated and whether a new BE will be created.

If you do not get any error messages when you use the `-nv` options, run the command again without the `-n` option to actually perform the installation or update. If you do get error

messages, run the command again with more `-v` options (for example, `-nvv`) or more of the package FMRI pattern to get more information to help you diagnose and fix the problem. For troubleshooting information, see Appendix A, “Troubleshooting Package Installation and Update,” in “Adding and Updating Software in Oracle Solaris 11.2”.

3. **Install the HA for Sybase ASE software package.**

```
# pkg install ha-cluster/data-service/sybase
```

4. **Verify that the package installed successfully.**

```
$ pkg info ha-cluster/data-service/sybase
```

Installation is successful if output shows that `State` is `Installed`.

5. **Perform any necessary updates to the Oracle Solaris Cluster software.**

For instructions on updating your software, see Chapter 11, “Updating Your Software,” in “Oracle Solaris Cluster System Administration Guide”.

Registering and Configuring HA for Sybase ASE

Register and configure HA for Sybase ASE as a failover data service.

Setting HA for Sybase ASE Extension Properties

The section that follows contains instructions for registering and configuring resources. These instructions explain how to set *only* extension properties that HA for Sybase ASE requires you to set. For information about all HA for Sybase ASE extension properties, see Appendix A, “HA for Sybase ASE Extension Properties”. You can update some extension properties dynamically. You can update other properties, however, only when you create or disable a resource. The Tunable entry indicates when you can update a property.

To set an extension property of a resource, include the following option in the `clresource`(1CL) command that creates or modifies the resource:

`-p` *property*=*value*

`-p` *property*	Identifies the extension property that you are setting
value	Specifies the value to which you are setting the extension property

You can also use the procedures in Chapter 2, “Administering Data Service Resources,” in “Oracle Solaris Cluster Data Services Planning and Administration Guide ” to configure resources after the resources are created.

How to Register and Configure HA for Sybase ASE

This procedure describes how to use the Oracle Solaris Cluster maintenance commands to register and configure HA for Sybase ASE.

This procedure includes creating the `SUNW.HAStoragePlus` resource type. This resource type synchronizes actions between `HAStorage` and HA for Sybase ASE and enables you to use a highly available local file system. HA for Sybase ASE is disk intensive, and therefore you should configure the `SUNW.HAStoragePlus` resource type.

For more information about the `SUNW.HAStoragePlus` resource type, see the following documentation:

- `SUNW.HAStoragePlus`(5) man page
- "Relationship Between Resource Groups and Device Groups" in "Oracle Solaris Cluster Data Services Planning and Administration Guide "

Note - Other options also enable you to register and configure the data service. For details about these options, see "Tools for Data Service Resource Administration" in "Oracle Solaris Cluster Data Services Planning and Administration Guide ".

To perform this procedure, you must have the following information.

- The names of the cluster nodes or zones that master the data service.
- The network resource that clients use to access the data service. You typically configure the IP address when you install the cluster. For information about planning the Oracle Solaris Cluster environment and the installation of the Oracle Solaris software, see Chapter 1, "Planning the Oracle Solaris Cluster Configuration," in "Oracle Solaris Cluster Software Installation Guide ".
- The path to the Sybase ASE application installation.

Note - Perform the following steps on one cluster node.

---

1. **On a cluster node, assume the `root` role or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2. **Run the `clresourcetype` command to register resource types for HA for Sybase ASE.**

```
# clresourcetype register SUNW.sybase
```

3. **Register the `SUNW.HAStoragePlus` resource type with the cluster.**

```
# clresourcetype register SUNW.HAStoragePlus
```

4. **Create the resource `sybase-hastp-rs` of type `SUNW.HAStoragePlus`.**

```
# clresource create -g sybase-rg -t SUNW.HAStoragePlus \
-p GlobalDevicePaths=sybase-device-group1,/dev/global/dsk/dl \
-p FilesystemMountPoints=/global/sybase-inst \
-p AffinityOn=TRUE sybase-hastp-rs
```

The resource is created in the enabled state.

Note - `AffinityOn` must be set to `TRUE` and the local file system must reside on global disk groups to be failover.

---

5. **Run the `clresourcegroup` command to complete the following tasks and bring the resource group `sybase-rg` online on a cluster node or zone.**

- Moving the resource group into a managed state
- Bringing the resource group online

This node becomes the primary for device group `sybase-set1` and raw device `/dev/global/dsk/d1`. Device groups that are associated with file systems such as `/global/sybase-inst` are also made primaries on this node.

```
# clresourcegroup online -M sybase-rg
```

6. **Create Sybase ASE application resources in the failover resource group.**

```
# clresource create -g resource-group \
-t SUNW.sybase \
-p Environment_File=environment-file-path \
-p Adaptive_Server_Name=adaptive-server-name \
-p Backup_Server_Name=backup-server-name \
-p Text_Server_Name=text-server-name \
-p Monitor_Server_Name=monitor-server-name \
-p Adaptive_Server_Log_File=log-file-path \
-p Stop_File=stop-file-path \
-p Connect_string=user/passwd \
-p resource_dependencies_offline_restart=storageplus-resource \
-p Wait_for_Online=TRUE \
-p DB_Wait_List=db1, db2, ... \
-p Restart_type=RESOURCE_RESTART|RESOURCE_GROUP_RESTART \
-p Custom_action_file=filepath  resource
```

`-g` *resource-group*

Specifies the resource group name into which the RGM places the resources.

