Cluster Server 7.4.3 Agent for Sybase Installation and Configuration Guide - Linux

Last updated: 2020-11-17

Legal Notice

THE 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. VERITAS TECHNOLOGIES LLC 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 as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054
http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:
https://www.veritas.com/support

You can manage your Veritas account information at the following URL:
https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:
https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:
infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:
http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:
https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1

Introducing the Cluster Server Agent for Sybase

This chapter includes the following topics:

About the Cluster Server Agent for Sybase

The Cluster Server Agent for Sybase brings the configured Sybase servers online, monitors them, and takes them offline.

For Sybase ASE enterprise edition, the following agents work together to make Sybase highly available in a VCS cluster.

- 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.

For Sybase ASE cluster edition, the VCS agent for Sybase makes Sybase adaptive server highly available in a VCS cluster. The VCS agent for Sybase backup server (SybaseBk agent) is not supported for Sybase ASE cluster edition.

Note: For ASE Enterprise Edition, the VCS agent for Sybase provides "active/passive" support. For "active/active" support for ASE Enterprise Edition, contact Sybase for their agent.

Supported software for the VCS agent for Sybase

Refer to the *Veritas InfoScale 7.4.3 Release Notes* for information on supported software and platform versions.

Refert to the Sybase agent support matrix at https://www.veritas.com/support/en_US/article.000126342.

How the agent makes Sybase highly available

The Cluster Server agent for Sybase continuously monitors the Sybase database and SybaseBk processes to verify whether they function properly.

The agent for Sybase can perform different levels of monitoring and different actions which you can configure.

- Primary or Basic monitoring: In the basic monitoring mode, the agent monitors the configured Sybase server process, and fails over or restarts the group during an application failure.
- Detail monitoring: In the optional detail monitoring mode, the agent detects application failure if it cannot perform a transaction in the test table in the Sybase database server.
- Intelligent monitoring framework (IMF): The Sybase agent is IMF-aware and uses asynchronous monitoring framework (AMF) kernel driver for resource state change notifications.

See "How the Sybase and SybaseBk agents support intelligent resource monitoring " on page 9.

For Sybase cluster edition, the agent uses the Sybase-provided utility, `qrmutil`, to know if the status of the instance is up or down. If `qrmutil` reports the status as failure pending, the agent reboots the node and the instance is automatically started again.

When the Sybase agent detects that the configured Sybase server is not running on a system, based on the value of OnlineRetryLimit attribute of the Sybase server group, the service group is restarted on the same system on which the group faulted.

For ASE Enterprise edition 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.

How the Sybase and SybaseBk agents support intelligent resource monitoring

With intelligent monitoring framework (IMF), VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring.

You can enable or disable the intelligent resource monitoring functionality of the VCS agents for Sybase and SybaseBk using `haimfconfig` script or manually.

See "Enabling and disabling intelligent resource monitoring (IMF) for agents" on page 38.

When an IMF-enabled agent starts up, the agent initializes the asynchronous monitoring framework (AMF) kernel driver. After the resource is in a steady state, the agent registers the details that are required to monitor the resource with the AMF kernel driver. For example, the Sybase agent registers the PIDs of the processes with the AMF kernel driver. The agent's `imf_getnotification` function waits for any resource state changes. When the AMFkernel driver module notifies the `imf_getnotification` function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS which takes the appropriate action.

See the *Cluster Server Administrator's Guide* for more information.

About the Sybase agent functions

The functions an agent performs are called entry points. The Cluster Server agent for Sybase can perform different operations or functions on the database. These functions are online, offline, monitor, clean, and action.

Review the functions for the following agents that are part of the Cluster Server agent suite for Sybase:

- Sybase (SQL server) agent functions
  See "Sybase agent functions " on page 10.
- SybaseBk (Backup Server) agent functions
  See "SybaseBk agent functions" on page 12.

Sybase agent functions

The agent for Sybase starts a Sybase ASE dataserver, monitors the server processes, shuts down the server, and restarts the server processes..

The Sybase agent is IMF-aware.

Table 1-1 lists the Sybase agent for SQL server functions.

Table 1-1 Sybase agent functions

Agent function	Description
Online	Starts the Sybase ASE dataserver by using the following command. `startserver -f $SYBASE/$SYBASE_ASE/install/RUN_$Server` where $Server is the instance_name, and `$SYBASE/$SYBASE_ASE/install/RUN_$Server` is the default location of the Run server file. If you specify the value of the Run_ServerFile attribute, then the value that you specify is used instead of the default location. If the WaitForRecovery attribute is enabled, the agent waits either till recovery has been completed and all databases that can be made online are brought online. The agent queries the recovery status by connecting to the `isql` session. The OnlineTimeout attribute must be set to a sufficiently large value so that the recovery completes before the OnlineTimeout is reached. By default, the WaitForRecovery attribute is not enabled. If the interfaces file location is specified using the interfaces_File attribute, agent uses [-I interfaces file] option while connecting to the isql session. When DelayAfterOnline attribute is set, the monitor function is invoked after completion of online function, and after the number of seconds specified in DelayAfterOnline attribute have elapsed.

Table 1-1 Sybase agent functions *(continued)*

Agent function	Description
Monitor	In the basic monitoring mode, agent scans the process table for the dataserver process. In the detail monitoring mode, agent runs the script that is specified in MonScript as an option. See "Monitoring options for the Sybase agent" on page 15.
Offline	Stops the Sybase SQL server by using the `isql` command in the following manner. If interfaces file location is specified using the interfaces_File attribute, agent uses the specified file while connecting to isql session. The agent first executes the `shutdown with wait` command. Sybase agent uses the `timeout` option during shutdown of Sybase dataserver if this option is supported. For Sybase ASE Enterprise edition the timeout option for shutdown command is supported for versions 12.5.4 and 15.0.2 onwards. If this timeout option is not supported for Sybase ASE Enterprise edition, the offline script waits in a loop till the dataserver completely stops. The agent waits for up to the OfflineTimeout duration. If the process is still running, the offline script issues a `shutdown with nowait` command. When DelayAfterOffline attribute is set, the monitor function is invoked after completion of offline function and after the number of seconds specified in DelayAfterOffline attribute have elapsed.
imf_init	Initializes the agent to interface with the AMF kernel driver, which is the IMF notification module for Sybase agent. This function runs when the agent starts up.
imf_register	Registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into steady state (online or offline).
imf_getnotification	Gets notification about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

Table 1-1 Sybase agent functions *(continued)*

Agent function	Description
Clean	Forcefully stops the Sybase SQL server by using the `isql` command in the following manner. The agent first executes the `shutdown with wait` command. For Sybase ASE Enterprise edition, if the `isql` command fails, the clean script executes the `shutdown with nowait` command. If the process does not respond to the `shutdown` command, the agent kills the Sybase dataserver process.
Action	Performs the predefined actions on a resource. See "Action function for the VCS agent for Sybase" on page 15.

SybaseBk agent functions

The agent for SybaseBk starts a Sybase Backup server, monitors the server process, and shuts down the server.

The SybaseBk agent is IMF-aware.

Table 1-2 lists the Sybase agent for Backup server functions.

Table 1-2 Sybase agent for Backup server functions

Agent operation	Description
Online	Starts the Sybase Backup server by using the following command. `startserver -f $SYBASE/$SYBASE_ASE/install/RUN_$BackupServer` Where `$SYBASE/$SYBASE_ASE/install/RUN_$BackupServer` is the default location of the Run server file. If you specify the value of the Run_ServerFile attribute, then the value that you specify is used instead of the default location.
Monitor	Scans the process table for the backupserver process.

Table 1-2 Sybase agent for Backup server functions *(continued)*

Agent operation	Description
Offline	Stops the Sybase Backup server by using the `isql` command in the following manner. If interfaces file location is specified using the interfaces_File attribute, agent uses the specified file while connecting to `isql` session. The agent first executes the command `shutdown SYB_BACKUP with wait`. If this command 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 command 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 the processes that are associated with the configured Sybase Backup server and kills them.
imf_init	Initializes the agent to interface with the AMF kernel driver, which is the IMF notification module for SybaseBk agent. This function runs when the agent starts up.
imf_ register	Registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into steady state (online or offline).
imf_getnotification	Gets notification about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

Using the IPC Cleanup feature for the Sybase agent

When the Adaptive Server starts, it creates shared memory files in $SYBASE to store information about the shared memory segments that it uses. Adaptive Server start-up parameter -M can be used to change the location of directory that stores shared memory files. The start-up parameter -M should be updated in RUN_$Server file.

If the Sybase home directory is unmounted, the Sybase clean script cannot access the shared memory files and does not clean the IPC resources that are allocated by the Sybase processes. Hence, the agent requires shared memory files to be present in the following directory on local system `/var/VRTSvcs/log/tmp/sybase_shm/$Server`.

In the $SYBASE/$SYBASE_ASE/install directory, edit the RUN_$Server file. Change the location of the directory that stores shared memory files to `/var/VRTSvcs/log/tmp/sybase_shm/$Server` using the -M option.

For example, the file RUN_Sybase_Server resembles the following before the change:

```
/home/sybase/ASE-15_0/bin/dataserver \

-sSybase_Server \

-d/home/sybase/data/master.dat \

-e/home/sybase/ASE-15_0/install/Sybase_Server.log \

-c/home/sybase/ASE-15_0/Sybase_Server.cfg \

-M/home/sybase/ASE-15_0 \
```

After the replacement, the file resembles:

```
/home/sybase/ASE-15_0/bin/dataserver \

-sSybase_Server \

-d/home/sybase/data/master.dat \

-e/home/sybase/ASE-15_0/install/Sybase_Server.log \

-c/home/sybase/ASE-15_0/Sybase_Server.cfg \

-M/var/tmp/sybase_shm/Sybase_Server \
```

Here Sybase_Server is the name of the Adaptive server.

