

Veritas™ High Availability Agent for Sybase Installation and Configuration Guide

Solaris

5.0

Veritas High Availability Agent for Sybase Installation and Configuration Guide

Copyright © 2006 Symantec Corporation. All rights reserved.

VCS 5.0

Symantec, the Symantec logo, Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID, SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Printed in the United States of America.

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

Solaris is a trademark of Sun Microsystems, Inc.

Licensing and registration

Veritas Cluster Server is a licensed product. See the *Veritas Cluster Server Installation Guide* for license installation instructions.

Technical support

For technical assistance, visit <http://support.veritas.com> and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1	Introduction	
	Supported software	7
	About the Veritas high availability agent for Sybase	8
	Agent for SQL Server: Sybase	8
	Agent for Backup Server: SybaseBk	9
	Monitoring Sybase instances running in Solaris 10 zones	9
	Monitoring options	10
	How the agent makes Sybase highly available	11
	Typical Sybase configuration in a VCS cluster	11
	Setting up a Sybase cluster	12
Chapter 2	Installing and configuring Sybase	
	VCS requirements for installing Sybase	13
	Sybase installation directory	13
	\$SYBASE directory on shared disks	14
	Database dbspaces	14
	Transparent TCP/IP failover	14
	System user for Sybase home directory	14
	Long pathname limitation for \$SYBASE	14
	Language settings	16
	Configuring Sybase for detail monitoring	17
Chapter 3	Installing the agent	
	Prerequisites	19
	Installing the agent	19
	Upgrading the agent	21
Chapter 4	Configuring the Sybase service group	
	About configuring a service group for Sybase	23
	Prerequisites	24
	Importing the SybaseTypes.cf file	24
	Configuring the service group from Cluster Manager (Java Console)	25
	Editing resource attributes	26
	Configuring the service group from the command line	27

	Encrypting passwords	28
	Setting up detail monitoring for the agent	30
	Enabling detail monitoring	30
	Disabling detail monitoring	31
Chapter 5	Administering the Sybase service group	
	Bringing the service group online	33
	Taking the service group offline	34
	Switching the service group	34
	Viewing the agent log	35
	Modifying the service group configuration	35
	Disabling the agent	35
	Removing the agent	36
Appendix A	Resource type definition	
	About the resource type and attribute definitions	37
	Sybase resource type	38
	Attribute definitions	39
	SybaseBk resource type	41
	Attribute definitions	41
Appendix B	Sample configuration	
	About the sample configuration for Sybase agent	43
	Resource dependency graph	44
	Resource dependency for Sybase configured in Solaris zones	44
	Sample configuration	47
	Sample configuration in Solaris zones	48
Index		49

Introduction

This chapter contains the following topics:

- Supported software
- About the Veritas high availability agent for Sybase
- Monitoring Sybase instances running in Solaris 10 zones
- Monitoring options
- How the agent makes Sybase highly available
- Typical Sybase configuration in a VCS cluster
- Setting up a Sybase cluster

Supported software

- Sybase Adaptive Server Enterprise (ASE) 12.5.x and 15.
- VCS 5.0 on Solaris
- Solaris 2.8, 2.9, and 2.10

Note: For each platform, Veritas recommends applying the latest Solaris operating system patches available from Sun. See <http://sunsolve.Sun.com>. Within the cluster, all systems must use the same operating system version and patch level.

About the Veritas high availability agent for Sybase

The Veritas high availability agent for Sybase brings the configured Sybase servers online, monitors them, and takes them offline.

The package contains two agents:

- Agent for SQL Server: Sybase
- Agent for Backup Server: SybaseBk

The agents include type declarations and agent executables, and are represented with Sybase and SybaseBk resource types, respectively. Both agents work together to make Sybase highly available in a VCS cluster.

Note: Veritas high availability agent for Sybase provides “active/passive” support for Sybase. For “active/active” support, contact Sybase for their agent.

Agent for SQL Server: Sybase

The agent for Sybase starts a Sybase SQL Server, monitors the server processes, and shuts down the server. Specific agent operations include:

- **Online**— Starts the Sybase SQL Server by using the following command:

```
startserver -f $SYBASE/$SYBASE_ASE/install/  
RUN_$Server
```
- **Monitor**— In basic monitoring mode, the agent scans process table for the `dataserver` process. In detail monitoring mode, the agent runs the script specified in `Monscript` as an option. For more information on the monitoring options, see “[Monitoring options](#)” on page 10.
- **Offline**— Stops the Sybase SQL Server by using the `isql` command in the following manner:

The agent first executes the command `shutdown with wait`. If this fails, the offline script executes `shutdown with nowait`.
- **Clean**— Forcefully stops the Sybase SQL Server by using the `isql` command in the following manner:

The agent first executes the command `shutdown with wait`. If this fails, the clean script executes `shutdown with nowait`.

If the process does not respond to the `shutdown` command, the agent scans the process table for processes associated with the configured database and kills them.

Agent for Backup Server: SybaseBk

The agent for SybaseBk starts a Sybase Backup Server, monitors the server process, and shuts down the server. Specific agent operations include:

- **Online**— Starts the Sybase Backup Server by using the following command:

```
startserver -f $SYBASE/$SYBASE_ASE/install/  
RUN_$BackupServer
```

- **Monitor**— Scans process table for the `backupserver` process.

- **Offline**— Stops the Sybase Backup Server by using the following `isql` command:

The agent first executes the command `shutdown SYB_BACKUP with wait`. If this fails, the offline script executes `shutdown SYB_BACKUP with nowait`.

- **Clean**— Forcefully stops the Sybase Backup Server by using the `isql` command in the following manner:

The agent first executes the command `shutdown SYB_BACKUP with wait`. If this fails, the clean script executes `shutdown SYB_BACKUP with nowait`.

If the process does not respond to the `shutdown` command, the agent scans the process table for processes associated with the configured Sybase Backup Server and kills them.

Monitoring Sybase instances running in Solaris 10 zones

Solaris 10 provides a means of virtualizing operating system services, allowing one or more processes to run in isolation from other activity on the system. Such a "sandbox" is called a "non-global zone." Each zone can provide a rich and customized set of services. There is also a "global zone" and processes running in this zone have the same set of privileges available on a Solaris system today. VCS provides high availability to applications running in non-global zones by extending the failover capability to zones. VCS is installed in a global zone and all VCS agents and engine components run in the global zone. For applications running within non-global zones, agents run entry points inside the zones. If a zone configured under VCS control faults, VCS fails over the entire service group containing the zone.

The Veritas high availability agent for Sybase is zone-aware and can monitor Sybase instances running in non-global zones.

Monitoring options

The Veritas high availability agent for Sybase provides two levels of application monitoring: basic and detail.

In the basic monitoring mode, the agent for Sybase monitors the Sybase daemon processes to verify whether they are running. In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Sybase is functioning properly. See “[Setting up detail monitoring for the agent](#)” on page 30 for more information.