`-t SUNW.sybase`

Specifies the resource type to add.

`-p Environment_File=`*environment-file*

Sets the name of the environment file.

`-p Adaptive_Server_Name=`*adaptive-server-name*

Sets the name of the adaptive server.

`-p Backup_Server_Name=`*backup-server-name*

Sets the name of the backup server.

`-p Text_Server_Name=`*text-server-name*

Sets the name of the text server.

`-p Monitor_Server_Name=`*monitor-server-name*

Sets the name of the monitor server.

`-p Adaptive_Server_Log_File=`*log-file-path*

Sets the path to the log file for the adaptive server.

`-p Stop_File=`*stop-file-path*

Sets the path to the stop file.

`-p Connect_string=`*user/passwd*

Specifies the user name and password that the fault monitor uses to connect to the database.

`-p Wait_for_Online=TRUE`

Specifies whether the `START` method has to wait for the database to be brought online before exiting.

`-p DB_Wait_List=`*db1, db2, ...*

Specifies the list of databases that need to be online before the resource can be brought online. The valid values are either an empty list (“ ”), ALL, or a list of databases.

`-p Restart_type=`*RESOURCE_RESTART*|*RESOURCE_GROUP_RESTART*

Defines the restart behavior of the resource. If the `Restart_type` extension property is set to `RESOURCE_RESTART`, the server fault monitor restarts the database server resource. If the `Restart_type` extension property is set to `RESOURCE_GROUP_RESTART`, the server fault monitor restarts the database server resource group. This property was introduced in `SUNW.sybase` resource type version 5. Prior to and including `SUNW.sybase` resource type version 5, the server fault monitor restarted the database server resource group.

`-p Custom_action_file=`*filepath*

Sets the path to the custom action file that contains the custom fault monitor actions.

resource

Specifies the resource name to add.

The resource is created in the enabled state.

You do not have to specify extension properties that have default values. For more information, see "Setting HA for Sybase ASE Extension Properties" on page 19.

Next Steps After you register and configure HA for Sybase ASE, go to "Verifying the HA for Sybase ASE Installation and Configuration" on page 23.

Verifying the HA for Sybase ASE Installation and Configuration

These checks ensure that all of the nodes that run HA for Sybase ASE can start the Sybase ASE data server. These checks also ensure that other nodes in the configuration can access the Sybase ASE data server. Perform these checks to isolate any problems with starting the Sybase ASE software from HA for Sybase ASE.

▼ How to Verify the HA for Sybase ASE Installation and Configuration

1. **Log in to the node that masters the Sybase ASE resource group.**
2. **Set the Sybase ASE environment variables.**

   The environment variables are the variables that you specify with the `Environment_file` extension property. For information about setting these environment variables, see Appendix A, "HA for Sybase ASE Extension Properties".
3. **Verify that the HA for Sybase ASE resource is online.**

   ```
   # clresource status
   ```
4. **Inspect the Sybase ASE logs to determine the cause of any errors that have occurred.**
5. **Confirm that you can connect to the data server and execute the following test command.**

```
# isql -S adaptive-server -U sa -P password

isql> sp_help
isql> go
isql> quit
```

6. **Kill the process for the Sybase ASE data server.**

   The Oracle Solaris Cluster software restarts the process.

7. **Switch the resource group that contains the Sybase ASE resource to another cluster node or zone.**

   ```
   # clresourcegroup switch -n node[:zone] resource-group
   ```

8. **Log in to the node that now contains the resource group.**

9. **Repeat Step 3 through Step 5.**

Note - Sybase ASE client connections cannot survive a HA for Sybase ASE switchover. If a switchover occurs, the existing client connections to Sybase ASE terminate, and clients must reestablish their connections. After a switchover, the time that is required to replay the Sybase ASE transaction log determines HA for Sybase ASE recovery time.

Location of HA for Sybase ASE Log Files

Each instance of the HA for Sybase ASE data service maintains log files in the `/opt/SUNWscsyb/log` directory.

These files contain information about actions that the HA for Sybase ASE data service performs. Refer to these files to obtain diagnostic information for troubleshooting your configuration or to monitor the behavior of the HA for Sybase ASE data service.

See also "HA for Sybase ASE Logging Issues" on page 25.

HA for Sybase ASE Logging and Security Issues

The following sections contain information about HA for Sybase ASE logging and security issues.

- "HA for Sybase ASE Logging Issues" on page 25

- "HA for Sybase ASE Security Issues" on page 25

HA for Sybase ASE Logging Issues

HA for Sybase ASE logs messages to the file `message_log` in the `/opt/SUNWscsyb/log` directory. Although this file cannot exceed 512 Kbytes, HA for Sybase ASE does not delete old log files. The number of log files, therefore, can grow to a large number.

HA for Sybase ASE writes all of the error messages in the `syslog` file. HA for Sybase ASE also logs fault monitor history to the file `restart_history` in the `log` directory. These files can also grow to a large number.

As part of your regular file maintenance, check the following log files and remove files that you no longer need.

- `syslog`
- `message_log`
- `restart_history`

HA for Sybase ASE Security Issues

HA for Sybase ASE requires that you embed the system administrator's password in a stop file. The `/opt/SUNWscsyb/bin` directory contains the template for the stop file, `sybase_stop_servers`. HA for Sybase ASE uses this file to log in to the Sybase ASE environment and to stop the Sybase ASE servers. Enable the *sybase* user to execute the stop file, but protect the file from general access. Give read, write, and execute privileges to only the following users.