Note: Make sure you create the /var/VRTSvcs/log/tmp/sybase_shm/Sybase_Server directory with proper permissions.

Monitoring options for the Sybase agent

The VCS agent for Sybase provides two levels of application monitoring: basic and detail.

In the basic monitoring mode, the agent for Sybase monitors the Sybase dataserver processes to verify whether they are running.

For Sybase cluster edition, the agent uses `qrmutil` utility that Sybase provides to get the status of the Sybase instance. If the state returned by `qrmutil` utility is 'failure pending', the agent panics the node. When the Sybase agent detects that the configured Sybase server is not running on a system, based on the value of the OnlineRetryLimit attribute of the Sybase service group, the service group is restarted on the same system on which the group faulted.

For example:

```
# qrmutil --quorum_dev=/quorum/quorum.dat --monitor=ase1
Executing 'monitor' command for instance 'ase1'
Instance 'ase1' has a failure pending.
# echo $?
99
```

In this example instance ‘ase1’ has a failure pending state. The agent will panic the node running the instance ‘ase1’. The node will automatically rejoin the cluster after reboot.

In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Sybase server is funtioning properly. The test table should be created by the user, and the table is specified in the attribute Table for the Sybase agent. The agent uses this test table for internal purposes. Veritas recommends that you do not perform any other transaction on the test table.

See “About setting up detail monitoring for the agent for Sybase” on page 36.

For the Sybase ASE Enterprise edition, 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.

Action function for the VCS agent for Sybase

The agent for Sybase supports the Action function, which enables you to perform predefined actions on a resource.

To perform an action on a resource, type the following command:

```
# hares -action res token [-actionargs arg1 ...] \
[-sys system] [-clus cluster]
```

You can also add custom actions for the agent.

For more information, refer to the *Cluster Server Agent Developer's Guide*.

Table 1-3 describes the agent's predefined action.

Table 1-3 Predefined agent action

Action	Description
checkpoint_all	Performs "checkpoint all" for the Sybase dataserver by connecting to isql session. **Note:** The checkpoint_all action function is not applicable for Sybase Cluster Edition.

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 the nodes. The shared disks can be managed using Veritas Volume Manager (VxVM).

Figure 1-1 illustrates a sample configuration in which the Sybase servers, including binaries and data are installed completely on shared disks or shared cluster disk groups managed using VxVM.

Figure 1-1 Sybase binaries and data on shared disks

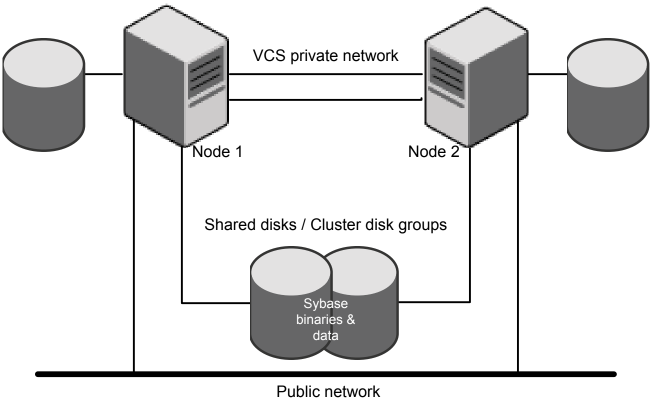
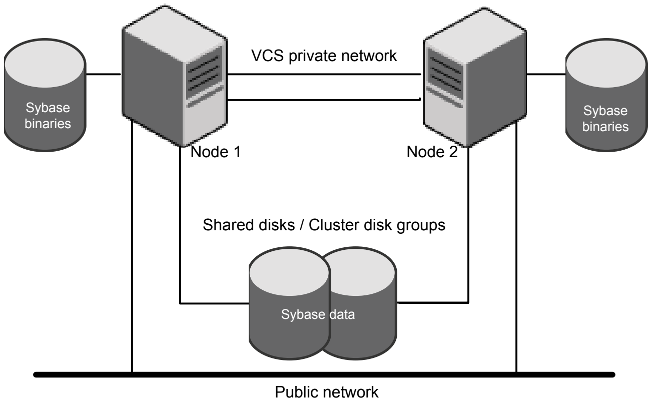


Figure 1-2 illustrates a sample configuration in which Sybase binaries are installed locally on each node in the cluster and the Sybase data is on shared disks or shared cluster disk groups managed using VxVM.

Figure 1-2 Binaries on local disk and Sybase data on shared disks



Chapter 2

Installing and configuring Sybase

This chapter includes the following topics:

VCS requirements for installing Sybase

Review the following requirements before you install Sybase in a VCS cluster. Before installing Sybase, make sure that the systems in the cluster have adequate resources to run Sybase and VCS.

Sybase installation directory

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

Review the following prerequisites:

- If the Sybase binaries are installed on a local disk, verify that 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.

For Sybase ASE cluster edition, use Storage Foundation clustered file system (SFCFS) for the Sybase ASE quorum device.

System Group and User for Sybase home directory

Make sure you have a system group and user, with the same name 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:

```
# groupadd -g group_id group_name
# useradd -u user_id  -g group_name user_name
# chown -R user_name:group_name $SYBASE
```

Database dbspaces

If you use 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 use 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.

Language settings for the Sybase agent

For the VCS 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

LC_CTYPE=en_US.UTF-8
export LC_CTYPE

LC_ALL=en_US.UTF-8
export LC_ALL
```

This change affects all the agents that are configured on the nodes.

Installing Sybase in a VCS environment

For information on how to install Sybase, refer to Sybase documentation.

Configuring Sybase for detail monitoring

This section describes the tasks to be performed to configure a Sybase server for detail monitoring.

See “About setting up detail monitoring for the agent for Sybase” on page 36.

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

Perform these steps only once in a Sybase cluster.

To configure Sybase for detail monitoring

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.

   ```
   # startserver -f ./$SYBASE/$SYBASE_ASE/install/RUN_server_name
   ```

3 Start the Sybase client on any cluster node.

```
# isql -Usa -SSYBASE_SERVER_NAME
```

Enter the administrator password when prompted to do so.

4 Connect to the master database.

```
# use master
# go
```

5 Create a Sybase user account.

```
# sp_addlogin user_name, password
# go
```

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

6 Create a database.

```
# create database database_name
# go
```

The detail monitor script should make transactions on this database.

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

```
# sp_dboption database_name, "trunc log on chkpt", true
# go
```

8 Connect to the database that is created in step 6.

```
# use database_name
# go
```

9 Associate the user created in step 5 with the database created in step 6.

```
# sp_adduser user_name
# go
```

10 Change the user to the one created in step 5.

```
# setuser "user_name"
# go
```

11 Create a table in the database.

```
# create table table_name (lastupd datetime)
# go
```

The detail monitor script should make transactions on this table.

If you 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.

```
# insert into table_name (lastupd) values (getdate())
# go
```

13 Exit the database.

```
# exit
```

Chapter 3

Installing, upgrading, and removing the agent for Sybase

This chapter includes the following topics:

Before you install or upgrade the agent for Sybase

Ensure that you meet the following prerequisites to install the VCS agent for Sybase:

- Verify that VCS is installed and configured. If required, review the *Cluster Server Configuration and Upgrade Guide*.
- Verify that Sybase is installed and configured on all cluster nodes on which you will install the agent. Review the Sybase documentation for more information.
  See "VCS requirements for installing Sybase" on page 18.
- Verify that the sybase account is valid and identical on all cluster systems that will run Sybase. Verify that the sybase user account has permissions to execute Sybase binaries.
  See "System Group and User for Sybase home directory" on page 19.

Installing the agent for Sybase

The agent binaries for Cluster Server agent for Sybase are a part of the VRTSvcsea RPM. The VRTSvcsea RPM is already installed if you chose to install all packages or recommended packages during VCS installation. You must manually install the VRTSvcsea RPM if you installed minimal packages during VCS installation. For Sybase ASE cluster edition, the VCS product installer installs the Sybase agent VRTSvcsea RPM along with other components.

You can also install the Cluster Server agent for Sybase from the product disc.

Install the agent for Sybase on each node in the cluster.

To install the agent on Linux systems

1 Log in as superuser.

2 Make sure that the agent is not already installed.

Navigate to `/opt/VRTSagents/ha/bin` and list the contents of the directory. If the command returns the VCS agents for Sybase, you can skip this procedure.

3 Insert the software disc that contains the Sybase agent software into the system's drive. The disc automatically mounts.

If the disc does not automatically mount, enter:

```
# mount -o ro /dev/cdrom /mnt/cdrom
```

4 Navigate to the folder that contains the agent software.

```
# cd /mnt/cdrom/dist_x86_64/rpms
```

Where

dist is a supported Linux distribution, for example, rhel7

5 Install the agent software.