How the agent makes Sybase highly available

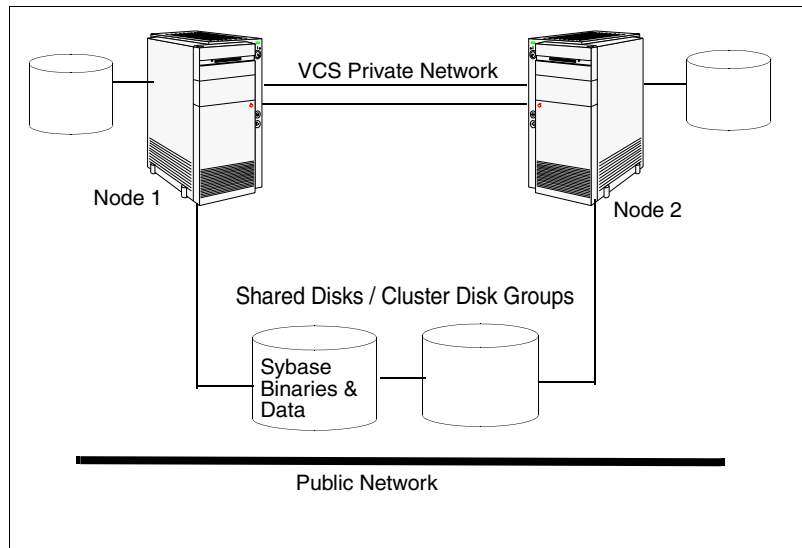
In the basic monitoring mode, the agent detects an application failure if a configured Sybase server process is not running. In the detail monitoring mode, the agent detects application failure if it cannot perform a transaction in the test table in the Sybase database server.

When the agent detects that the configured Sybase server is not running on a system, the Sybase service group is failed over to the next available system in the service group's SystemList. The configured Sybase servers are started on the new system, thus ensuring high availability for the Sybase server and data.

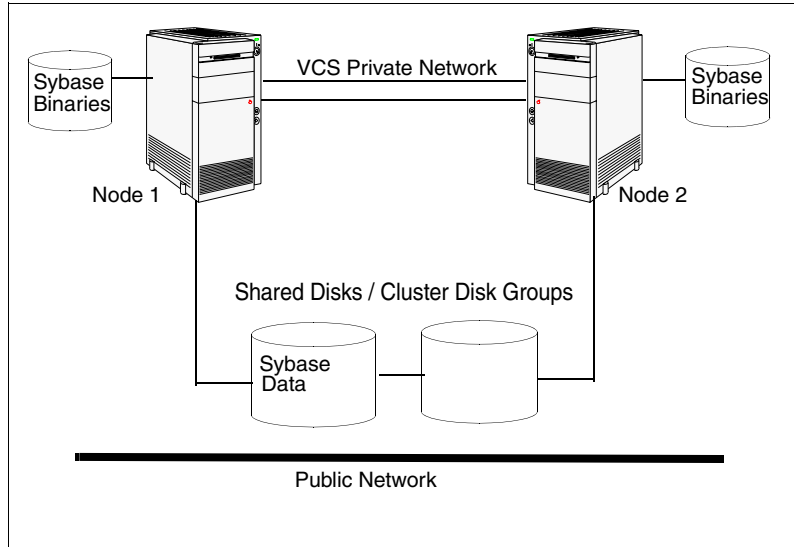
Typical Sybase configuration in a VCS cluster

In a typical configuration, VCS is configured in a two node cluster. The Sybase data is installed on shared disks. The Sybase server binaries can be installed locally on both nodes or on shared disks. The agent for Sybase is installed on both nodes. The shared disks can be managed using Veritas Volume Manager (VxVM).

In the configuration depicted in the illustration below, the Sybase servers, including binaries and data, are installed completely on shared disks or shared cluster disk groups managed using VxVM.

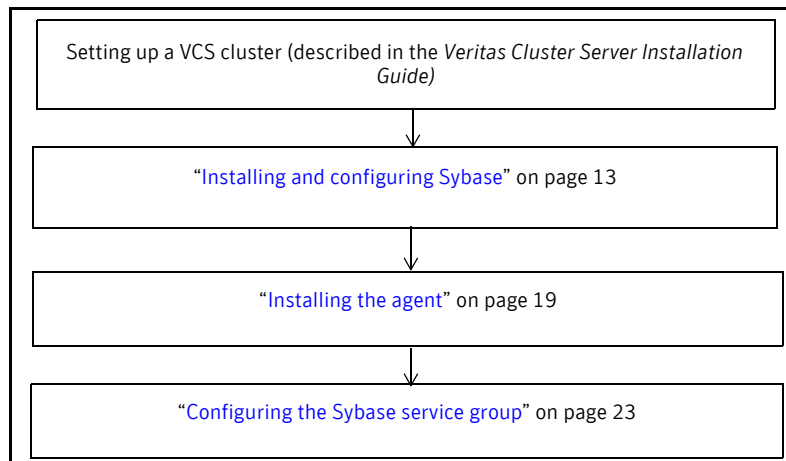


In the configuration depicted in the illustration below, the Sybase binaries are installed locally on each node in the cluster and the Sybase data is on shared disks or shared cluster disk groups managed using VxVM.



Setting up a Sybase cluster

Setting up a Sybase cluster in a VCS environment involves the following tasks. Each task is described in detail in subsequent chapters.



Installing and configuring Sybase

This chapter contains the following topics:

- [VCS requirements for installing Sybase](#)

VCS requirements for installing Sybase

Make sure you meet the requirements mentioned here before installing Sybase in a VCS cluster. For information on how to install Sybase, refer to Sybase documentation. Before installing Sybase, make sure the systems in the cluster have adequate resources to run Sybase and VCS.

Sybase installation directory

The Sybase installation directory can be on a local disk or a shared storage.

- If the Sybase binaries are installed on a local disk, verify the installation path is same on all the nodes in the cluster. Make sure the Sybase configuration files are identical on all the nodes in the cluster.
- If the Sybase binaries are installed on shared disks, make sure the mount points for the shared disks are same on all the nodes. The Sybase installation directory is specified by the environment variable `$SYBASE`. Create the same `$SYBASE` mount points on each system.

\$SYBASE directory on shared disks

All database devices, including master devices, `sybsystemprocs`, and information about Sybase user must be located on shared disks. If the database devices are created on file systems, the file systems must also be located on shared disks. Create the same file system mount points on each system to access the shared disks.

Database dbspaces

If you are using shared disks for dbspaces, change the permissions, or access mode on the disk groups that store the Sybase data. Change the permissions for `sybase` to `660`.

For example, if you are using Veritas Volume Manager, type:

```
# vxedit -g diskgroup_name set group=dba
      user=sybase mode=660 volume_name
```

Transparent TCP/IP failover

For Sybase server failover to be transparent to Sybase clients, create an IP address as part of the Sybase service group. This IP address must match the `dataserver` and `backup server` entries in the `$SYBASE/interfaces` file. For information on the format for adding entries to the `$SYBASE/interfaces` file, refer to the Sybase documentation.