- *sybase* user
- *sybase* group

Tuning the HA for Sybase ASE Fault Monitor

The HA for Sybase ASE fault monitor queries the Sybase ASE server to determine server health.

Note - The HA for Sybase ASE fault monitor monitors only the Adaptive server. The fault monitor does not monitor auxiliary servers.

The HA for Sybase ASE fault monitor is contained in the resource that represents Sybase ASE. You create this resource when you register and configure HA for Sybase ASE. For more information, see “Registering and Configuring HA for Sybase ASE” on page 19.

System properties and extension properties of this resource control the behavior of the fault monitor. The default values of these properties determine the preset behavior of the fault monitor. The preset behavior should be suitable for most Oracle Solaris Cluster installations. Therefore, you should tune the HA for Sybase ASE fault monitor *only* if you need to modify this preset behavior.

Tuning the HA for Sybase ASE fault monitor involves the following tasks:

- Setting the interval between fault monitor probes
- Setting the timeout for fault monitor probes
- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

Perform these tasks when you register and configure HA for Sybase ASE. For more information, see the following sections:

- “Registering and Configuring HA for Sybase ASE” on page 19
- “Tuning Fault Monitors for Oracle Solaris Cluster Data Services” in “Oracle Solaris Cluster Data Services Planning and Administration Guide ”

The HA for Sybase ASE fault monitor consists of the following processes.

- Main fault-monitor process
- Database-client fault probe

Main Fault-Monitor Process

The fault monitor process diagnoses errors and checks statistics. The monitor labels an operation successful if the following conditions occur.

- The database is online.
- The activity check returns no errors.
- The test transaction returns no errors.

If an operation fails, the main process checks the action table for an action to perform and then performs the predetermined action. If an operation fails, the main process can perform the following actions.

1. Restarting the resource on the current node
2. Restarting the resource group on the current node
3. Failing over the resource group to the next node on the resource group's node list

These actions execute external programs as separate processes in the background.

The server fault monitor also scans the `Adaptive_Server_Log` file and corrects any errors that the scan identifies.

Database-Client Fault Probe

The database-client fault probe performs activity checks and test transactions. The extension property `Connect_string` specifies an account that performs all of the database operations. The extension property `Probe_timeout` sets the time-out value that the probe uses to determine the time that has elapsed in a successful database probe.

Obtaining Core Files for Troubleshooting

To facilitate troubleshooting of unexplained DBMS timeouts, you can enable the fault monitor to create a core file when a probe timeout occurs. The contents of the core file relate to the fault monitor process. The fault monitor creates the core file in the root (`/`) directory. To enable the fault monitor to create a core file, use the `coreadm` command to enable set-ID core dumps. For more information, see the `coreadm`(1M) man page.

Customizing the HA for Sybase ASE Fault Monitor

Customizing the HA for Sybase ASE fault monitor enables you to modify the behavior of the server fault monitor as follows:

- Overriding the preset action for an error
- Specifying an action for an error for which no action is preset

Caution - Before you customize the HA for Sybase ASE fault monitor, consider the effects of your customizations, especially if you change an action from restart or switch over to ignore or stop monitoring. If errors remain uncorrected for long periods, the errors might cause problems with the database. If you encounter problems with the database after customizing the HA for Sybase ASE fault monitor, revert to using the preset actions. Reverting to the preset actions enables you to determine if the problem is caused by your customizations.

Customizing the HA for Sybase ASE fault monitor involves the following activities:

- “Defining Custom Behavior for Errors” on page 28
- “Propagating a Custom Action File to All Nodes in a Cluster” on page 36
- “Specifying the Custom Action File That a Server Fault Monitor Should Use” on page 36

Defining Custom Behavior for Errors

The HA for Sybase ASE fault monitor detects the following types of errors:

- DBMS errors that occur during a probe of the database by the server fault monitor
- Alerts that Sybase ASE logs in the Sybase ASE log file
- Timeouts that result from a failure to receive a response within the time that is set by the `Probe_timeout` extension property

To define custom behavior for these types of errors, create a custom action file.

Custom Action File Format

A custom action file is a plain text file. The file contains one or more entries that define the custom behavior of the HA for Sybase ASE fault monitor. Each entry defines the custom behavior for a single DBMS error, a single timeout error, or several logged alerts. A maximum of 1024 entries is allowed in a custom action file.

Note - Each entry in a custom action file overrides the preset action for an error, or specifies an action for an error for which no action is preset. Create entries in a custom action file *only* for the preset actions that you are overriding or for errors for which no action is preset. Do *not* create entries for actions that you are not changing.

An entry in a custom action file consists of a sequence of keyword-value pairs that are separated by semicolons. Each entry is enclosed in braces.

The format of an entry in a custom action file is as follows:

```
{
[ERROR_TYPE=DBMS_ERROR|SCAN_LOG|TIMEOUT_ERROR;]
ERROR=error-spec;
[ACTION=SWITCH|RESTART|STOP|NONE;]
[CONNECTION_STATE=co|di|on|*;]
[NEW_STATE=co|di|on|*;]
[MESSAGE="message-string"]
}
```

White space might be used between separated keyword-value pairs and between entries to format the file.

The meaning and permitted values of the keywords in a custom action file are as follows:

`ERROR_TYPE`

Indicates the type of the error that the server fault monitor has detected. The following values are permitted for this keyword:

`DBMS_ERROR` Specifies that the error is a DBMS error.