```
# rpm -i VRTSvcsea-7.4.3.0.000-dist.x86_64.rpm
```

Where

dist is a supported Linux distribution, for example, RHEL7

6 The VRTSvcsea RPM includes the agent binaries for the Sybase agent. The RPM also installs the VCS agents for DB2 and Oracle.

7 Copy the `SybaseTypes.cf` file from `/etc/VRTSagents/ha/conf/Sybase/` to `/etc/VRTSvcs/conf/config`.

Disabling the agent for Sybase

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

To disable the agent

1. Determine if the service group is online. At the prompt, type:

   ```
   # hagrp -state service_group -sys system_name
   ```

2. If the service group is online, switch the service group to another system or take it offline. At the prompt, type:

   ```
   # hagrp -switch service_group -to system_name
   ```

   Or

   ```
   # hagrp -offline service_group -sys system_name
   ```

3. Stop the agent on the system. At the prompt, type:

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

4. 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 that the agent has stopped.

5. When the agent has stopped, you can remove the system, the service group, or the resource type from the VCS configuration.

   For more information, refer to the chapter on reconfiguring VCS from the command line in *Cluster Server Administrator's Guide*.

Removing the agent for Sybase

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

Warning: The VRTSvcsea RPM includes the VCS agents for Oracle, Sybase, and DB2. So, the following procedure to remove the VCS agent for Sybase removes all the other agents also.

To remove the agent for Sybase

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

   ```
   # haagent -stop Sybase -sys system_name
   ```

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

   ```
   # haagent -stop SybaseBk -sys system_name
   ```

   Perform this step on all nodes on which the agent for SybaseBk is running.
4. Type the following command to remove the agent from all nodes in the cluster. Answer prompts accordingly.

   ```
   # rpm -e VRTSvcsea
   ```

Upgrading the agent for Sybase

This section describes the procedure to upgrade the VCS agent for Sybase in a Cluster Server (VCS) cluster. Execute the following steps on each node of the VCS cluster.

Note: For details on the supported upgrade paths for the agent, refer to the Configuration and Upgrade Guide for your product.

To upgrade the VCS agent for Sybase

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

   ```
   # haconf -dump -makero
   # hastop -all -force
   ```

2. Take a back up of the configuration file (main.cf) to a location on the cluster node.

3 Based on your current VCS Agent version for Sybase, perform the appropriate steps on all systems that have the agent installed:

To	Perform the following steps

Upgrade the VCS agent according to the supported upgrade path	1. Navigate to the folder that contains the agent software, by running the following command:
	`# cd /mnt/cdrom/dist_x86_64/rpms`
	Where:
	dist is a supported Linux distribution, for example, rhel7
	2. Upgrade the agent by running the following command:
	`# rpm -U VRTSvcsea-7.4.3.0.0.000-dist.x86_64.rpm`
	Where:
	dist is a supported Linux distribution, for example, RHEL7
	Note: This command also upgrades the VCS agents for Oracle and DB2.

4 Make sure that you have copied the SybaseTypes.cf file from the /etc/VRTSagents/ha/conf/Sybase/ directory to /etc/VRTSvcs/conf/config directory.

Make sure to update the newly copied SybaseTypes.cf file with all the type-level changes that you had made to the older SybaseTypes.cf file.

For example, if you had changed the value of the MonitorInterval attribute from the default 60 to 120 seconds, the SybaseTypes.cf file gets updated. You must apply these type-level changes manually to the newly copied SybaseTypes.cf file.

5 Copy the main.cf from the backed up location to /etc/VRTSvcs/conf/config directory.

6 Verify the configuration.

```
# cd /etc/VRTSvcs/conf/config
# hacf -verify config
```

7 Start VCS on the local node.

8 Start VCS on the other nodes.

Chapter 4

Configuring VCS service groups for Sybase

This chapter includes the following topics:

About configuring service groups 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. Before you configure the agent, review the Sybase resource types and their attributes.

Before configuring the service group for Sybase

Before you configure the Sybase service group, you must meet the following prerequisites:

- Verify that VCS is installed and configured on all nodes in the cluster where you plan to configure the service group. For more information on configuring VCS see the *Cluster Server Configuration and Upgrade Guide*.

  Caution: In Sybase cluster edition, the Sybase agent assumes that the **membership-mode** is set to **vcs**. When Sybase instances are configured under VCS control, the installer automatically sets the **membership-mode** to **vcs**. If the Sybase server instances are configured manually under VCS control or if the quorum device is rebuilt in ‘native’ mode, ensure that the **membership-mode** is set to **vcs**.

- Verify that Sybase is installed and configured identically on all nodes in the cluster.
  See “VCS requirements for installing Sybase” on page 18.
- Verify that the Cluster Server agent for Sybase is installed on all nodes in the cluster.
  See “Installing the agent for Sybase” on page 24.
- Verify the type definition for Cluster Server agent for Sybase is imported into the VCS engine.
  See “Importing the SybaseTypes.cf file” on page 29.

Note: If you use the VCS Cluster Manager (Java Console) to configure attributes, the passwords are automatically encrypted; you need not encrypt them manually.

Caution: If you do not set the Quorum_dev attribute for Sybase Cluster Edition, the Sybase agent does not perform the qrmutil-based checks. This error in configuration may lead to undesirable results. For example, if qrmutil returns failure pending, the agent does not panic the system.

Importing the SybaseTypes.cf file

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

To import the SybaseTypes.cf file using the Cluster Manager (Java Console)

1. On one of the nodes in the cluster, start the Cluster Manager (Java Console).
2. Log in to the cluster and wait for the Cluster Explorer to launch.
3. From the **File** menu select **Import Types**. Switch to the read/write mode if prompted.
4. In the Import Types dialog box, select the file:

   ```
   /etc/VRTSagents/ha/conf/Sybase/SybaseTypes.cf
   ```

5. Click **Import** and wait for the file to import.
6. Save the configuration.

To import the SybaseTypes.cf file using the command line

1. Log in to a cluster system as superuser.
2. Make the cluster configuration as read-only. This action ensures that 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. Leave the resources that are 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 "/etc/VRTSagents/ha/conf/Sybase/SybaseTypes.cf"
   ```

   The Sybase types definition is imported to the VCS engine. The agent for Sybase can be configured without interrupting or stopping VCS.

Using SystemD attributes for Sybase and SybaseBk

InfoScale provides the optional attributes, UseSystemD and SystemDAttrList, to the Sybase and the SybaseBk agents in SystemD environments.

You can customize the Sybase or the SybaseBk application unit service file with the attributes for SystemD in the following ways:

- Adding key-value pairs to the SystemDAttrList attribute
  The SystemDAttrList attribute contains a set of key-value data pairs that map to system-specific attributes and their values. These values are used to set the platform- and application-specific environment.
- Customizing the application unit service file with SystemD attributes
  Veritas recommends that you do not modify or add new SystemD attributes to the existing `vcs-`*`resourceName`*`.service` unit file, because it gets overwritten when the online entry point is invoked.

Adding key-value pairs to the SystemDAttrList attribute

1. Set the UseSystemD attribute to 1.
2. To add a key-value pair in the SystemDAttrList attribute, run the command:

   ```
   # hares -modify resourceType SystemDAttrList key value
   ```

   For example:

   ```
   # hares -modify sybase SystemDAttrList LimitNOFILE 2048
   ```

3. (Optional) To modify an existing attribute, run the command:

   ```
   # hares -modify resourceType SystemDAttrList -update key value
   ```

   For example:

   ```
   # hares -modify sybase SystemDAttrList -update LimitNOFILE 1024
   ```

   After you update the SystemDAttrList attribute, the `vcs-custom.conf` file reflects the added key-value pairs.

   A sample set of SystemD attribute-value pairs:

   ```
   Systemd attribute = Value
   LimitLOCKS = infinity
   LimitMEMLOCK = infinity
   LimitCORE = infinity
   LimitNOFILE = 1024
   ```

Customizing the application unit service file with SystemD attributes

1. To customize additional SystemD attributes in the `vcs-`*`resourceName`*`.service` file, create the `foo.conf` unit configuration file under `/etc/systemd/system/vcs-sybase_res.service.d` with new attributes.

   When the online entry point is invoked, SystemD applies the new attributes from `foo.conf` to `vcs-`*`resourceName`*`.service`, and then uses the unit service file for the application.

2. After you add the required system attributes in `foo.conf`, load the SystemD daemon.

   ```
   # systemctl --system daemon-reload
   ```

3. Verify the status of the application service.

   ```
   # systemctl status -l vcs-resourceName.service
   ```

Configuring the service groups for Sybase

You can configure the service groups for Sybase 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 for Sybase from Cluster Manager (Java console)" on page 32.
- 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 for Sybase using the command line" on page 34.
- By using the Veritas Operations Manager (VOM).
  For more information, see VOM documentation.

Configuring the service group for Sybase from Cluster Manager (Java console)

A template for the Sybase resource groups is automatically installed with the VCS 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 that the Sybase type definition file SybaseTypes.cf is imported in your configuration.
2. Launch the Cluster Configuration wizard using any of the following 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 is configured.
   - Enter the name of the service group.
   - From the **Available Systems** box, select the systems on which to configure the service group.
   - 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.
   - Specify system priority for the service group to failover. System priority is numbered sequentially, with the lowest assigned number denoting the highest priority.
   - 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 to notify 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 that is based on the selected template.

   A progress indicator displays the percentage of the commands that are 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 of the resources.

   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. You can modify the attribute values as given in the procedure that follows.

9. Click **Finish** to accept the default values and complete the configuration.

To edit the Sybase service group 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.
6. Follow the wizard instructions to complete the configuration. Click **Finish** to quit the wizard.

   Caution: For added security, you must always provide a secure value for passwords. The password may contain letters (a-z), numbers (0-9), or special characters.

   See “Encrypting passwords for Sybase” on page 36.

Configuring the service group for Sybase using the command line

The VCS agent for Sybase contains 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 for Sybase from the command line

1. Log in to a cluster system as superuser.
2. Make sure the Sybase type definition is imported into VCS engine.

   See “Configuring the service groups for Sybase” on page 32.

3 Edit the main.cf file at `/etc/VRTSvcs/conf/config/main.cf`. For reference, use the sample files at `/etc/VRTSagents/ha/conf/Sybase`.