System user for Sybase home directory

Make sure you have a system user, with the same username and ID, on all cluster nodes. Also, the system user should have the ownership of the Sybase home directory on the shared disk. Type the following commands:

```
# useradd -u user_id user_name
# chown -R user_name $SYBASE
```

Long pathname limitation for \$SYBASE

The Solaris process table limits process names to 79 characters. A process having a longer pathname is truncated in the table, making it unrecognizable. The Sybase home directory (`$SYBASE`) could possibly have such a long pathname. In such a case, you can create a soft link to the `$SYBASE` directory and use it in place of the long filename in the appropriate Sybase installation files. See “[Using a soft link to a \\$SYBASE pathname](#)” on page 15 for a list of files to be edited and examples.

Using a soft link to a \$SYBASE pathname

Using a soft link pathname avoids the problems posed by the long pathname. After creating the soft link on each system, you must edit the following files, making the appropriate substitutions:

- The file `RUN_Server` in the directory `$SYBASE/$SYBASE_ASE/install`.
- The file `RUN_Server_back` in the directory `$SYBASE/$SYBASE_ASE/install`.

Example: Replacing \$SYBASE pathname with a soft link

The following example demonstrates how to replace a \$SYBASE pathname with a soft link.

To replace a \$SYBASE pathname with a soft link

- 1 On each system in the cluster, create a soft link to the long pathname. For example:

```
# ln -s /opt/apps/sybase/home/directory/is/longer/than
    eighty/characters/sybase /opt/link_to_longpath
```

Now the process will be invoked with the short pathname of the soft link.

- 2 In the `$SYBASE/$SYBASE_ASE/install` directory, edit the two files `RUN_Server` and `RUN_Backupserver_back`. Find all instances of the long pathname (e.g. `/opt/apps/sybase/home/directory/is/longer/than/eighty/characters/sybase`) and replace them with the soft link (e.g. `/opt/link_to_longpath`).

For example, the file `RUN_Server` resembles the following before the change:

```
/opt/apps/sybase/home/directory/is/longer/than/eighty
    /characters/sybase/ASE-12_5/bin/dataserver\
-d/dev/vx/rdisk/db_dg1/vol2\
-e/opt/apps/sybase/home/directory/is/longer
    /than/eighty/characters/sybase/install/fw17i.log\
-M/opt/apps/sybase/home/directory/is/longer/than/eighty
    /characters/sybase\
-sfw17i\
```

After the replacement, the file resembles:

```
/opt/link_to_longpath/ASE-12_5/bin/dataserver\
-sfw17i\
-d/dev/vx/rdisk/db_dg1/vol2\
-e/opt/link_to_longpath/install/fw17i.log\
-M/opt/link_to_longpath\
```

Make sure the `-s` option and its argument (`fw17i` in the above example) are the first to be listed; they must be placed within the first eighty characters of the file. Failure to do this will *not* bring the service group online.

For example, if you do not put the `-s` option and the argument in the first eighty characters, the command string that will be considered is as follows:

```
/opt/apps/sybase/home/directory/is/longer/than/eighty  
/characters/sybase/ASE-12_5/
```

In this case, the `-s` option will be omitted and the service group will fail to come online. However, if you moved the `-s` to be the first option, the first eighty characters of the command string for the same example will be as follows:

```
/opt/link_to_longpath/ASE-12_5/bin/dataserver\  
-sfw17i\  
-d/dev/vx/rdisk/db_dg1/vol2
```

In this case, the `-s` option is among the first eighty characters.

- 3 In the file `/etc/VRTSvcs/conf/config/main.cf`, edit the Home attribute for both the Sybase type and the SybaseBk type (see “[Sybase resource type](#)” on page 38 and “[SybaseBk resource type](#)” on page 41) For example:

```
Home = "/opt/link_to_longpath"
```

Language settings

For the Veritas high availability agent for Sybase to function with the desired locale, make sure that the Sybase installation has the correct localization files. For example, if the Sybase server requires ‘LANG=en_US.UTF-8’ environment variable, verify that the localization files corresponding to language ‘en_US.UTF-8’ are installed with Sybase. Also, edit the file `$VCS_HOME/bin/vcsenv` to contain the following:

```
LANG=en_US.UTF-8;export LANG
```

Note that this change would affect all VCS agents configured on the nodes.

Configuring Sybase for detail monitoring

This section describes the tasks to be performed for configuring a Sybase server to be monitored in detail. For more information about detail monitoring, see [“Setting up detail monitoring for the agent”](#) on page 30.

Note: The steps described here are specific to the sample script, `SqlTest.pl`, provided with the agent. If you will use a custom script for detail monitoring, you must configure the Sybase database accordingly.

To set up Sybase for detail monitoring

Perform these steps only once in a Sybase cluster.

- 1 Source the `SYBASE.sh` file or `SYBASE.csh` file (depending on the user shell) to set the `$SYBASE` and `$SYBASE_ASE` environment variables.

- 2 Start the Sybase server.

```
# $SYBASE/$SYBASE_ASE/install/RUN_<server_name>
```

- 3 Start the Sybase client on any cluster node.

```
# isql -Usa
```

Enter the administrator password when prompted to do so.

- 4 Connect to the master database.

```
1> use master
2> go
```

- 5 Create a Sybase user account.

```
1> sp_addlogin <user_name>, <password>
2> go
```

The detail monitor script should use this account to make transactions on the database.

- 6 Create a database.

```
1> create database <database_name>
2> go
```

The detail monitor script should make transactions on this database.

- 7 If required, restrict the size of the log file for the database.

```
1> sp_dboption <database_name>, "trunc log on chkpt", true
2> go
```

- 8 Connect to the database created in [step 6](#).

```
1> use <database_name>
2> go
```

- 9 Associate the user created in [step 5](#) with the database created in [step 6](#).

```
1> sp_adduser <user_name>
2> go
```

- 10 Change the user to the one created in [step 5](#).

```
1> setuser "user_name"  
2> go
```

- 11 Create a table in the database.

```
1> create table <table_name> (lastupd datetime)  
2> go
```

The detail monitor script should make transactions on this table.

Note: If you will use the `SqlTest.pl` for detail monitoring, make sure you create a table with a `lastupd` field of type `datetime`.

- 12 Verify the configuration by adding an initial value to the table.

```
1> insert into <table_name> (lastupd) values (getdate())  
2> go
```

- 13 Exit the database.

```
1> exit
```

Installing the agent

This chapter contains the following topics:

- [Prerequisites](#)
- [Installing the agent](#)
- [Upgrading the agent](#)

Prerequisites

- ✓ Verify VCS is installed and configured. Veritas recommends installing the VCS graphical user interface. If required, review the *Veritas Cluster Server Installation Guide*.
- ✓ Verify Sybase is installed and configured, with the considerations described under “[VCS requirements for installing Sybase](#)” on page 13, on all cluster nodes where you will be installing the agent. Review the Sybase documentation for more information.
- ✓ Verify that the `sybase` account is valid and identical on all cluster systems that will run Sybase. Also, verify the `sybase` user account has execute permissions to Sybase binaries. See “[System user for Sybase home directory](#)” on page 14 for more information.