`SCAN_LOG` Specifies that the error is an alert that is logged in the alert log file.

`TIMEOUT_ERROR` Specifies that the error is a timeout.

The `ERROR_TYPE` keyword is optional. If you omit this keyword, the error is assumed to be a DBMS error.

`ERROR`

Identifies the error. The data type and the meaning of *error-spec* are determined by the value of the `ERROR_TYPE` keyword as shown in the following table.

`Error_Type` Keyword	Data Type	Meaning
`DBMS_ERROR`	Integer	The error number of a DBMS error that is generated by Sybase ASE
`SCAN_LOG`	Quoted regular expression	A string in an error message that Sybase ASE has logged to the Sybase ASE log file
`TIMEOUT_ERROR`	Integer	The number of consecutive timed-out probes since the server fault monitor was last started or restarted

You must specify the `ERROR` keyword. If you omit this keyword, the entry in the custom action file is ignored.

`ACTION`

Specifies the action that the server fault monitor is to perform in response to the error. The following values are permitted for this keyword:

`NONE` Specifies that the server fault monitor ignores the error.

`STOP` Specifies that the server fault monitor is stopped.

`RESTART` Specifies that the server fault monitor stops and restarts the entity that is specified by the value of the `Restart_type` extension property of the SUNW.sybase resource.

`SWITCH` Specifies that the server fault monitor switches over the database server resource group to another node or zone.

The `ACTION` keyword is optional. If you omit this keyword, the server fault monitor ignores the error.

`CONNECTION_STATE`

Specifies the required state of the connection between the database and the server fault monitor when the error is detected. The entry applies only if the connection is in the required state when the error is detected. The following values are permitted for this keyword:

`*` Specifies that the entry always applies, regardless of the state of the connection.

`co` Specifies that the entry applies only if the server fault monitor is attempting to connect to the database.

`on` Specifies that the entry applies only if the server fault monitor is online. The server fault monitor is online if it is connected to the database.

`di` Specifies that the entry applies only if the server fault monitor is disconnecting from the database.

The `CONNECTION_STATE` keyword is optional. If you omit this keyword, the entry always applies, regardless of the state of the connection.

`NEW_STATE`

Specifies the state of the connection between the database and the server fault monitor that the server fault monitor must attain after the error is detected. The following values are permitted for this keyword:

`*` Specifies that the state of the connection must remain unchanged.

`co` Specifies that the server fault monitor must disconnect from the database and reconnect immediately to the database.

`di` Specifies that the server fault monitor must disconnect from the database. The server fault monitor reconnects when it next probes the database.

The `NEW_STATE` keyword is optional. If you omit this keyword, the state of the database connection remains unchanged after the error is detected.

`MESSAGE`

Specifies an additional message that is printed to the resource's log file when this error is detected. The message must be enclosed in double quotes. This message is additional to the standard message that is defined for the error.

The `MESSAGE` keyword is optional. If you omit this keyword, no additional message is printed to the resource's log file when this error is detected.

Changing the Response to a DBMS Error

The action that the server fault monitor performs in response to each DBMS error is preset as listed in Table A-1. To determine whether you need to change the response to a DBMS error,

consider the effect of DBMS errors on your database to determine if the preset actions are appropriate. For examples, see the subsections that follow.

To change the response to a DBMS error, create an entry in a custom action file in which the keywords are set as follows:

- `ERROR_TYPE` is set to `DBMS_ERROR`.
- `ERROR` is set to the error number of the DBMS error.
- `ACTION` is set to the action that you require.

Responding to an Error Whose Effects Are Major

If an error that the server fault monitor ignores affects more than one session, action by the server fault monitor might be required to prevent a loss of service.

For example, see the following error message:

```
Illegal attempt to change contents of buffer: %S_BUF.
```

No action is preset for Sybase error 835, `Illegal attempt to change contents of buffer: %S_BUF`. However, this Sybase error indicates that the when a client process completes, Adaptive Server performs some cleanup tasks such as closing the buffers and releasing the resources taken up by the buffers. If the client process terminates abnormally, however (for example if the process is killed during execution), Adaptive Server might be unable to carry out the appropriate cleanup, buffers are left open, and Error 835 is raised. If this error affects only a single session, ignoring the error might be appropriate. However, if this error affects more than one session, consider specifying that the server fault monitor restart the database.

The following example shows an entry in a custom action file for changing the response to a DBMS error to restart.

EXAMPLE 1-1 Changing the Response to a DBMS Error to Restart

```
{
ERROR_TYPE=DBMS_ERROR;
ERROR=835;
ACTION=restart;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Illegal attempt to change contents of buffer: %S_BUF.";
}
```

This example shows an entry in a custom action file that overrides the preset action for DBMS error 835. This entry specifies the following behavior:

- In response to DBMS error 835, the server fault monitor performs a restart.

- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- The following message is printed to the resource's log file when this error is detected:

```
"Illegal attempt to change contents of buffer: %S_BUF."
```

Ignoring an Error Whose Effects Are Minor

If the effects of an error to which the server fault monitor responds are minor, ignoring the error might be less disruptive than responding to the error.