- Create a Sybase service group.
- Create the Sybase and SybaseBk resources.
  See "Sybase resource type" on page 44.
  See "SybaseBk resource type" on page 53.
- Edit the default attributes to match the parameters in your configuration.
  For added security, you must always provide a secure value for passwords.
  See "Encrypting passwords for Sybase" on page 36.
- Assign dependencies to the newly created resources. Refer to the sample file at `/etc/VRTSagents/ha/conf/Sybase/`.
  See the *Cluster Server Administrator'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`

```
# cd /etc/VRTSvcs/conf/config
# hacf -verify .
```

6 Start VCS on the local node.

```
# hastart
```

7 Start VCS on the other nodes.

8 If the system is listed in AutoStartList attribute of the Sybase service group, verify that all Sybase service group resources are brought online.

```
# hagrp -state
```

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

```
# hagrp -offline service_group -sys system_name

# hagrp -state
```

10 Bring the service group online again and verify that all the resources are available.

```
# hagrp -online service_group -sys system_name

# hagrp -state
```

11 Switch the Sybase service group to the other node.

```
# hagrp -switch service_group -to system_name
```

12 Verify that all Sybase service group resources are brought online on the other node.

```
# hagrp -state
```

13 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 for Sybase

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

The `vcsencrypt` utility also allows you to encrypt the agent passwords using a security key. The security key supports AES-256 encryption, which creates a secure password for the agent. For details, refer to the *Cluster Server Administrator's Guide*.

About setting up detail monitoring for the agent for Sybase

The VCS agent for Sybase provides two levels of application monitoring: basic and detail. In basic monitoring, Sybase resource monitors the Sybase dataserver 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 functions properly. The agent uses this test table for internal purposes. Veritas recommends that you do not perform any other transaction on the test table. The agent uses the script that is 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 LevelTwoMonitorFreq 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 must have execute permissions for the root.

Enabling detail monitoring for the agent for Sybase

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

Note: All Sybase resources in the cluster gets modified when you enable detail monitoring for agents.

To enable detail monitoring

1. Make sure the Sybase server is configured for detail monitoring.

   See "Configuring Sybase for detail monitoring" on page 20.

2. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

3. Enable detail monitoring for Sybase.

   ```
   # hatype -modify Sybase LevelTwoMonitorFreq <value>
   # hares -modify Sybase_resource User user_name
   # 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"
   ```

   Note: To enable detail monitoring, the LevelTwoMonitorFreq attribute must be set to a positive value. You can also override the value of this attribute at the resource level.

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 for the agent for Sybase

1 Make the VCS configuration writable with:

```
# haconf -makerw
```

2 To disable detail monitoring for Sybase run the following command:

```
# hatype -modify Sybase LevelTwoMonitorFreq 0
```

Note: If the LevelTwoMonitorFreq attribute is overridden at resource level, then use `hares` command and set the LevelTwoMonitorFreq attribute to 0 at resource level.

3 Save the configuration with:

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

Enabling and disabling intelligent resource monitoring (IMF) for agents

Review the following procedures to enable or disable intelligent resource monitoring manually. The intelligent resource monitoring feature is enabled by default. The IMF resource type attribute determines whether an IMF-aware agent must perform intelligent resource monitoring.

Note: VCS provides `haimfconfig` script to enable and disable IMF for agents. You can use the script when VCS is running or when VCS is stopped. Use the script to enable or disable IMF for the IMF-aware bundled agents, enterprise agents, and custom agents.

For additional information, refer to *Enabling and disabling IMF for agents by using script* section in *Cluster Server Administrator's Guide*.

See “How the Sybase and SybaseBk agents support intelligent resource monitoring ” on page 9.

Note: All Sybase resources in the cluster gets modified when you enable or disable IMF monitoring for agents.

To enable intelligent resource monitoring

1. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

2. Run the following command to enable intelligent resource monitoring.

   - To enable intelligent monitoring of offline resources:

     ```
     # hatype -modify resource_type IMF -update Mode 1
     ```

   - To enable intelligent monitoring of online resources:

     ```
     # hatype -modify resource_type IMF -update Mode 2
     ```

   - To enable intelligent monitoring of both online and offline resources:

     ```
     # hatype -modify resource_type IMF -update Mode 3
     ```

   You can also change the values of the MonitorFreq key and the RegisterRetryLimit key of the IMF attribute.

   See “Attribute definitions for the Sybase agent” on page 45.

   See “Attribute definitions for the SybaseBk agent” on page 54.

3. Save the VCS configuration

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

4. Make sure that the AMF kernel driver is configured on all nodes in the cluster.

   ```
   /opt/VRTSamf/bin/amf status
   ```

5. Restart the agent. Run the following commands on each node.

   ```
   # haagent -stop agent_name -force -sys sys_name
   # haagent -start agent_name -sys sys_name
   ```

To disable intelligent resource monitoring

1. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

2. To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

   ```
   # hatype -modify Sybase IMF -update Mode 0
   ```

3 To disable intelligent resource monitoring for a specific resource, run the following command:

```
# hares -override resource_name IMF
# hares -modify resource_name IMF -update Mode 0
```

4 Save the VCS configuration.

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

Note: For Sybase cluster edition, the agent performs a monitoring check based on `qrmutil` whenever the monitor entry point is executed, either because IMF MonitorFreq is reached or LevelTwoMonitorFreq is reached. For example, if IMF MonitorFreq is set to a higher value such as 10, monitoring check based on `qrmutil` is performed after 10 x MonitorInterval seconds.

Administering the AMF kernel driver

Perform the following procedures to start or stop the IMF kernel driver.

To start the AMF kernel driver

1 Set the value of the AMF_START variable to 1 in the following file:

```
# /etc/sysconfig/amf
```

2 Start the AMF kernel driver. Run the following command:

```
# systemctl start amf
```

To stop the AMF kernel driver

1 Set the value of the AMF_STOP variable to 1 in the following file:

```
# /etc/sysconfig/amf
```

2 Run the following command:

```
# systemctl stop amf
```

Chapter 5

Administering VCS service groups for Sybase

This chapter includes the following topics:

About administering service groups for Sybase

You can administer service groups with the Cluster Manager (Java Console), the Veritas InfoScale Operations Manager, or from the command line. The following procedures use the Cluster Manager (Java Console).

Bringing the Sybase service group online

Perform the following steps to bring a service group online. Note that in the initial few cycles of bringing a service group online, the memory usage by the agent can spike.

To bring a service group online

1. From Cluster Explorer, click the **Service Groups** tab in the configuration tree.
2. Right-click the service group and click **Enable Resources** to enable all the resources in this group.
3. Right-click the service group, hover over **Enable**, and select either the node or all the nodes where you want to enable the service group.
4. Save and close the configuration. Click **File > Save Configuration**, then **Close Configuration**.
5. Right-click the service group, pause over Online, and select the system where you want to bring the service group online.

Taking the Sybase service group offline

Perform the following procedure from Cluster Manager (Java Console) to take the service group offline.

To take a service group offline

1. In the Cluster Explorer configuration tree with the Service Groups tab selected, right-click the service group that you want to take offline.
2. Choose **Offline**, and select the appropriate system from the pop-up menu.

Switching the Sybase service group

The process of switching a service group involves taking it offline on its current system and bringing it online on another system. Perform the following procedure from Cluster Manager (Java Console) to switch the service group.

To switch a service group

1. In the Cluster Explorer configuration tree with the Service Groups tab selected, right-click the service group.
2. Choose **Switch To**, and select the appropriate system from the pop-up menu.

Modifying the Sybase service group configuration

You can dynamically modify the Sybase agent using several methods, including the Cluster Manager (Java Console), Veritas InfoScale Operations Manager, and the command line.

Viewing the agent log for Sybase

The VCS 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

Resource type definitions for Sybase

This appendix includes the following topics:

About the resource type and attribute definitions for Sybase

The resource type represents the VCS configuration definition of the agent. It also specifies how the the configuration file, main.cf, defines the agent. Configuring the agent involves assigning values to the resource type attributes.

For the resource type attributes:

See "Attribute definitions for the Sybase agent" on page 45.

See "Attribute definitions for the SybaseBk agent" on page 54.

For the sample main.cf configuration files:

See "Sample configuration for a Sybase service group on Linux systems" on page 60.

Sybase resource type

The type definitions and attribute definitions for the Sybase resource type are described as follows.

Type definition for the Sybase agent

The resource type definition for the agent for Sybase is as follows.

```
type Sybase (
    static boolean AEPTimeout = 1
    static keylist SupportedActions = { "checkpoint_all" }
    str Server
    str Owner
    str Home
    str Version
    str SA
    str SApswd
    str Run_ServerFile
    str User
    str UPword
    str Db
    str Table
    str Monscript = "/opt/VRTSagents/ha/bin/Sybase/SqlTest.pl"
    boolean WaitForRecovery = 0
    str Quorum_dev
    str interfaces_File
    int ShutdownWaitLimit = 60
    int DelayAfterOnline = 10
    int DelayAfterOffline = 2
    static int ToleranceLimit = 1
    str SystemDAttrList{}
    boolean UseSystemD = 0
    static str ArgList[] = { Server, Owner, Home, Version, SA, SApswd,
      User, UPword, Db, Table, Monscript, WaitForRecovery,
      Run_ServerFile, Quorum_dev, State, interfaces_File,
      ShutdownWaitLimit, DelayAfterOnline, DelayAfterOffline,
      UseSystemD , SystemDAttrList }
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
    static str IMFRegList[] = { Server, Owner, Quorum_dev }
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Sybase"
)
```

Attribute definitions for the Sybase agent

Review the description of the Sybase agent attributes. The agent attributes are classified as required, optional, and internal.

Table A-1 lists the required attributes.