Installing the agent

Install the Veritas high availability agent for Sybase on each node in the cluster.

To install the agent

- 1 Log in as root.
- 2 Create a temporary directory for installation:

```
# mkdir /tmp/install
```

- 3 Insert the software disk in a disk drive connected to the system.
 - If you are running Solaris volume-management software, the software automatically mounts the disc as `/cdrom/cdrom0`. Type the following to go to the location:

```
# cd /cdrom/cdrom0
```

- If you are not running Solaris volume-management software, you must mount the disc manually. For example:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
```

Where, in this example, `/dev/dsk/c0t6d0s2` is the default for the disc drive.

Once the disc is mounted, type the following commands to go to the location:

```
# cd /cdrom
```

- 4 Copy the compressed package files from the software disc to the temporary directory:

```
# cp -r solaris/application/sybase_agent/5.0/* /tmp/install
```
- 5 Go to the temporary directory and unzip the compressed package file:

Note: If your system does not have the gunzip utility, copy it from the disc:

```
# cp /cdrom_path/gnu/gunzip /tmp/install
```

```
# cd /tmp/install  
# gunzip VRTSvcsy.tar.gz
```

- 6 Extract the compressed file from the tar file:

```
# tar -xvf VRTSvcsy.tar
```
- 7 Install the package:

```
# pkgadd -d . VRTSvcsy
```
- 8 Copy the file `SybaseTypes.cf` from `/etc/VRTSagents/ha/conf/Sybase/` directory to `/etc/VRTSvcs/conf/config` directory.

To install the Japanese language pack

- 1 After installing the agent, insert the “Language” disk into a drive connected to the system. See [step 3](#) on page 20 for instructions on mounting the disk.
- 2 Copy the compressed package files from the software disc to the temporary directory:

```
# cp -r ja/sybase_agent/pkgs/* /tmp/install
```
- 3 Go to the temporary directory and unzip the compressed package file:

```
# cd /tmp/install  
# gunzip VRTSjacss.tar.gz
```
- 4 Extract the compressed file from the tar file:

```
# tar -xvf VRTSjacss.tar
```

5 Install the Japanese package:

```
# pkgadd -d . VRTSjacss
```

Upgrading the agent

This section describes steps to upgrade from the VCS enterprise agent 1.3.1 or 4.0 for Sybase in a VCS cluster.

To upgrade the agent

- 1 Save the VCS configuration and stop the VCS engine.

```
# haconf -dump -makero
```

```
# hastop -all -force
```
- 2 Back up the configuration file, main.cf to a location on the cluster node.
- 3 Repeat [step a](#) through [step c](#) on all systems that have the agent for Sybase installed.
 - a Remove the agent for Sybase

```
# pkgrm VRTSvcssy
```
 - b Delete the file `/etc/VRTSvcs/conf/config/SybaseTypes.cf`.
 - c Install the Veritas high availability agent for Sybase. See [“Installing the agent”](#) on page 19.
- 4 Copy the file `SybaseTypes.cf` from the `/etc/VRTSagents/ha/conf/Sybase/` directory to `/etc/VRTSvcs/conf/config` directory.
- 5 Copy the `main.cf` from the backed up location to `/etc/VRTSvcs/conf/config` directory.
- 6 Edit the `main.cf` located under `/etc/VRTSvcs/conf/config`.

Note: Perform this steps only while upgrading the Veritas high availability agent for Sybase from 1.3.1 or 4.0 to 5.0.

- a If you configured detail monitoring, make sure the `Monscript` attribute of the Sybase resource type is properly configured.
- b Remove the `Monscript` attribute for the SybaseBk resource type, if configured.
- c Replace the password specified for the `SAPswd` attribute (of Sybase and SybaseBk resources) and `UPword` attribute (of Sybase resource) with encrypted passwords. See [“Encrypting passwords”](#) on page 28 for instructions on encrypting passwords.

- d** Save the main.cf
- 7** Verify the configuration.

```
# cd /etc/VRTSvcs/conf/config  
# hacf -verify.
```
- 8** Start VCS on the local node.
- 9** Start VCS on other nodes.

Configuring the Sybase service group

This chapter contains the following topics:

- [About configuring a service group for Sybase](#)
- [Prerequisites](#)
- [Configuring the service group from Cluster Manager \(Java Console\)](#)
- [Configuring the service group from the command line](#)
- [Encrypting passwords](#)
- [Setting up detail monitoring for the agent](#)

About configuring a service group for Sybase

Configuring the Sybase service group involves configuring service group resources and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the Sybase service using the following methods:

- By using VCS Cluster Manager (Java Console) to edit a resource group template for the agent. See “[Configuring the service group from Cluster Manager \(Java Console\)](#)” on page 25).
- By using the types configuration files and directly editing the sample `main.cf` file supplied with the agent. This method requires you to restart VCS before the configuration takes effect. See “[Configuring the service group from the command line](#)” on page 27.

Before configuring the agent, see [Appendix A, “Resource type definition”](#) on page 37 to review the Sybase resource types and their attributes.

Prerequisites

- ✓ Verify Sybase is installed and configured, with the considerations described under “[VCS requirements for installing Sybase](#)” on page 13, on all cluster nodes where you will be configuring the service group.
- ✓ Verify the Veritas high availability agent for Sybase is installed on all cluster systems. For more information, see “[Installing the agent](#)” on page 19.
- ✓ Verify the type definition for Veritas high availability agent for Sybase is imported into the VCS engine. See “[Importing the SybaseTypes.cf file](#)” on page 24 for instructions.

Importing the SybaseTypes.cf file

Before configuring the Sybase service group, you must import the `SybaseTypes.cf` file to the VCS engine.

To import using the Cluster Manager

- 1 Start Cluster Manager and log on to the cluster.
- 2 From the **File** menu select **Import Types...**
- 3 In the Import Types dialog box, select the file:
`/etc/VRTSagents/ha/conf/Sybase/SybaseTypes.cf`
- 4 Click **Import**.
- 5 Save the configuration.

To import using the command line

- 1 Log in to a cluster system as `root`.
- 2 Make the cluster configuration as read-only. This ensures all changes to the existing configuration have been saved and further changes are prevented while you modify `main.cf`:

```
# haconf -dump -makero
```
- 3 To ensure that VCS is not running while you edit `main.cf`, issue the following command to stop the VCS engine on all systems and leave the resources available online:

```
# hastop -all -force
```
- 4 Make a backup copy of the `main.cf` file:

```
# cd /etc/VRTSvcs/conf/config  
# cp main.cf main.cf.orig
```
- 5 Edit the `main.cf` file to include the `SybaseTypes.cf` file:

```
include "SybaseTypes.cf"
```


At this point, the Sybase types definition has been imported to the VCS engine. The agent for Sybase can be configured without interrupting or stopping VCS.

Configuring the service group from Cluster Manager (Java Console)

A template for the Sybase resource groups is automatically installed with the Veritas high availability agent for Sybase. Using the VCS Cluster Manager (Java Console), you can view the template, which displays the Sybase service group, its resources and their attributes. You can dynamically modify the attributes' values as necessary for your configuration.

To configure a service group from the Java Console

- 1 Make sure the Sybase type definition file `SybaseTypes.cf` is imported in your configuration. See [“Importing the SybaseTypes.cf file”](#) on page 24 for instructions.
- 2 Launch the Cluster Configuration wizard using any of the following two ways:
 - From the Cluster Explorer menu, select **Tools > Configuration Wizard**.
 - If no service groups exist on the system, **Cluster Explorer** prompts you to launch the **Cluster Configuration wizard**. Click **Yes** when prompted. The Loading Templates Information window appears, and launches the wizard.
- 3 Review the information in the Welcome dialog box and click **Next**.
- 4 Specify the name of the service group and the target systems on which the service group will be configured.
 - a Enter the name of the service group.
 - b From the **Available Systems** box, select the systems on which to configure the service group.
 - c Click the right arrow to move the selected systems to the **Systems for Service Group** box. To remove a system from the box, select the system and click the left arrow.
 - d Specify system priority for the service group to failover. System priority is numbered sequentially, with the lowest assigned number denoting the highest priority.
 - e Select the **Service Group Type** as *Failover* and click **Next**.

- 5 On the Would you like to use a template to configure the service group? dialog box, click **Next** to configure the service group using a template.
- 6 Select the **SybaseGroup** template to configure a Sybase service group. If applicable, a window opens notifying that names of some resources within the new service group are already in use. Resolve the name clashes, if any and click **Next**.
- 7 Click **Next** to create the service group based on the selected template. A progress indicator displays the percentage of the commands executed to create the service group. The actual commands are displayed at the top of the indicator.
- 8 After the service group is created, click **Next** to edit the attributes for the resources.
- 9 The left pane in the dialog box lists all the resources for the Sybase service group. Select a resource from the left pane to list the attributes on the right pane. The attributes in bold denote mandatory attributes.
- 10 See “[Editing resource attributes](#)” on page 26 for instructions on modifying the attribute values. Click **Finish** to accept the default values and complete the configuration.

Editing resource attributes

Edit the resource attributes to modify the values of the resources.

To edit resource attributes

- 1 Select the resource from the list on the left pane. The resource attributes appear in the right pane.
- 2 Select the attribute to be modified and click the edit icon in the **Edit** column.
- 3 In the Edit Attribute dialog box, enter the attribute values. To modify the scope of the attribute, choose the **Global** or **Local** option.
- 4 Click **OK**.
- 5 Repeat the procedure for each resource and click **Finish**. Edit the attributes for all the resources according to your configuration.

Caution: For added security, you must always provide a secure value for passwords. See “[Encrypting passwords](#)” on page 28 for instructions on setting secure passwords.

- 6 Follow the wizard instructions to complete the configuration. Click **Finish** to quit the wizard.

Configuring the service group from the command line

The Veritas high availability agent for Sybase comes with a sample configuration file that can be used as reference to directly modify your present configuration file. This method requires you to restart VCS before the configuration takes effect.

To configure a service group from the command line

- 1 Log in to a cluster system as `root`.
- 2 Make sure the Sybase type definition is imported into VCS engine. See [“Importing the SybaseTypes.cf file”](#) on page 24 for instructions.
- 3 Edit the `main.cf` file. Use the file `/etc/VRTSagents/ha/conf/Sybase/sample_main.cf` for reference:
 - a Create a Sybase service group.
 - b Create the Sybase and SybaseBk resources. Refer to the type definitions under [“Sybase resource type”](#) on page 38 and [“SybaseBk resource type”](#) on page 41.
 - c Edit the default attributes to match the parameters in your configuration.

Caution: For added security, you must always provide a secure value for passwords. See [“Encrypting passwords”](#) on page 28 for instructions on setting secure passwords.

- d Assign dependencies to the newly created resources. Refer to the sample file `/etc/VRTSagents/ha/conf/Sybase/sample_main.cf`. (See the *Veritas Cluster Server User's Guide* for more information on assigning dependencies.)
- 4 Save and close the file.
 - 5 Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

```
# hacf -verify config
```
 - 6 Start VCS on local node:

```
# hstart
```
 - 7 Start VCS on other nodes.
 - 8 Verify all Sybase service group resources are brought online:

```
# hagr -state
```

- 9 Take the service group offline and verify that all resources are stopped:

```
# hagr -offline <service_group> -sys <system_name>
# hagr -state
```
- 10 Bring the service group online again and verify that all resources are available:

```
# hagr -online <service_group> -sys <system_name>
# hagr -state
```
- 11 Start the VCS engine on the other node:

```
# hstart
```
- 12 Switch the Sybase service group to the other node:

```
# hagr -switch <service_group> -to <system_name>
```
- 13 Verify that all Sybase service group resources are brought online on the other node:

```
# hagr -state
```
- 14 On all systems, look at the following log files for any errors or status:

```
/var/VRTSvcs/log/engine_A.log
/var/VRTSvcs/log/Sybase_A.log
/var/VRTSvcs/log/SybaseBk_A.log
```

Encrypting passwords

VCS provides a `vcseencrypt` utility to encrypt user passwords. Encrypt passwords before specifying them for Sybase and SybaseBk resource type definition.

Note: You need not encrypt passwords when using the VCS Cluster Manager (Java Console) to configure attributes.

To encrypt passwords

- 1 From the path `$VCS_HOME/bin/`, run the `vcseencrypt` utility.
 - a Type the following command:

```
# vcseencrypt -agent
```
 - b The utility prompts you to enter the password twice. Enter the password and press Return.

```
# Enter New Password:
# Enter Again:
```
- 2 The utility encrypts the password and displays the encrypted password.
- 3 Enter this encrypted password as the value for the attribute.
Copy the encrypted password for future reference.

Setting up detail monitoring for the agent

The Veritas high availability agent for Sybase provides two levels of application monitoring: basic and detail. In basic monitoring, Sybase resource monitors the Sybase daemon processes to verify that they are continuously active.

In detail monitoring, the Sybase resource performs transactions on a test table in the database to ensure that the Sybase server is functioning properly. The agent uses the script defined in the attribute `Monscript` of the Sybase resource. During detail monitoring, the agent executes the specified script. If the script successfully executes, the agent considers the database available. You can customize the default script according to your configurations.

To activate detail monitoring, the `DetailMonitor` attribute must be set to a positive integer and `User`, `UPword`, `Db`, and `Table` attributes must not be empty (""). The attribute `Monscript`, which contains the path of the detail monitor script, must also exist and have execute permissions for the `root`.

Enabling detail monitoring

Perform the following steps to enable detail monitoring on a database.

To enable detail monitoring

- 1 Make sure Sybase server is configured for detail monitoring. See [“Configuring Sybase for detail monitoring”](#) on page 17 for instructions.
- 2 Make the VCS configuration writable:

```
# haconf -makerw
```
- 3 Enable detail monitoring for Sybase:

```
# hares -modify <Sybase_resource> DetailMonitor 1
# hares -modify <Sybase_resource> User <user>
# hares -modify <Sybase_resource> UPword <encrypted_password>
# hares -modify <Sybase_resource> Db <database_name>
# hares -modify <Sybase_resource> Table <table_name>
# hares -modify <Sybase_resource> Monscript
      "/opt/VRTSagents/ha/bin/Sybase/SqlTest.pl"
```
- 4 Save the configuration:

```
# haconf -dump [-makero]
```

Note: If detail monitoring is configured and the database is full, the SQL queries take considerable time to commit the results. In such a case, the monitor routine for the agent fails and attempts to fail over the service group. This issue is not encountered if detail monitoring is not configured.

Disabling detail monitoring

Perform the following steps to disable detail monitoring

To disable detail monitoring

- 1 Make the VCS configuration writable:
`# haconf -makerw`
- 2 Disable detail monitoring for Sybase:
`# hares -modify <Sybase_resource> DetailMonitor 0`
- 3 Save the configuration:
`# haconf -dump [-makero]`

Administering the Sybase service group

This chapter contains the following topics:

- [Bringing the service group online](#)
- [Taking the service group offline](#)
- [Switching the service group](#)
- [Viewing the agent log](#)
- [Modifying the service group configuration](#)
- [Disabling the agent](#)
- [Removing the agent](#)

Bringing the service group online

Perform the following steps to bring a service group online

To bring a service group online

- 1 In the Cluster Explorer configuration tree, select the newly created service group.
- 2 Right-click the service group name, and select **Enable Resources**. This enables all the resources in the group.
- 3 Right-click the service group name, and select the systems on which to enable the service group (Right-click>Enable>*system_name* or Right-click>Enable>All).
- 4 Save your configuration (File>Close Configuration).

- 5 Right-click the service group and select to online the service group on the system (Right-click>Online>*system_name*).

Note: In the initial few cycles of bringing a service group online, the memory usage by the agent for Sybase increases by a noticeable value.

Taking the service group offline

Perform the following steps to take a service group offline

To take a service group offline

- 1 On the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.
or
Select the cluster in the Cluster Explorer configuration tree, select the **Service Groups** tab, and right-click the service group icon in the view panel.
- 2 Choose **Offline**, and choose the appropriate system from the pop-up menu (Right-click>Offline>*system_name*).

Note: In the initial few cycles of taking a service group offline, the memory usage by the agent for Sybase increases by a noticeable value.

Switching the service group

The process of switching a service group involves taking it offline on its current system and bringing it online on the another system.

To switch a service group

- 1 On the **Service Groups** tab of Cluster Explorer configuration tree, right-click the service group.
or
Select the cluster in the Cluster Explorer configuration tree, select the **Service Groups** tab, and right-click the service group icon in the view panel.
- 2 Choose **Switch To**, and choose the appropriate system from the pop-up menu (Right-click>Switch To>*system_name*).

Viewing the agent log

The Veritas high availability agent for Sybase logs messages to the following files:

```
/var/VRTSvcs/log/engine_A.log  
/var/VRTSvcs/log/Sybase_A.log  
/var/VRTSvcs/log/SybaseBk_A.log
```

Modifying the service group configuration

You can dynamically modify the Veritas high availability agent for Sybase using several methods, including the Cluster Manager (Java Console), Cluster Manager (Web Console), and the command line. See the *Veritas Cluster Server User's Guide* for more information.

Disabling the agent

To disable the agent on a system, you must first take the Sybase service group offline. You can stop the application completely, or switch the service group to another system.

To disable an agent

- 1 Determine if the service group is online by entering:

```
# hagr -state <service_group> -sys <system_name>
```
- 2 If the service group is online, take it offline by entering:

```
# hagr -offline <service_group> -sys <system_name>
```

or

```
# hagr -switch <service_group> -to <system_name>
```
- 3 Stop the agents on the system by entering:

```
# haagent -stop Sybase -sys <system_name>  
# haagent -stop SybaseBk -sys <system_name>
```

When you get the message “Please look for messages in the log file,” check the file `/var/VRTSvcs/log/engine_A.log` for a message confirming the agent has stopped. You can also use the `ps` command to verify whether the agent is stopped.

You can remove the system, service group, or resource type from the VCS configuration after disabling the agent on all systems. See the chapter on reconfiguring VCS from the command line in the *Veritas Cluster Server User's Guide* for more information.

Removing the agent

This section provides steps to remove the agent from the cluster.

To remove the agent

- 1 Take the Sybase and SybaseBk resources offline.
- 2 Stop the agent for Sybase.

```
# haagent -stop Sybase -<system>
```

Perform this step on all nodes where the agent for Sybase is running.
- 3 Stop the agent for SybaseBk.

```
# haagent -stop SybaseBk -<system>
```

Perform this step on all nodes where the agent for SybaseBk is running.
- 4 Remove the agent from all nodes in the cluster.

```
# pkgrm VRTSvcsy
```

Answer prompts accordingly.

Resource type definition

This appendix contains the following topics:

- [About the resource type and attribute definitions](#)
- [Sybase resource type](#)
- [SybaseBk resource type](#)

This appendix lists resource type definitions and attribute definitions of the agents for Sybase and SybaseBk.

About the resource type and attribute definitions

The resource type represents the VCS configuration definition of the agent and specifies how the agent is defined in the configuration file main.cf. The Attribute Definitions table explains the attributes associated with the agent. The Required attributes table lists the attributes that must be configured for the agent to function properly.

Sybase resource type

```
type Sybase (
  static str AgentDirectory = "/opt/VRTSagents/ha/bin/Sybase"
  static str ContainerType = Zone
  static str ArgList[] = { Server, Owner, Home, Version, SA,
                          SApswd, User, UPword, Db, Table,
                          Monscript, DetailMonitor }