For example, see the following error message:

```
Unable to find buffer 0x%lx holding logical page %ld in sdes 0x%lx
kept buffer pool for object '%.*s'.
```

The preset action for Sybase ASE error 804, `Unable to find buffer 0x%lx holding logical page %ld in sdes 0x%lx kept buffer pool for object '%.*s'.` is restart. This error occurs when Adaptive Server cannot find the pointer to a buffer header in a session descriptor. This error can be transient. In this situation, consider specifying that the server fault monitor ignore the error.

The following example shows an entry in a custom action file for ignoring a DBMS error.

EXAMPLE 1-2 Ignoring a DBMS Error

```
{
ERROR_TYPE=DBMS_ERROR;
ERROR=804;
ACTION=none;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Unable to find buffer 0x%lx holding logical page %ld in sdes
0x%lx kept buffer pool for object '%.*s'.";
}
```

This example shows an entry in a custom action file that overrides the preset action for DBMS error 804. This entry specifies the following behavior:

- The server fault monitor ignores DBMS error 804.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.

- No additional message is printed to the resource's log file when this error is detected.

Changing the Response to Logged Alerts

The Sybase ASE software logs alerts in a file that is identified by the `Adaptive_Server_Log_File` extension property. The server fault monitor scans this file and performs actions in response to alerts for which an action is defined.

Logged alerts for which an action is preset are listed in Table A-2. Change the response to logged alerts to change the preset action, or to define new alerts to which the server fault monitor responds.

To change the response to logged alerts, create an entry in a custom action file in which the keywords are set as follows:

- `ERROR_TYPE` is set to `SCAN_LOG`.
- `ERROR` is set to a quoted regular expression that identifies a string in an error message that Sybase ASE has logged to the Sybase ASE log file.
- `ACTION` is set to the action that you require.

The server fault monitor processes the entries in a custom action file in the order in which the entries occur. Only the first entry that matches a logged alert is processed. Later entries that match are ignored. If you are using regular expressions to specify actions for several logged alerts, ensure that more specific entries occur before more general entries. Specific entries that occur after general entries might be ignored.

For example, a custom action file might define different actions for errors that are identified by the regular expressions `Attempt to dirty non-log` and `Attempt to unhash buffer`. To ensure that the entry that contains the regular expression `Attempt to unhash buffer` is not ignored, ensure that this entry occurs before the entry that contains the regular expression `Attempt to`.

The following example shows an entry in a custom action file for changing the response to a logged alert.

EXAMPLE 1-3 Changing the Response to a Logged Alert

```
{
ERROR_TYPE=SCAN_LOG;
ERROR="Attempt to";
ACTION=RESTART;
}
```

This example shows an entry in a custom action file that overrides the preset action for logged alerts about internal errors. This entry specifies the following behavior:

- In response to logged alerts that contain the text `Attempt to`, the server fault monitor performs a restart.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- No additional message is printed to the resource's log file when this error is detected.

Changing the Maximum Number of Consecutive Timed-Out Probes

By default, the server fault monitor restarts the database after the second consecutive timed-out probe. If the database is lightly loaded, two consecutive timed-out probes should be sufficient to indicate that the database is hanging. However, during periods of heavy load, a server fault monitor probe might time out even if the database is functioning correctly. To prevent the server fault monitor from restarting the database unnecessarily, increase the maximum number of consecutive timed-out probes.

Caution - Increasing the maximum number of consecutive timed-out probes increases the time that is required to detect that the database is hanging.

To change the maximum number of consecutive timed-out probes allowed, create one entry in a custom action file for each consecutive timed-out probe that is allowed *except* the first timed-out probe.

Note - You are not required to create an entry for the first timed-out probe. The action that the server fault monitor performs in response to the first timed-out probe is preset.

For the last allowed timed-out probe, create an entry in which the keywords are set as follows:

- `ERROR_TYPE` is set to `TIMEOUT_ERROR`.
- `ERROR` is set to the maximum number of consecutive timed-out probes that are allowed.
- `ACTION` is set to `RESTART`.

For each remaining consecutive timed-out probe except the first timed-out probe, create an entry in which the keywords are set as follows:

- `ERROR_TYPE` is set to `TIMEOUT_ERROR`.
- `ERROR` is set to the sequence number of the timed-out probe. For example, for the second consecutive timed-out probe, set this keyword to 2. For the third consecutive timed-out probe, set this keyword to 3.

- `ACTION` is set to `NONE`.

Tip - To facilitate debugging, specify a message that indicates the sequence number of the timed-out probe.

The following example shows the entries in a custom action file for increasing the maximum number of consecutive timed-out probes to five.

EXAMPLE 1-4 Changing the Maximum Number of Consecutive Timed-Out Probes

```
{
ERROR_TYPE=TIMEOUT;
ERROR=2;
ACTION=NONE;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #2 has occurred.";
}

{
ERROR_TYPE=TIMEOUT;
ERROR=3;
ACTION=NONE;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #3 has occurred.";
}

{
ERROR_TYPE=TIMEOUT;
ERROR=4;
ACTION=NONE;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #4 has occurred.";
}

{
ERROR_TYPE=TIMEOUT;
ERROR=5;
ACTION=RESTART;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #5 has occurred. Restarting.";
}
```

This example shows the entries in a custom action file for increasing the maximum number of consecutive timed-out probes to five. These entries specify the following behavior:

- The server fault monitor ignores the second consecutive timed-out probe through the fourth consecutive timed-out probe.