Table A-1 Required attributes

Required Attributes	Definition
Home	The $SYBASE path to Sybase binaries and configuration files. Type and dimension: string-scalar Default value: No default value
Owner	Sybase user as the defined owner of executables and database files in any of the sources (such as NIS+, /etc/hosts, and so on) specified in the /etc/nsswitch.conf file for passwd entry. The Sybase executables and database files are accessed in the context of this user. Type and dimension: string-scalar See "System Group and User for Sybase home directory" on page 19.
Quorum_dev	The quorum device manages the cluster membership, stores cluster configuration data, and contains information shared among server instances and nodes. The quorum device is a disk that is accessible to all the nodes in the cluster. Specify a fully qualified quorum device name. Type and dimension: string-scalar Default value: No default value Note: This attribute should be specified only for the cluster edition. **Note:** If you do not set the Quorum_dev attribute for Sybase Cluster Edition, the Sybase agent does not perform the qrmutil-based checks. This error in configuration may lead to undesirable results. For example, if qrmutil returns failure pending, the agent does not panic the system.
SA	Sybase database administrator. This attribute is required to connect to the ASE for shutdown. Type and dimension: string-scalar Default value: No default value

Table A-1 Required attributes *(continued)*

Required Attributes	Definition
SApswd	Encrypted password for Sybase database administrator. This password is required to connect to the ASE for shutdown. Type and dimension: string-scalar Default value: No default value See "Encrypting passwords for Sybase" on page 36. **Note:** You need not specify a value for this attribute if the SA user does not require a password.
Server	The $DSQUERY ASE name. Only one server should be configured in a Sybase service group. The advantage of configuring Sybase resources in a separate service group is, each Sybase data server can failover independently. When multiple Sybase resources are configured in a single service group, even if one of the "n" data servers fail, the entire service group will be failed over. Type and dimension: string-scalar Default value: No default value
Version	Version of Sybase ASE. Type and dimension: string-scalar Default value: No default value **Note:** After the Sybase resource is online in VCS, you must not modify the Home and Version attributes. For the Sybase cluster edition, setting invalid values for Home and Version attributes when the resource is in Online state causes the node to panic. **Note:** For Sybase cluster edition, after the Sybase resource is online in VCS, you must not modify the Home and Version attributes. For the Sybase cluster edition, setting invalid values for Home and Version attributes when the resource is in Online state causes the node to panic.

Table A-2 lists the optional attributes.

Table A-2 Optional attributes

Optional Attributes	Definition
DetailMonitor	Specifies whether the Sybase server is monitored in detail. A positive integer value indicates that the resource monitors the Sybase server in detail. Value 0 denotes it does not. **Note:** This attribute has been removed from the agent version 6.1.0 onwwards. Default is 0. Type and dimension: int-scalar **Note:** The DetailMonitor attribute is deprecated in VCS 7.4.3. Instead, LevelTwoMonitorFreq attribute at the Sybase resource type level may be used.
LevelTwoMonitorFreq	Specifies the frequency at which the agent for this resource type must perform second-level or detailed monitoring. This is a resource type level attribute. You can override the value of this attribute at the resource level. The default value of LevelTwoMonitorFreq attribute is 0 (zero). If required, the value of LevelTwoMonitorFreq attribute can be overridden at resource level.
User	The database user, in the context of which, the transactions are performed on the database. You must specify a value for this attribute if LevelTwoMonitorFreq is set to a positive integer value. Type and dimension: string-scalar Default value: No default value
UPword	Encrypted password for the database user. You must specify a value for this attribute if LevelTwoMonitorFreq is set to a positive integer value. However, you need not specify a value for this attribute if the database user does not require a password. See "Encrypting passwords for Sybase" on page 36. intercType and dimension: string-scalar Default value: No default value

Table A-2 Optional attributes *(continued)*

Optional Attributes	Definition
Db	Name of the database used for detailed monitoring. The table used by the detail monitor script resides in this database. You must specify a value for this attribute if LevelTwoMonitorFreq is set to a positive integer value. Type and dimension: string-scalar Default value: No default value
Table	Name of the table on which the detail monitoring script performs the transactions. You must specify a value for this attribute if LevelTwoMonitorFreq is set to a positive integer value. Type and dimension: string-scalar Default value: No default value
Monscript	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. You must specify a value for this attribute if LevelTwoMonitorFreq is set to a positive integer value. Type and dimension: string-scalar Default value: No default value **Note:** By default, SqlTest.pl script has the execute permission set. If you specify custom detail monitor script, ensure that custom detail monitor script also has the execute permissions set.
WaitForRecovery	The recovery procedures rebuild the server's databases from the transaction logs. This boolean attribute indicates if the agent should wait during Online agent function, till recovery has been completed and all databases that can be made online are brought online. This feature is not supported for pre-12.5.1 Sybase ASE server. Type and dimension: boolean-scalar Default value: 0
Run_ServerFile	Specifies the location of the RUN_SERVER file for the Sybase instance. The default location of this file is used if no value is specified for this attribute. Type and dimension: string-scalar Default value: No default value

Table A-2 Optional attributes *(continued)*

Optional Attributes	Definition
IMF	This resource-type level attribute determines whether the Sybase agent must perform intelligent resource monitoring. You can also override the value of this attribute at resource-level. This attribute includes the following keys: ■ Mode: Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources Default: 3

Table A-2 Optional attributes *(continued)*

Optional Attributes	Definition
IMF (cont.)	■ MonitorFreq: This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. Default: 5 You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring. If the value is 0, the agent does not perform poll-based process check monitoring. After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows: ■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources ■ RegisterRetryLimit: If you enable intelligent resource monitoring, the agent invokes the sybase_imf_register agent function to register the resource with the AMFkernel driver. The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes. Default: 3 See “Enabling and disabling intelligent resource monitoring (IMF) for agents” on page 38. Type and dimension: Integer-association.
interfaces_File	Specifies the location of interfaces file, including the directory name and the file name for the Sybase instance. If this attribute is configured, [-I interfaces file] option is used when connecting to the isql session. If this attribute is not configured, the agent does not use the -I option. Type and dimension: string-scalar Default value: No default value For example: /sybase/my_interfaces_file **Note:** It is assumed that you have modified the RUN_ServerFile with the non-default interface file location if the interfaces_File attribute is configured.

Table A-2 Optional attributes *(continued)*

Optional Attributes	Definition
DelayAfterOnline	Specifies the number of seconds that elapse after the Online entry point is complete and before the next monitor cycle is invoked. Type and dimension: integer-scalar Default value: 10
DelayAfterOffline	Specifies the number of seconds that elapse after the Offline entry point is complete and before the next monitor cycle is invoked. Type and dimension: integer-scalar Default value: 2
ShutdownWaitLimit	Maximum number of seconds for which the agent waits for the Sybase instance to stop after issuing the `shutdown with wait` command, and before attempting to issue the `kill -15 <data server-pid>` command, if required. Type and dimension: integer-scalar Default value: 60
Quorum_dev	The quorum device manages the cluster membership, stores cluster configuration data and contains information shared among server instances and nodes. It must be a disk accessible to all nodes in the cluster. Specify fully qualified quorum device name. **Note:** If you do not set the Quorum_dev attribute for Sybase Cluster Edition, the Sybase agent does not perform the qrmutil-based checks. This error in configuration may lead to undesirable results. For example, if qrmutil returns failure pending, the agent does not panic the system. Type and dimension: String-scalar Default value: No default value
Run_ServerFile	Specifies the location of the RUN_SERVER file of the Sybase instance. The default location of the file is used if no value is specified for this attribute. Type and dimension: String-Scalar Default value: No default value

Table A-2 Optional attributes *(continued)*

Optional Attributes	Definition
UseSystemD	SystemD is a system and service manager for Linux operating systems. It helps manage applications across Linux distributions that support SystemD feature. ■ When this attribute is set to 0, the online entry point starts the resource in user.slice. ■ When this attribute is set to 1, the online entry point brings the Sybase or the SybaseBk resource online as a service in system.slice. The application process can be viewed as `vcs-`*`resourceName`*`.service`, in `/etc/systemd/system`. **Note:** After a system restart, if an application resource is started under user.slice, there is a possibility that the processes are not stopped gracefully. In this case, you can enable the UseSystemD attribute. Type and dimension: Boolean-Scalar
SystemDAttrList	Set the key-value pairs for the SystemD environment using this attribute. **Note:** Ensure that UseSystemD is set to 1 before you set the key-value pairs. To know more about the different limits that you can apply to an application, refer to the appropriate Red Hat documentation. Type and dimension: String-scalar

SybaseBk resource type

Type definition for the SybaseBk agent

The resource type definition for the SybaseBk agent is as follows:

```
type SybaseBk (
    static boolean AEPTimeout = 1
    str Backupserver
    str Owner
    str Home
    str Version
    str Server
    str SA
    str SApswd
    str Run_ServerFile
    str interfaces_File
    str SystemDAttrList{}
    boolean UseSystemD = 0
    static str ArgList[] = { Backupserver, Owner, Home, Version,
      Server, SA, SApswd, Run_ServerFile, interfaces_Filei,
      UseSystemD , SystemDAttrList }
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
    static str IMFRegList[] = { Backupserver, Owner }
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/SybaseBk"
)
```

Attribute definitions for the SybaseBk agent

Review the description of the SybaseBk agent attributes. The agent attributes are classified as required and internal.

Table A-3 lists the required attributes for SybaseBk resource.