  str Server
  str Owner
  str Home
  str Version
  str SA
  str SApswd
  int DetailMonitor = 0
  str User
  str UPword
  str Db
  str Table
  str Monscript="/opt/VRTSagents/ha/bin/Sybase/SqlTest.pl"
  str ContainerName
)
```

Attribute definitions

Required Attributes	Type and Dimension	Definition
Server	string-scalar	The \$DSQUERY ASE name. Only one server must be configured in a Sybase service group.
Owner	string-scalar	Sybase user as defined <code>/etc/passwd</code> . The Sybase executables and database files are accessed in the context of this user. Make sure you specify the same user created under “ System user for Sybase home directory ” on page 14.
Home	string-scalar	The \$SYBASE path to Sybase binaries and configuration files.
Version	string-scalar	Version of Sybase ASE.
SA	string-scalar	Sybase database administrator. This is required to connect to the ASE for shutdown.
SAPswd	string-scalar	Encrypted password for Sybase database administrator. This password is required to connect to the ASE for shutdown. See “ Encrypting passwords ” on page 28 for instructions on encrypting passwords. Note: You need not specify a value for this attribute if the SA user does not require a password.

Optional Attributes	Type and Dimension	Definition
AgentDirectory	static-string	This attribute is for internal use only. Specifies the location of the binaries, scripts, and other files related to the agent for Sybase. Symantec recommends not to modify the value of this attribute.
ContainerType	static-string	Setting this attribute to “Zone” indicates that the agent for Syabase monitors Sybase instances running in Solaris 10 zones. Leaving this attribute blank indicates it does not.
DetailMonitor	int-scalar	Specifies whether the Sybase server is monitored in detail. Value 1 indicates that the resource monitors the Sybase server in detail. Value 0 denotes it does not. Default is 0.

Optional Attributes	Type and Dimension	Definition
User	string-scalar	<p>The database user, in the context of which, the transactions are performed on the database. Make sure you specify the user name created in step 5 on page 17.</p> <p>Note: You must specify a value for this attribute if DetailMonitor is set to a non-zero value.</p>
UPword	string-scalar	<p>Encrypted password for the database user. Make sure you provide the password specified in step 5 on page 17. See “Encrypting passwords” on page 28 for instructions on encrypting passwords.</p> <p>Note: You must specify a value for this attribute if DetailMonitor is set to a non-zero value. However, you need not specify a value for this attribute if the database user does not require a password.</p>
Db	string-scalar	<p>Name of the database, in which the table resides, on which the detail monitor script will perform the transactions. Make sure you specify the name of the database created in step 6 on page 17.</p> <p>Note: You must specify a value for this attribute if DetailMonitor is set to a non-zero value.</p>
Table	string-scalar	<p>Name of the table on which the detail monitoring script will perform the transactions. Make sure you specify the name of the table created in step 11 on page 18.</p> <p>Note: You must specify a value for this attribute if DetailMonitor is set to a non-zero value.</p>
Monscript	string-scalar	<p>The Path to the detail monitor script; the default value for this attribute is the path for the script, SqlTest.pl, provided with the agent.</p> <p>Note: You must specify a value for this attribute if DetailMonitor is set to a non-zero value.</p>
ContainerName	string-scalar	<p>Specifies the name of the zone where the Sybase instance to be monitored is running.</p>

SybaseBk resource type

```

type SybaseBk (
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/
        SybaseBk"
    static str ContainerType = Zone
    static str ArgList [] = { Backupserver, Owner, Home, Version,
        Server, SA, SApswd }

    str Server
    str Owner
    str Home
    str Version
    str Backupserver
    str SA
    str SApswd
    str ContainerName
)

```

Attribute definitions

Attributes	Type and Dimension	Definition
AgentDirectory	static-string	This attribute is for internal use only. Specifies the location of the binaries, scripts, and other files related to the agent for SybaseBk. Symantec recommends not to modify the value of this attribute.
ContainerType	static-string	Setting this attribute to "Zone" indicates that the agent for SybaseBk monitors SybaseBk instances running in Solaris 10 zones. Leaving this attribute blank indicates it does not.
Server	string-scalar	The \$DSQUERY Backup server name.
Owner	string-scalar	Sybase user as defined /etc/passwd. The Sybase executables and database files are accessed in the context of this user. Make sure you specify the same user created under " System user for Sybase home directory " on page 14.
Home	string-scalar	The \$SYBASE path to Sybase binaries and configuration files.
Version	string-scalar	Version of Sybase Backup Server.
Backupserver	string-scalar	The \$BACKUP SYBASE Backup Server name.

Attributes	Type and Dimension	Definition
SA	string-scalar	Sybase database administrator. This is required to connect to the ASE for shutdown.
SAPswd	string-scalar	<p>Encrypted password of Sybase database administrator. This password is required to connect to the ASE for shutdown. See “Encrypting passwords” on page 28 for instructions on encrypting passwords.</p> <p>Note: You need not specify a value for this attribute if the SA user does not require a password.</p>
ContainerName	string-scalar	Specifies the name of the zone where the SybaseBk instance to be monitored is running.

Sample configuration

This appendix contains the following topics:

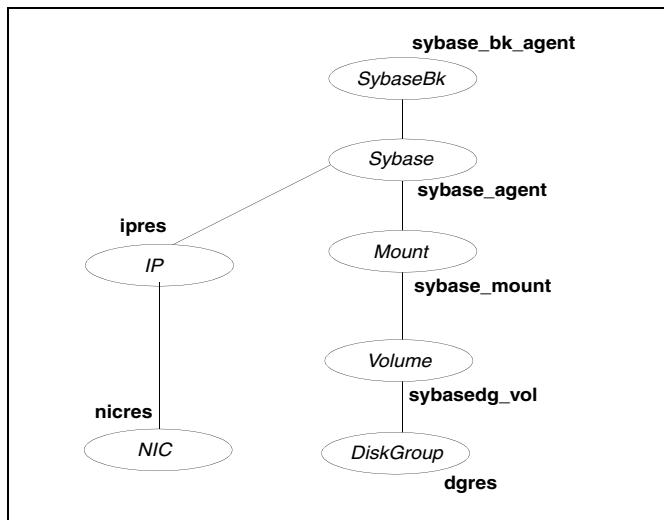
- [About the sample configuration for Sybase agent](#)
- [Resource dependency graph](#)
- [Sample configuration](#)

About the sample configuration for Sybase agent

This appendix describes a typical service group configured to monitor the state of Sybase servers in a VCS cluster. The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agent. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

Resource dependency graph

The figure below illustrates the configuration's dependency graph.



This dependency graph depicts a single Sybase instance configuration. The configuration contains a disk group with a single volume. The volume is monitored using a Volume resource and mounted using a Mount resource. The Mount resource requires Volume resource, which in turn requires the DiskGroup resource. The service group IP address for Sybase server is monitored using the IP and NIC resource types. The Sybase server can be started after each of these resources are brought online. The Backup Server is started after the Sybase SQL Server is online.

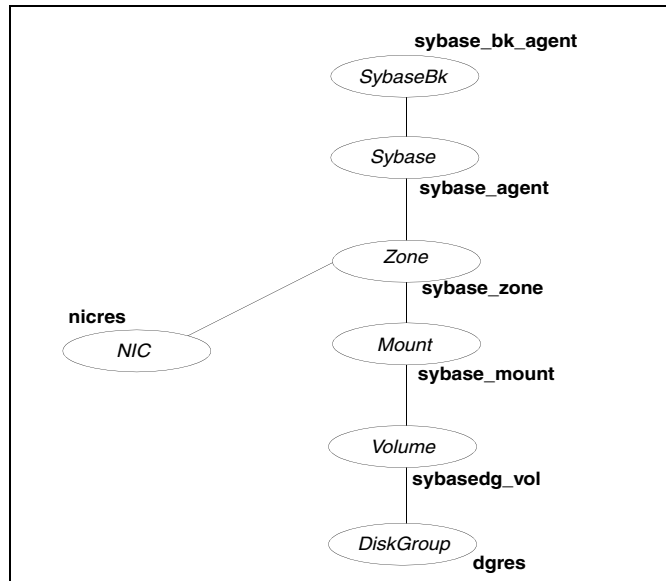
Note: If your configuration does not use VERITAS Volume Manager, use the LVMVolumeGroup and LVMLogicalVolume resource types to configure shared storage instead of DiskGroup and Volume resource types.

Resource dependency for Sybase configured in Solaris zones

The following examples describe a typical service group configured to monitor the state of a Sybase instance configured in a Solaris zone.

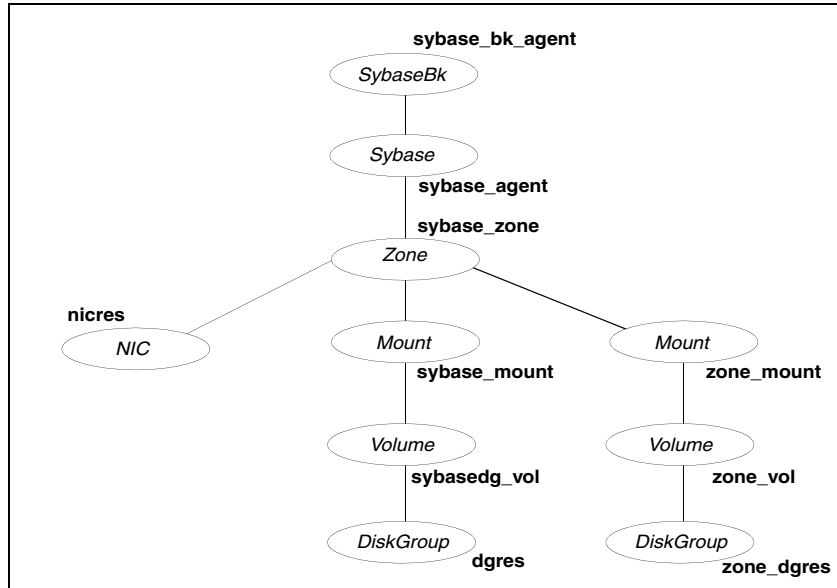
Zone root on local disk

If the root file system of a zone is on the local disk of each node, the file system is mounted when the system is booted. Hence, the service group does not need to have separate DiskGroup and Volume resources for the zone.



Zone root on shared disk

If the root file system of a zone is on a shared disk, the file system should be mounted by VCS. Hence, separate DiskGroup and Volume resources are required for the zone.



Sample configuration

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Sybase/SybaseTypes.cf"

cluster mycluster (
  UserNames = { admin = anoGniNkoJooMwoIn1 }
  Administrators = { admin }
  CounterInterval = 5
)

system system1 (
)

system system2 (
)

group Sybase_group (
  SystemList = { system1 = 0, system2 = 1 }
  AutoStartList = { system1, system2 }
)

DiskGroup dgres (
  DiskGroup = sybasedg
)

IP ipres (
  Device = le0
  Address = "192.168.175.28"
  NetMask = "255.255.0.0"
)

NIC nicres (
  Device = le0
  NetworkType = ether
)

Mount sybase_mount (
  MountPoint = "/opt/Sybase"
  BlockDevice = "/dev/vx/dsk/sybasedg/ase1103"
  FsckOpt = "-n"
  FSType = vxfs
  MountOpt = rw
)

Sybase sybase_agent (
  Server = SYBASE
  Owner = sybase
  Home = "/opt/Sybase"
  Version = sybasel1
  SA = sa
)
```

```
SApswd = sybase
User = abcd
UPword = xxxxxxxx
Db = testdb1
Table = testtab
Monscript = "/opt/VRTSagents/ha/bin/Sybase/SqlTest.pl"
)

SybaseBk sybase_bk_agent (
  Server = SYBASE
  Owner = sybase
  Home = "/opt/Sybase"
  Version = sybase11
  Backupserver = SYB_BACKUP
  SA = sa
  SApswd = sybase
)

Volume sybasedg_vol (
  Volume = ase1103
  DiskGroup = dgres
)

sybase_agent requires sybase_mount
sybase_agent requires ipres
sybase_bk_agent requires sybase_agent
sybase_mount requires sybasedg_vol
ipres requires nicres
sybasedg_vol requires dgres
```