- In response to the fifth consecutive timed-out probe, the server fault monitor performs a restart.
- The entries apply regardless of the state of the connection between the database and the server fault monitor when the timeout occurs.
- The state of the connection between the database and the server fault monitor must remain unchanged after the timeout occurs.
- When the second consecutive timed-out probe through the fourth consecutive timed-out probe occurs, a message of the following form is printed to the resource's log file:

  `Timeout #`*number* `has occurred.`
- When the fifth consecutive timed-out probe occurs, the following message is printed to the resource's log file:

  `Timeout #5 has occurred. Restarting.`

Propagating a Custom Action File to All Nodes in a Cluster

A server fault monitor must behave consistently on all cluster nodes. Therefore, the custom action file that the server fault monitor uses must be identical on all cluster nodes. After creating or modifying a custom action file, ensure that this file is identical on all cluster nodes by propagating the file to all cluster nodes. To propagate the file to all cluster nodes, use the method that is most appropriate for your cluster configuration:

- Locating the file on a file system that all nodes share
- Locating the file on a highly available local file system
- Copying the file to the local file system of each cluster node by using operating system commands such as the `rcp` command

Specifying the Custom Action File That a Server Fault Monitor Should Use

To apply customized actions to a server fault monitor, you must specify the custom action file that the fault monitor should use. Customized actions are applied to a server fault monitor when the server fault monitor reads a custom action file. A server fault monitor reads a custom action file when the you specify the file.

Specifying a custom action file also validates the file. If the file contains syntax errors, an error message is displayed. Therefore, after modifying a custom action file, specify the file again to validate the file.

Caution - If syntax errors in a modified custom action file are detected, correct the errors before the fault monitor is restarted. If the syntax errors remain uncorrected when the fault monitor is restarted, the fault monitor reads the erroneous file, ignoring entries that occur after the first syntax error.

▼ How to Specify the Custom Action File That a Server Fault Monitor Should Use

1. **On a cluster node or zone, assume the `root` role or assume a role that provides `solaris.cluster.modify` RBAC authorization.**

2. **Set the `Custom_action_file` extension property of the `SUNW.sybase` resource.**

   Set this property to the absolute path of the custom action file.

   ```
   # clresource set -p custom_action_file=filepath  server-resource
   ```

   `-p custom_action_file=`*filepath*

   Specifies the absolute path of the custom action file.

   server-resource

   Specifies the `SUNW.sybase` resource.

♦ ♦ ♦ APPENDIX A

HA for Sybase ASE Extension Properties

For details about system-defined properties, see the `r_properties`(5) man page and the `rg_properties`(5) man page.

The `SUNW.sybase` resource type represents the Sybase ASE application in a Oracle Solaris Cluster configuration. The extension properties of this resource type are as follows:

`Adaptive_Server_Log_ File`

The absolute path to the log file for the adaptive server. HA for Sybase ASE continually reads this property for error monitoring.

Default: None

Range: Minimum=1

Tunable: When disabled

`Adaptive_Server_Name`

The name of the data server. HA for Sybase ASE uses this property to locate the runserver file for Adaptive Server. The file is named `RUN_`*Adaptive-Server-name* and is located in the `$SYBASE/$SYBASE_ASE/install` directory.

Default: None

Range: Minimum=1

Tunable: When disabled

`Backup_Server_Name`

The name of the backup server. HA for Sybase ASE uses this property to locate the runserver file for the backup server. The file is named `RUN_`*backup-server-name* and is located in the `$SYBASE/$SYBASE_ASE/install` directory. If you do not set this property, HA for Sybase ASE will not manage the server.

Default: `Null`

Range: None

Tunable: When disabled

`Connect_cycle`

Number of fault-monitor probe cycles before HA for Sybase ASE establishes a new connection.

Default: 5

Range: 1 – 100

Tunable: Any time

`Connect_string`

String of format *userid/password*, which specifies the database user's user ID and password. HA for Sybase ASE uses this property for database probes. When you set up the HA for Sybase ASE data service, define the database user ID and password before you enable the server resource and the server resource's fault monitor. Do *not* use the `sa` account for the database user. You must set this property, even if you do not set the `Monitor_Server_Name` property.

Default: None

Range: Minimum=1

Tunable: Any time

`DB_Wait_List`

List of databases that need to be online before the resource can be brought online.

Default: Null

Range: None

Tunable: Any time

`Debug_level`

Debug level for writing to the HA for Sybase ASE log.

Default:1

Range: 0 – 100

Tunable: Any time

`Environment_File`

Absolute path to the file that contains all of the Sybase ASE environment variables. HA for Sybase ASE requires that you define the variables `SYBASE`, `SYBASE_ASE`, and `SYBASE_OCS`. Other variables that you define are passed as environment variables to the Sybase server.

The definition of each environment variable must conform to the following format:

variable=value

Each environment variable that you define must also be specified, one per line in the `Environment_File`.

Typically, users use the `SYBASE.sh` environment file that is created by the Sybase installation in the `$SYBASE` directory.

Note - The value of this property is independent of the shell that is being used by the *sybase* user. Specifically, the *sybase* user can have `csh` as its default shell.

Default: None

Range: Minimum=1

Tunable: When disabled

`Monitor_Server_Name`

The name of the monitor server. HA for Sybase ASE uses this property to locate the runserver file in the `$SYBASE/$SYBASE_ASE/install` directory. If you do not set this property, HA for Sybase ASE will not manage the server.

Default: `Null`

Range: None

Tunable: When disabled

`Probe_timeout`

The time after which the fault monitoring probe will time out and register a timeout error while probing the ASE DBMS.