Table A-3 Required attributes

Required Attributes	Definition
Server	The $DSQUERY Backup name. Only one server should be configured in a Sybase service group. The advantage of configuring Sybase resources in a separate service group is, each Sybase data server can failover independently. When multiple Sybase resources are configured in a single service group, even if one of the “n” data servers fail, the entire service group will be failed over. Type and dimension: string-scalar
Owner	Sybase user as the defined owner of executables and database files in any of the sources (such as NIS+, /etc/hosts, and so on) specified in the /etc/nsswitch.conf file for passwd entry. The Sybase executables and database files are accessed in the context of this user. Type and dimension: string-scalar
Home	The $SYBASE path to Sybase binaries and configuration files. Type and dimension: string-scalar
Version	Version of Sybase Backup Server. Type and dimension: string-scalar

Table A-3 Required attributes *(continued)*

Required Attributes	Definition
Backupserver	The $BACKUP SYBASE Backup Server name. Type and dimension: string-scalar
SA	Sybase database administrator. This attribute is required to connect to the ASE for shutdown. Type and dimension: string-scalar
SApswd	Encrypted password of Sybase database administrator. This password is required to connect to the ASE for shutdown. Type and dimension: string-scalar See "Encrypting passwords for Sybase" on page 36. **Note:** You need not specify a value for this attribute if the SA user does not require a password.

Table A-4 lists the optional attribute for SybaseBk agent.

Table A-4 Optional attribute

Optional attribute	Definition
Run_ServerFile	Specifies the location of the RUN_SERVER file for the Sybase instance. The default location of this file is used if no value is specified for this attribute. Type and dimension: String-scalar Default value: No default value
interfaces _File	Specifies the location of interfaces file for the Sybase instance. If this attribute is configured, [-I interfaces file] option is used when connecting to the isql session. If this attribute is not configured, the agent does not use the -I option. Type and dimension: string-scalar Default value: No defalult value For example: /sybase/my_interfaces_file **Note:** It is assumed that you have modified the RUN_ServerFile with the non-default interface file location if the interfaces_File attribute is configured.

Table A-4 Optional attribute *(continued)*

Optional attribute	Definition
IMF	

Table A-4 Optional attribute *(continued)*

Optional attribute	Definition
	This resource-type level attribute determines whether the Sybase agent must perform intelligent resource monitoring. You can also override the value of this attribute at resource-level. This attribute includes the following keys: ■ Mode: Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources Default: 3 ■ MonitorFreq: This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. Default: 5 You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring. If the value is 0, the agent does not perform poll-based process check monitoring. After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows: ■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources ■ RegisterRetryLimit: If you enable intelligent resource monitoring, the agent invokes the oracle_imf_register agent function to register the resource with theAMFkernel driver. The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes. Default: 3

Table A-4 Optional attribute *(continued)*

Optional attribute	Definition
	Type and dimension: Integer-association.
UseSystemD	SystemD is a system and service manager for Linux operating systems. It helps manage applications across Linux distributions that support SystemD feature. ■ When this attribute is set to 0, the online entry point starts the resource in user.slice. ■ When this attribute is set to 1, the online entry point brings the Sybase or the SybaseBk resource online as a service in system.slice. The application process can be viewed as `vcs-`*`resourceName`*`.service`, in `/etc/systemd/system`. **Note:** After a system restart, if an application resource is started under user.slice, there is a possibility that the processes are not stopped gracefully. In this case, you can enable the UseSystemD attribute. Type and dimension: Boolean-Scalar
SystemDAttrList	Set the key-value pairs for the SystemD environment using this attribute. **Note:** Ensure that UseSystemD is set to 1 before you set the key-value pairs. To know more about the different limits that you can apply to an application, refer to the appropriate Red Hat documentation. Type and dimension: String-scalar

Table A-5 lists the internal attribute for SybaseBk agent.

This attribute is for internal use only. Veritas recommends not to modify the value of this attribute.

Table A-5 Internal attribute

Internal attribute	Definition
AgentDirectory	Specifies the location of the binaries, scripts, and other files related to the agent for SybaseBk. Type and dimension: static-string

Appendix B

Sample configurations for Sybase

This appendix includes the following topics:

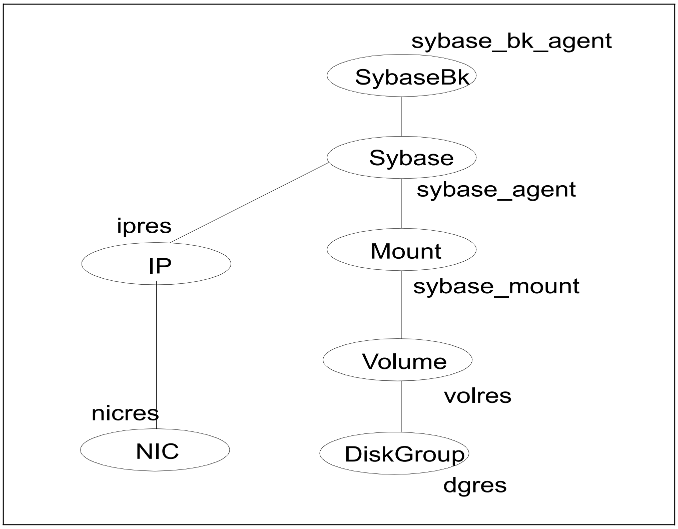
About the sample configuration for the Sybase agent

After Sybase agent installation, several sample `main.cf` file types can be viewed in the `/etc/VRTSagents/ha/conf/Sybase`. This appendix describes a typical service group that is 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 you configure the agent. For more information about these resource types, see the *Custer Server Bundled Agents Reference Guide*.

Resource dependency graph for the Sybase agent

Figure B-1 illustrates the configuration's dependency graph for Linux.

Figure B-1 Dependency graph



The 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, LVMLogicalVolume, and DiskReservation resource types to configure shared storage instead of DiskGroup and Volume resource types.

Sample configuration for a Sybase service group on Linux systems

The following sample depicts a basic configuration with a Sybase service group configured with one NIC, one database instance configured with detail monitoring, and one backup instance.

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

```
cluster vcs_cluster (
        UserNames = { admin = anoGniNkoJooMwoInl }
        CounterInterval = 5
        )

system system1 (
        )

system system2 (
        )

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

IP ipres (
   Device = eth0
   Address = "16.9.1.9"
   NetMask = "255.255.0.0"
   )

Mount sybase_mount (
   MountPoint = "/sybase"
   BlockDevice = "/dev/vx/dsk/vxdgG01/vxvolG01"
   FSType = vxfs
   MountOpt = rw
   FsckOpt = "-y"
   )

Volume sybase_volres (
   Volume="vxvolG01"
   DiskGroup="vxdgG01"
   )

DiskGroup sybase_dres (
   DiskGroup = "vxdgG01"
   StartVolumes=0
   StopVolumes=0
   )

NIC nicres (
   Device = eth0
   NetworkHosts = { "16.9.10.1", "16.9.10.2" }
   )

Sybase sybase_agent (
   Server = SYB15LNX32
   Owner = sybase
   Home = "/home/sybase"
   Version = "15.0.0"
   SA = sa
   LevelTwoMonitorFreq = 1
   User = vcsuser
   UPword = GSNsLSoSLsPSt
   Db = vcsdb
   Table = vcstable
   Monscript = "/opt/VRTSagents/ha/bin/Sybase/SqlTest.pl"
   WaitForRecovery = 1
   Run_ServerFile = "/home/sybase/RUN_SYB15LNX32"
   interfaces_File = "/newfs/interfaces"
   )

SybaseBk sybase_bk_agent (
   Server = SYB15LNX32
   Owner = sybase
   Home = "/home/sybase"
   Version = "15.0.0"
   Backupserver = SYB15LNX32_BS
   SA = sa
   Run_ServerFile = "/home/sybase/RUN_SYB15LNX32_BS"
   interfaces_File = "/newfs/interfaces"
   )


ipres requires nicres
sybase_agent requires ipres
sybase_agent requires sybase_mount
sybase_mount requires sybase_volres
sybase_volres requires sybase_dres
sybase_bk_agent requires sybase_agent
```

Sample main.cf files for Sybase ASE CE configurations

Sample main.cf file examples are provided for the following Sybase ASE CE configurations:

- Basic cluster configuration
  - With shared mount point on CFS for Sybase binary installation
  - With local mount point on VxFS for Sybase binary installation
- Replicating data between two clusters
  - For a primary site in a CVM VVR configuration
  - For a secondary site in a CVM VVR configuration

Sample main.cf for a basic Sybase ASE CE cluster configuration under VCS control with shared mount point on CFS for Sybase binary installation

This sample main.cf is for a single site with a basic cluster configuration with shared mount point on CFS for Sybase binary installation.