Sample configuration in Solaris zones

A resource of type Zone will be added to the sample configuration if the Sybase agent is configured to run in Solaris zones, irrespective of whether the root file system of the zone is on local or shared disks.

```
Zone ORA_zone (
  ZoneName = z-ora
)
```

Along with this resource addition, the service group configuration would also change to have the NIC and Mount resources to depend on the Zone resource.

Index

A

- agent for Backup server
 - attribute definitions 41
 - operations 9
 - resource type 41
- agent for Sybase Server
 - attribute definitions 39
 - operations 8
 - resource type 38
- agent operations
 - agent for Backup Server 9
 - agent for Sybase Server 8
- Attribute definitions
 - agent for Backup server 41
 - agent for SQL server 39
- attributes
 - editing 26
 - modifying 26

B

- Backup server agent attributes
 - AgentDirectory 41
 - Backupserver 41
 - Home 41
 - Owner 41
 - SA 42
 - SAPswd 42
 - Server 41
 - Version 41

C

- cluster manager 25
- configuration
 - typical setup 11
- configuring the service group
 - from Cluster Manager (Java Console) 25
 - from the command line 27

D

- database tables 17
- detail monitoring
 - configuring Sybase 17
 - disabling 31
 - enabling 30
- Detail monitoring,setting up 30

E

- editing attributes 26
- encrypting passwords 28
- enterprise agents
 - about 8
 - configuring using Cluster Manager 25
 - installing 19
 - monitoring in detail 30
 - removing 36
 - supported software 7
 - typical setup 11
 - upgrading 21

F

- failover 11

M

- monitoring options 10

O

- offlining service group 34
- onlining service group 33
- operations
 - agent for Backup Server 9
 - agent for Sybase Server 8

P

- passwords, encrypting 28

R

- removing the agent 36
- resource attributes, editing 26
- resource types
 - agent for Backup Server 41
 - agent for Sybase Server 38

S

- service groups
 - disabling 35
 - offlining 34
 - onlining 33
 - removing 36
 - switching 34
 - viewing log 35
- setting secure passwords
 - using vcsencrypt 28
- supported software 7
- switching service groups 34
- Sybase
 - configuration in a cluster 11
 - database dbspaces 14
 - directory on shared disks 14
 - installation 13
 - language settings 16
 - long pathname limitations 14
 - setting up for detail monitoring 17
 - transparent TCP/IP failover 14
- Sybase Server agent attributes
 - AgentDirectory 39
 - Db 40
 - DetailMonitor 39
 - Home 39
 - Monscript 40
 - Owner 39
 - SA 39
 - SAPswd 39
 - Server 39
 - Table 40
 - UPword 40
 - User 40
 - Version 39

V

- vcsencrypt utility 28

U

- uninstalling the agent 36