Default: 60 seconds

Range: 1 – 99999 seconds

Tunable: Any time

`Stop_File`

The absolute path to the script that the `STOP` method executes to stop the Sybase ASE servers. This file stores the password of the Sybase ASE system administrator (`sa`). Protect the path so that only the user and group that are associated with the Sybase ASE installation can access the file. The HA for Sybase ASE package includes the `sybase_stop_servers` template. You must replace the existing password.

Default: None

Range: Minimum=1

Tunable: Any time

`Text_Server_Name`

The name of the text server. The HA for Sybase ASE data service uses this property to locate the runserver file in the `$SYBASE/$SYBASE_ASE/install` directory. If you do not set this property, the HA for Sybase ASE data service will not manage the server.

Default: `Null`

Range: None

Tunable: When disabled

`Wait_for_online`

Whether the `START` method waits for the database to be brought online before exiting.

Default: `TRUE`

Range: `TRUE` – `FALSE`

Tunable: Any time

`Restart_type`

When a fault with the resource is detected, the `RESOURCE_RESTART` value will ensure that only this resource is restarted. A value of `RESOURCE_GROUP_RESTART` will ensure that all resources in the resource group are restarted.

Default: `RESOURCE_RESTART`

Range: `RESOURCE_RESTART` – `RESOURCE_GROUP_RESTART`

Tunable: Any time

`Custom_action_file`

This file contains the custom fault monitor actions.

Default: None

Range: None

Tunable: Any time

Preset actions for DBMS errors and Sybase ASE logged alerts are listed as follows:

- DBMS errors for which an action is preset are listed in Table A-1.
- Logged alerts for which an action is preset are listed in Table A-2.

TABLE A-1 Preset Actions for DBMS Errors

Error Number	Action	Connection State	New State	Message
601	RESTART	*	*	Descriptor for system table in database not found in the descriptor hash table.
605	RESTART	*	co	Attempt to fetch logical page in database belongs to different object. See errorlog file for further details.
623	RESTART	on	*	Attempt to retrieve row from page via RID failed because logical page is not a data page.
706	RESTART	on	*	Process tried to remove PROC_HDR that it does not hold in Pss.
707	RESTART	on	*	System error detected during attempt to free memory at address ox%1x. See errorlog file for further details.

Error Number	Action	Connection State	New State	Message
803	RESTART	on	*	Unable to place buffer holding logical page in sdes. See errorlog file for further details.
804	RESTART	on	co	Unable to find buffer holding logical page in sdes.
806	RESTART	on	co	Could not find virtual page for logical page %ld in database '%S_DBID'.
813	NONE	on	co	Logical page in database is already hashed. See errorlog file for further details.
821	RESTART	on	co	Attempt to unhash buffer. See errorlog file for further details.
822	SWITCH	*	*	Could not start I/O for request BLKIO. See errorlog file for further details.
823	STOP	*	*	I/O error detected during %S_MSG for %S_BUF.
921	NONE	*	*	Database has not recovered yet.
945	RESTART	on	co	Function %s was asked to locate page in database.
1207	RESTART	on	co	Locksleep called with bad process id. See errorlog file for further details.
1265	RESTART	on	*	A lock manager routine expected a lock record as an argument but was passed something else.
1509	STOP	on	*	Row compare failure.
1510	STOP	on	*	Sort failed: Out of space in database.
1602	STOP	on	*	Unable to initialize network.
1605	STOP	on	*	Failed to open virtual socket for new connections.
1813	RESTART	on	*	Cannot open new database. CREATE DATABASE is aborted.
3002	RESTART	on	co	DUMP DATABASE for database '%.*' failed to remove alarm with ID = %d. See errorlog file for further details.
3307	RESTART	on	co	Process %d was expected to hold logical lock on page %ld. See errorlog file for further details.
4002	NONE	*	co	Login failed. Please check fault monitor user id and password specified in the CONNECT_STRING property
4067	NONE	*	co	Login failed. Please check fault monitor user id and password specified in the CONNECT_STRING property
6902	SWITCH	on	co	Page timestamp value falls between the old and new timestamps from log.
7409	SWITCH	on	*	This task tried to sleep waitng for space to become available in the log, but could not.
7410	SWITCH	on	*	Function called with invalid argument. See errorlog file for further details.
7412	RESTART	on	*	Space available in the log segment has fallen critically low in database.
7413	RESTART	on	*	Task(s) are sleeping, waiting for space to become available in the log segment for database.
8201	RESTART	on	*	Keep count of descriptor was expected to be %d. Instead %d was found.
8203	RESTART	on	*	Expected to find the descriptor in %s state.

Error Number	Action	Connection State	New State	Message
8204	STOP	on	*	Expected to receive system catalog id.
8211	RESTART	on	*	Mismatch found between the name and id descriptor hash table.
8704	RESTART	on	*	Could not get the address lock.
17716	RESTART	on	*	Could not update the last-chance threshold for database"; }
16843058	NONE	*	co	Connection broken.
16843175	NONE	*	co	Error in performing requested operation
33620275	NONE	*	co	Error in performing requested operation
84083972	RESTART	*	co	Server is not up, restarting
84083974	NONE	*	co	Server is recovering