The following are the configuration details for this Sybase ASE CE configuration sample main.cf:

- File name: sybasece_main.cf
- File location: /etc/VRTSagents/ha/conf/Sybase/

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "SybaseTypes.cf"

cluster cluster101 (
    UserNames = { admin = HopHojOlpKppNxpJom }
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    UseFence=SCSI3
    )

system system1 (
    )
```

```
system system2 (
    )

// binmounts group for configuring CFS mounts for Sybase binaries.

group binmnt (
    SystemList = { system1 = 0, system2 = 1 }
    Parallel = 1
    AutoStartList = { system1, system2 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    CFSMount sybbindg_101_sybbinvol_mnt (
         MountPoint = "/sybase"
         BlockDevice = "/dev/vx/dsk/sybbindg_101/sybbin_vol"
         )

    CVMVolDg sybbindg_101_voldg (
         CVMDiskGroup = sybbindg_101
         CVMVolume = { sybbin_vol }
         CVMActivation = sw
         )

 requires group cvm online local firm
 sybbindg_101_sybbinvol_mnt requires sybbindg_101_voldg


 // resource dependency tree
 //
 // group binmnt
 // {
 // CFSMount sybbindg_101_sybbinvol_mnt
 //     {
 //     CVMVolDg sybbindg_101_voldg
 //     }
 // }

// cvm group for CVM and CFS specific agents.

group cvm (
    SystemList = { system1 = 0, system2 = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { system1, system2 }
    )

    CFSfsckd vxfsckd (
        )

    CVMCluster cvm_clus (
        CVMClustName = sfsyb_90
        CVMNodeId = { system1 = 0, system2 = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )

    CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

 cvm_clus requires cvm_vxconfigd
 vxfsckd requires cvm_clus


 // resource dependency tree
 //
 // group cvm
 // {
 // CFSfsckd vxfsckd
 //     {
 //     CVMCluster cvm_clus
 //         {
 //         CVMVxconfigd cvm_vxconfigd
 //         }
 //     }
 // }

// sybasece group for:
// 1. CVM volumes for Sybase database and quorum device
// 2. CFS mount for Sybase database and quorum device
// 3. Process agent for vxfend process.
// 4. Sybase database instance.

group sybasece (
    SystemList = { system1 = 0, system2 = 1 }
    Parallel = 1
    AutoStartList = { system1, system2 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    CFSMount quorum_101_quorumvol_mnt (
        MountPoint = "/quorum"
        BlockDevice = "/dev/vx/dsk/quorum_101/quorumvol"
        )

    CFSMount sybdata_101_sybvol_mnt (
        MountPoint = "/sybdata"
        BlockDevice = "/dev/vx/dsk/sybdata_101/sybvol"
        )

    CVMVolDg quorum_101_voldg (
        CVMDiskGroup = quorum_101
        CVMVolume = { quorumvol }
        CVMActivation = sw
        )

    CVMVolDg sybdata_101_voldg (
        CVMDiskGroup = sybdata_101
        CVMVolume = { sybvol }
        CVMActivation = sw
        )

   Process vxfend (
        PathName = "/sbin/vxfend"
        Arguments = "-m sybase -k /var/VRTSvcs/log/tmp/vcmp_socket"
        )

   Sybase ase (
        Server @system1 = ase1
        Server @system2 = ase2
        Owner = sybase
        Home = "/sybase"
        Version = 15
        SA = sa
        Quorum_dev = "/quorum/q.dat"
        )
```

```
requires group binmnt online local firm
ase requires quorum_101_quorumvol_mnt
ase requires sybdata_101_sybvol_mnt
ase requires vxfend
quorum_101_quorumvol_mnt requires quorum_101_voldg
sybdata_101_sybvol_mnt requires sybdata_101_voldg


// resource dependency tree
//
// group sybasece
// {
// Sybase ase
//     {
//     CFSMount quorum_101_quorumvol_mnt
//         {
//         CVMVolDg quorum_101_voldg
//         }
//     CFSMount sybdata_101_sybvol_mnt
//         {
//         CVMVolDg sybdata_101_voldg
//         }
//     Process vxfend
//     }
// }
```

Sample main.cf for a basic Sybase ASE CE cluster configuration with local mount point on VxFS for Sybase binary installation

This sample main.cf is for a single site with a basic cluster configuration with local mount point on VxFS for Sybase binary installation.

The following are the configuration details for this Sybase ASE CE configuration sample main.cf:

- File name: sybasece_main.cf
- File location: /etc/VRTSagents/ha/conf/Sybase/

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "SybaseTypes.cf"

cluster cluster101 (
    UserNames = { admin = HopHojOlpKppNxpJom }
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    UseFence=SCSI3
    )

system system1 (
    )

system system2 (
    )

// binmounts group for configuring VxFS mounts for Sybase binaries.

group binlocalmnt (
    SystemList = { system1 = 0, system2 = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { system1, system2 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    DiskGroup sybbindg_101_voldg (
         DiskGroup = sybbindg
         )

    Mount sybbindg_101_sybbinvol_mnt (
         MountPoint = "/sybase"
         BlockDevice = "/dev/vx/dsk/sybbindg_101/sybbin_vol"
         FSType = vxfs
         FsckOpt = "-y"
         )

    Volume sybbindg_101_vol (
         DiskGroup = sybbindg
         Volume = sybbinvol
         )

 requires group cvm online local firm
 sybbindg_101_sybbinvol_mnt requires sybbindg_101_vol
 sybbindg_101_vol requires sybbindg_101_voldgdg


 // resource dependency tree
 //
 // group binlocalmnt
 // {
 // Mount sybbindg_101_sybbinvol_mnt
 //     {
 //     Volume sybbindg_vol
 //         {
 //         DiskGroup sybbindg_101_voldg
 //         }
 //     }
 // }

// cvm group for CVM and CFS specific agents.

group cvm (
    SystemList = { system1 = 0, system2 = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { system1, system2 }
    )

    CFSfsckd vxfsckd (
        )

    CVMCluster cvm_clus (
        CVMClustName = cluster101
        CVMNodeId = { system1 = 0, system2 = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )

    CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

 cvm_clus requires cvm_vxconfigd
 vxfsckd requires cvm_clus
 // resource dependency tree
 //
 // group cvm
 // {
 // CFSfsckd vxfsckd
 //     {
 //     CVMCluster cvm_clus
 //         {
 //         CVMVxconfigd cvm_vxconfigd
 //         }
 //     }
 // }

// sybasece group for:
// 1. CVM volumes for Sybase database and quorum device
// 2. CFS mount for Sybase database and quorum device
// 3. Process agent for vxfend process.
// 4. Sybase database instance.

group sybasece (
    SystemList = { system1 = 0, system2 = 1 }
    Parallel = 1
    AutoStartList = { system1, system2 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    CFSMount quorum_101_quorumvol_mnt (
        MountPoint = "/quorum"
        BlockDevice = "/dev/vx/dsk/quorum_101/quorumvol"
        )

    CFSMount sybdata_101_sybvol_mnt (
        MountPoint = "/sybdata"
        BlockDevice = "/dev/vx/dsk/sybdata_101/sybvol"
        )

    CVMVolDg quorum_101_voldg (
        CVMDiskGroup = quorum_101
        CVMVolume = { quorumvol }
        CVMActivation = sw
        )

   CVMVolDg sybdata_101_voldg (
        CVMDiskGroup = sybdata_101
        CVMVolume = { sybvol }
        CVMActivation = sw
        )

  Process vxfend (
        PathName = "/sbin/vxfend"
        Arguments = "-m sybase -k /var/VRTSvcs/log/tmp/vcmp_socket"
        )

  Sybase ase (
        Server @system1 = ase1
        Server @system2 = ase2
        Owner = sybase
        Home = "/sybase"
        Version = 15
        SA = sa
        Quorum_dev = "/quorum/q.dat"
        )

requires group binlocalmnt online local firm
ase requires quorum_101_quorumvol_mnt
ase requires sybdata_101_sybvol_mnt
ase requires vxfend
quorum_101_quorumvol_mnt requires quorum_101_voldg
sybdata_101_sybvol_mnt requires sybdata_101_voldg


// resource dependency tree
//
// group sybasece
// {
// Sybase ase
//     {
//     CFSMount quorum_101_quorumvol_mnt
//         {
//         CVMVolDg quorum_101_voldg
//         }
//     CFSMount sybdata_101_sybvol_mnt
//         {
//         CVMVolDg sybdata_101_voldg
//         }
//     Process vxfend
//     }
// }
```

Sample main.cf for a primary CVM VVR site

This sample main.cf is for a primary site in a CVM VVR configuration. It is one of two sample main.cfs for replicating data between two clusters.

The following are the configuration details for this Sybase ASE CE configuration sample main.cf:

- File name: sybasece_cvmvvr_primary_main.cf
- File location: /etc/VRTSagents/ha/conf/Sybase

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "VVRTypes.cf"
include "SybaseTypes.cf"

cluster cluster101 (
        UserNames = { admin = HopHojOlpKppNxpJom }
        ClusterAddress = "10.180.88.188"
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        UseFence=SCSI3
        )

remotecluster syb_cluster102 (
        ClusterAddress = "10.190.99.199"
        )

heartbeat Icmp (
        ClusterList = { syb_cluster102 }
        Arguments @syb_cluster102 = { "10.190.99.199" }

        )
system system1 (
        )
```

```
system system2 (
        )


group ClusterService (
        SystemList = { system1 = 0, system2 = 1 }
        AutoStartList = { system1, system2 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Application wac (
                StartProgram = "/opt/VRTSvcs/bin/wacstart"
                StopProgram = "/opt/VRTSvcs/bin/wacstop"
                MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
                RestartLimit = 3
                )

        IP gcoip (
                Device = eth0
                Address = "10.180.88.188"
                NetMask = "255.255.255.0"
                )

        NIC csgnic (
                Device = eth0
                )

        gcoip requires csgnic
        wac requires gcoip


        // resource dependency tree
        //
        //      group ClusterService
        //      {
        //      Application wac
        //          {
        //          IP gcoip
        //              {
        //              NIC csgnic
        //              }
        //          }
        //      }


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

        CVMVolDg sybdata_voldg (
                CVMDiskGroup = sybdata_101
                CVMActivation = sw
                )

        RVGShared sybdata_rvg (
                RVG = syb_rvg
                DiskGroup = sybdata_101
                )

        requires group binmnt online local firm
        sybdata_rvg requires sybdata_voldg



group binmnt (
        SystemList = { system1 = 0, system2 = 1 }
        Parallel = 1
        AutoStartList = { system1, system2 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        CFSMount sybbindg_101_sybbinvol_mnt (
                MountPoint = "/sybase"
                BlockDevice = "/dev/vx/dsk/sybbindg_101/sybbin_vol"
                )
        CVMVolDg sybbindg_101_voldg (
                CVMDiskGroup = sybbindg_101
                CVMVolume = { sybbin_vol }
                CVMActivation = sw
                )
        requires group cvm online local firm
        sybbindg_101_sybbinvol_mnt requires sybbindg_101_voldg

group cvm (
        SystemList = { system1 = 0, system2 = 1 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { system1, system2 }
        )