TABLE A-2 Preset Actions for Sybase ASE Alerts

Alert String	Action	Connection State	New State	Message
server: Buffer	STOP	on	co	server error
kistartup: could	RESTART	*	co	kistartup: could not create shared memory
Kernel: current	NONE	*	co	Kernel: current process infected with %d
kernel: dstart I/O	STOP	*	*	kernel: dstart I/O request repeatedly delayed; block number: %ld, vdn: %ld. See errorlog file for further details
basis_dlock:	STOP	*	*	basis_dlock: file '%s' already in use by a SQL Server
Physical memory	STOP	*	*	Physical memory on this machine might be too fragmented
server: Warning:	NONE	*	co	OPEN OBJECTS parameter might be too low; attempt was made to free up descriptor in desfree(). Run sp_configure to increase parameter value.
kernel: write	STOP	*	*	kernel: write error on virtual disk %d, block %ld. See errorlog file for further details
kernel: read	STOP	*	*	kernel: read error on virtual disk %d, block %ld. See errorlog file for further details
Stack Guardword	SWITCH	*	*	kernel: *** Stack Guardword corrupted

Index

A

action file *See* custom action file
`ACTION` keyword, 29
actions
  server fault monitor
    modifying, 29
`Adaptive_Server_Log_File` extension property, 39
`Adaptive_Server_Name` extension property, 39
adding
  messages to log files, 30
alert logs
  modifying response to errors, 33

B

`Backup_Server_Name` extension property, 39

C

caution notice
  server fault monitor customizations, 27
configuring
  HA for Sybase ASE, 20
  Sybase ASE database with Solaris Volume Manager, 15
`Connect_cycle` extension property, 40
`Connect_string` extension property, 40
`CONNECTION_STATE` keyword, 29
creating
  Sybase ASE database environment, 16
custom action file
  format, 28
  maximum number of entries in, 28
  order of entries in, 33
  propagating to cluster nodes, 36
  specifying, 36
  validating, 36
`Custom_action_file` extension property, 42
customizing
  server fault monitor, 27

D

database management system (DBMS)
  errors
    modifying response to, 30
    preset actions, 42
`DB_Wait_List` extension property, 40
DBMS (database management system)
  errors
    modifying response to, 30
    preset actions, 42
`Debug_level` extension property, 40

E

`Environment_File` extension property, 40
`ERROR` keyword, 29
`ERROR_TYPE` keyword, 28
errors
  in custom action file, 36
  DBMS
    modifying response to, 30
    preset actions for, 42
  ignoring, 32
  responding to, 31
  SGA, 31
  timeouts
    modifying maximum allowed, 34
  types detected by fault monitor, 28
extension properties

HA for Sybase ASE, 39

F

fault monitors
actions
modifying, 29
caution notice, 27
customizing, 27
error types detected by, 28
tuning, 25
files
alert log
modifying response to errors in, 33
custom action
format, 28
order of entries in, 33
propagating to cluster nodes, 36
specifying, 36
validating, 36
HA for Sybase ASE logs, 24
additional messages in, 30
fragmentation
of memory, 31

H

HA for Sybase ASE, 10
See also Sybase ASE
extension properties, 39
fault monitors, 25
installing, 18
preparing nodes for, 10
log files, 24
additional messages in, 30
logging and security issues, 24
registering and configuring, 20
software package, installing, 18
`SUNW.HAStoragePlus` resource type, 20
verifying installation, 23
heap memory, 32

I

ignoring
minor errors, 32
installing
HA for Sybase ASE, 18
Sybase ASE, 12
insufficient memory errors, 31, 32

K

keywords
custom action file, 28

L

log files
HA for Sybase ASE, 24
additional messages in, 30
logged alerts
modifying response to, 33

M

maximum values
entries in custom action file, 28
number of allowed timeouts, 34
memory
shortage of, 31, 32
`MESSAGE` keyword, 30
modifying
number of allowed timeouts, 34
response to DBMS errors, 30
response to logged alerts, 33
server fault monitor actions, 29
`Monitor_Server_Name` extension property, 41

N

`NEW_STATE` keyword, 30

O

Oracle Solaris Cluster software
publisher, 18, 18
order

entries in custom action file, 33
overriding
server fault monitor presets, 27

P

package, 18
preventing
unnecessary restarts
for DBMS errors, 32
for timeouts, 34
`Probe_timeout` extension property, 41
properties
HA for Sybase ASE, 39
publisher
Oracle Solaris Cluster software, 18, 18

R

registering
HA for Sybase ASE, 20
resource types
fault monitors, 25
responding
to major errors, 31
`Restart_type` extension property, 42
restarts
preventing
for DBMS errors, 32
for timeouts, 34

S

security issues
HA for Sybase ASE, 24
server fault monitor
actions
modifying, 29
caution notice, 27
customizing, 27
error types detected by, 28
sessions
effect of errors on, 31, 32
SGA (shared global area)
errors, 31
shared global area (SGA)
errors, 31
software package, 18
`Stop_File` extension property, 41
`SUNW.HAStoragePlus` resource type
HA for Sybase ASE, 20
Sybase
error numbers, 42
Sybase ASE, 10
See also HA for Sybase ASE
database
configuring with Solaris Volume Manager, 15
creating environment, 16
fault monitors, 25
installing
performing, 12
preparing nodes for, 11
verifying installation, 14
syntax errors
custom action file, 36
system properties
effect on fault monitors, 26

T

`Text_Server_Name` extension property, 41
timeouts
modifying maximum allowed, 34
tuning
fault monitors, 25

V

validating
custom action file, 36
verifying
HA for Sybase ASE, 23
Sybase ASE installation, 14

W

`Wait_for_online` extension property, 42