        CFSfsckd vxfsckd (
                )

        CVMCluster cvm_clus (
                CVMClustName = syb_cluster101
                CVMNodeId = { system1 = 0, system2 = 1 }
                CVMTransport = gab
                CVMTimeout = 200
                )

        CVMVxconfigd cvm_vxconfigd (
                Critical = 0
                CVMVxconfigdArgs = { syslog }
                )

        cvm_clus requires cvm_vxconfigd
        vxfsckd requires cvm_clus


        // resource dependency tree
        //
        //      group cvm
        //      {
        //      CFSfsckd vxfsckd
        //          {
        //          CVMCluster cvm_clus
        //              {
        //              CVMVxconfigd cvm_vxconfigd
        //              }
        //          }
        //      }


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

        IP logowner_ip (
                Device = eth0
                Address = "10.10.9.101"
                NetMask = "255.255.255.0"
                )

        NIC nic (
                Device = eth0
                )

        RVGLogowner rvg_logowner (
                RVG = syb_rvg
                DiskGroup = sybdata_101
                )

        requires group RVGgroup online local firm
        logowner requires logowner_ip
        logowner_ip requires nic


        // resource dependency tree
        //
        //      group logowner
        //      {
        //      RVGLogowner rvg_logowner
        //          {
        //          IP logowner_ip
        //              {
        //              NIC nic
        //              }
        //          }
        //      }

group sybasece (
        SystemList = { system1 = 0, system2 = 1 }
        Parallel = 1
        ClusterList = { syb_cluster101 = 0, syb_cluster102 = 1 }
        AutoStartList = { system1, system2 }
        ClusterFailOverPolicy = Manual
```

```
Authority = 1
OnlineRetryLimit = 3
TriggerResStateChange = 1
OnlineRetryInterval = 120
)

CFSMount quorum_101_quorumvol_mnt (
        MountPoint = "/quorum"
        BlockDevice = "/dev/vx/dsk/quorum_101/quorumvol"
        )

CFSMount sybdata_101_sybvol_mnt (
        MountPoint = "/sybdata"
        BlockDevice = "/dev/vx/dsk/sybdata_101/sybvol"
        )

CVMVolDg quorum_101_voldg (
        CVMDiskGroup = quorum_101
        CVMVolume = { quorumvol }
        CVMActivation = sw
        )

Process vxfend (
        PathName = "/sbin/vxfend"
        Arguments = "-m sybase -k /var/VRTSvcs/log/tmp/vcmp_socket"
        )

RVGSharedPri syb_vvr_shpri (
        RvgResourceName = sybdata_rvg
        OnlineRetryLimit = 0
        )

Sybase ase (
        Server @system1 = ase1
        Server @system2 = ase2
        Owner = sybase
        Home = "/sybase"
        Version = 15
        SA = sa
        Quorum_dev = "/quorum/q.dat"
        )

requires group RVGgroup online local firm
```

```
        sybdata_101_sybvol_mnt requires syb_vvr_shpri
        ase requires vxfend
        ase requires sybdata_101_sybvol_mnt
        ase requires quorum_101_quorumvol_mnt
        quorum_101_quorumvol_mnt requires quorum_101_voldg

        // resource dependency tree
        //
        //      group sybasece
        //      {
        //      Sybase ase
        //          {
        //          CFSMount sybdata_101_sybvol_mnt
        //              {
        //              RVGSharedPri syb_vvr_shpri
        //              }
        //          Process vxfend
        //          CFSMount quorum_101_quorumvol_mnt
        //              {
        //              CVMVolDg quorum_101_voldg
        //              }
        //          }
        //      }
```

Sample main.cf for a secondary CVM VVR site

This sample main.cf is for a secondary site in a CVM VVR configuration. It is the second of two sample main.cfs for replicating data between two clusters.

The following are the configuration details for this Sybase ASE CE configuration sample main.cf:

- File name: sybasece_cvmvvr_secondary_main.cf
- File location: /etc/VRTSagents/ha/conf/Sybase

```
This is main.cf for CVM VVR configuration on Secondary site.
--------------------------------------------------------------

include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "VVRTypes.cf"
include "SybaseTypes.cf"
```

```
cluster cluster102 (
        UserNames = { admin = HopHojOlpKppNxpJom }
        ClusterAddress = "10.190.99.199"
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        UseFence=SCSI3
        )

remotecluster syb_cluster101 (
        ClusterAddress = "10.180.88.188"
        )

heartbeat Icmp (
        ClusterList = { syb_cluster101 }
        Arguments @syb_cluster101 = { "10.180.88.188" }
        )

system system3 (
        )

system system4 (
        )

group ClusterService (
        SystemList = { system3 = 0, system4 = 1 }
        AutoStartList = { system3, system4 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Application wac (
                StartProgram = "/opt/VRTSvcs/bin/wacstart"
                StopProgram = "/opt/VRTSvcs/bin/wacstop"
                MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
                RestartLimit = 3
                )

        IP gcoip (
                Device = eth0
                Address = "10.190.99.199"
                NetMask = "255.255.255.0"
                )
        NIC csgnic (
                Device = eth0
                )

        gcoip requires csgnic
        wac requires gcoip

 // resource dependency tree
 //
 // group ClusterService
 // {
 // Application wac
 //     {
 //     IP gcoip
 //         {
 //         NIC csgnic
 //         }
 //     }
 // }


group RVGgroup (
        SystemList = { system3 = 0, system4 = 1 }
        Parallel = 1
        AutoStartList = { system3, system4 }
        )

        CVMVolDg sybdata_voldg (
                CVMDiskGroup = sybdata_101
                CVMActivation = sw
                )

        RVGShared sybdata_rvg (
                RVG = syb_rvg
                DiskGroup = sybdata_101
                )

        requires group binmnt online local firm
        sybdata_rvg requires sybdata_voldg


group binmnt (
        SystemList = { system3 = 0, system4 = 1 }
        Parallel = 1
        AutoStartList = { system3, system4 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        CFSMount sybbindg_101_sybbinvol_mnt (
                MountPoint = "/sybase"
                BlockDevice = "/dev/vx/dsk/sybbindg_101/sybbin_vol"
                )
        CVMVolDg sybbindg_101_voldg (
                CVMDiskGroup = sybbindg_101
                CVMVolume = { sybbin_vol }
                CVMActivation = sw
                )
        requires group cvm online local firm
        sybbindg_101_sybbinvol_mnt requires sybbindg_101_voldg

group cvm (
        SystemList = { system3 = 0, system4 = 1 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { system3, system4 }
        )

        CFSfsckd vxfsckd (
                )

        CVMCluster cvm_clus (
                CVMClustName = syb_cluster102
                CVMNodeId = { system3 = 0, system4 = 1 }
                CVMTransport = gab
                CVMTimeout = 200
                )

        CVMVxconfigd cvm_vxconfigd (
                Critical = 0
                CVMVxconfigdArgs = { syslog }
                )

        cvm_clus requires cvm_vxconfigd
        vxfsckd requires cvm_clus
        // resource dependency tree
        //
        //      group cvm
        //      {
        //      CFSfsckd vxfsckd
        //          {
        //          CVMCluster cvm_clus
        //              {
        //              CVMVxconfigd cvm_vxconfigd
        //              }
        //          }


group logowner (
        SystemList = { system3 = 0, system4 = 1 }
        AutoStartList = { system3, system4 }
        )

        IP logowner_ip (
                Device = eth0
                Address = "10.11.9.102"
                NetMask = "255.255.255.0"
                )

        NIC nic (
                Device = eth0
                )

        RVGLogowner rvg_logowner (
                RVG = syb_rvg
                DiskGroup = sybdata_101
                )

        requires group RVGgroup online local firm
        logowner requires logowner_ip
        logowner_ip requires nic


 // resource dependency tree
 //
 // group logowner
 // {
```

```
 // RVGLogowner rvg_logowner
 //     {
 //     IP logowner_ip
 //         {
 //         NIC nic
 //         }
 //     }
 // }

group sybasece (
        SystemList = { system3 = 0, system4 = 1 }
        Parallel = 1
        ClusterList = { syb_cluster102 = 0, syb_cluster101 = 1 }
        AutoStartList = { system3, system4 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        CFSMount quorum_101_quorumvol_mnt (
                MountPoint = "/quorum"
                BlockDevice = "/dev/vx/dsk/quorum_101/quorumvol"
                )

        CVMVolDg quorum_101_voldg (
                CVMDiskGroup = quorum_101
                CVMVolume = { quorumvol }
                CVMActivation = sw
                )

        CFSMount sybdata_101_sybvol_mnt (
                MountPoint = "/sybdata"
                BlockDevice = "/dev/vx/dsk/sybdata_101/sybvol"
                )

        Process vxfend (
                PathName = "/sbin/vxfend"
                Arguments = "-m sybase -k /var/VRTSvcs/log/tmp/vcmp_socket"
                )

        RVGSharedPri syb_vvr_shpri (
                RvgResourceName = sybdata_rvg
                OnlineRetryLimit = 0
                )
        Sybase ase (
                Server @system3 = ase1
                Server @system4 = ase2
                Owner = sybase
                Home = "/sybase"
                Version = 15
                SA = sa
                Quorum_dev = "/quorum/q.dat"
                )

requires group RVGgroup online local firm
sybdata_101_sybvol_mnt requires syb_vvr_shpri
ase requires vxfend
ase requires sybdata_101_sybvol_mnt
ase requires quorum_101_quorumvol_mnt
quorum_101_quorumvol_mnt requires quorum_101_voldg
```