Cluster Server 7.3.1 Agent for DB2 Installation and Configuration Guide - Linux

VERITAS™

Last updated: 2018-06-12

Legal Notice

Copyright © 2018 Veritas Technologies LLC. All rights reserved.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third-party (“Third-Party Programs”). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

https://www.veritas.com/about/legal/license-agreements

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

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
500 E Middlefield Road
Mountain View, CA 94043

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:

doc.feedback@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 DB2 7

About the Cluster Server Agent for DB2 7
How Cluster Server Agent for DB2 makes DB2 highly available 8
How the DB2 agent supports intelligent resource monitoring 8
Supported software for VCS agent for DB2 9
About agent functions for VCS Agent for DB2 9
About the online agent function for VCS agent for DB2 9
About the offline agent function for VCS agent for DB2 9
About the monitor agent function for VCS agent for DB2 10
About the clean agent function for VCS agent for DB2 10
About the info agent function for VCS agent for DB2 11
About the action agent function for VCS agent for DB2 11
About IMF Integration functions for VCS Agent for DB2 12
About running the info agent function to get database information for VCS agent for DB2 12
Typical DB2 configuration in a Cluster Server 13
Road map for setting up DB2 UDB in a VCS environment 14

Chapter 2 Installing and configuring DB2 16

VCS requirements for installing DB2 16
Preparing to install DB2 UDB non-MPP versions 16
Preparing to install DB2 UDB MPP versions 17
Creating the file systems for non-MPP instances for DB2 18
Creating the shared file systems for MPP instances for DB2 18
Defining user and group accounts for DB2 19
Installing DB2 in a VCS environment 21
Setting up the DB2 configuration 21
Checking /etc/services for DB2 22
Enabling password less communication for DB2 users among all cluster nodes 23
Modifying the $DB2InstHome/sqllib/db2nodes.cfg file for DB2 23
Confirming the DB2 installation 26

Chapter 3 Installing and removing the Cluster Server Agent for DB2 28

Before you install or upgrade the VCS Agent for DB2 28
Installing the VCS agent for DB2 software 29
Upgrading the VCS Agent for DB2 30
Disabling the VCS Agent for DB2 31
Removing the VCS Agent for DB2 32

Chapter 4 Configuring VCS service groups for DB2 33

About configuring service groups for DB2 33
About DB2 configurations in VCS 34
Non-MPP configuration service groups for DB2 35
MPP configuration service groups for DB2 36
Before configuring the service group for DB2 37
Importing the Db2udbTypes.cf file 38
Configuring the VCS Agent for DB2 from Cluster Manager (Java Console) 39
Adding service group for DB2 MPP using the Cluster Manager (Java Console) 39
Adding a service group for DB2 non-MPP with the Cluster Manager (Java Console) 41
Configuring the VCS Agent for DB2 by editing the main.cf file 43
Configuring the agent to use the DB2 MPP configuration 44
Configuring the agent to use the DB2 non-MPP configurations 45
Verifying the configuration for VCS agent for DB2 46
Setting up in-depth monitoring of a DB2 instance 47
Enabling in-depth monitoring of a DB2 instance 48
Enabling in-depth monitoring from the command line for DB2 48
Enabling and disabling intelligent resource monitoring for Db2udb resource 49
Handling DB2 error codes during in-depth monitoring 51
Disabling in-depth monitoring for DB2 52

Chapter 5 Administering VCS service groups for DB2 53

About administering service groups for DB2 53
Bringing the DB2 service group online 53
Taking the DB2 service group offline 54
Switching the DB2 service group 54
Disabling the agent for DB2 54

Appendix A Troubleshooting Cluster Server Agent for DB2 56

Creating a db2profile for environment variables for DB2 56
Setting the RestartLimit attribute for VCS agent for DB2 57
Partition/Node switch-over scenarios for DB2 V9.5, V9.7, V10.1, and V10.5 in MPP mode 57

Appendix B Resource type information for Cluster Server Agent for DB2 58

About the resource type definitions for DB2 58
DB2 UDB type definition file: Db2udbTypes.cf on Linux 58

Appendix C Resource type attributes for DB2 60

About the Db2udb resource types and attribute definitions 60

Appendix D Sample configurations 66

Sample configuration files for VCS agent for DB2 on Linux 66
Non-MPP main.cf configuration for VCS agent for DB2 on Linux 66
MPP main.cf configuration for VCS agent for DB2 on Linux 69

Index 75

Chapter 1

Introducing the Cluster Server Agent for DB2

This chapter includes the following topics:

- About the Cluster Server Agent for DB2
- How Cluster Server Agent for DB2 makes DB2 highly available
- How the DB2 agent supports intelligent resource monitoring
- Supported software for VCS agent for DB2
- About agent functions for VCS Agent for DB2
- Typical DB2 configuration in a Cluster Server
- Road map for setting up DB2 UDB in a VCS environment

About the Cluster Server Agent for DB2

The Cluster Server Agent for DB2, version 7.3.1, for DB2 UDB is a high availability solution for the relational database management system.

This guide describes the agent for DB2 UDB, its modes of operation, and its attributes. It describes how to install and configure the agent. It also describes how to administer service groups where the resource runs.

How Cluster Server Agent for DB2 makes DB2 highly available

The agent monitors DB2 UDB database processes at the partition level. If the system fails, the agent detects the failure and takes the applicable database partition offline. VCS moves the failed instances to another node in the cluster, where the agent brings the database partition or partitions online.

The agent performs different levels of monitoring and different actions, which you can configure. You can also configure many of the actions that the agent can perform.

The DB2 agent also supports IMF (Intelligent Monitoring Framework) in the process check mode of basic monitoring. IMF enables intelligent resource monitoring. The DB2 agent is IMF aware and uses asynchronous monitoring framework (AMF) kernel driver for resource state change notifications.

See "About agent functions for VCS Agent for DB2" on page 9.

How the DB2 agent supports intelligent resource monitoring

With intelligent monitoring framework (IMF), VCS supports intelligent resource monitoring in addition to 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 DB2.

See "Enabling and disabling intelligent resource monitoring for Db2udb resource" on page 49.

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 DB2 agent registers the PIDs of the processes with the AMF kernel driver using its own imf_register function. The agent's imf_getnotification function waits for any resource state changes. When the AMF kernel 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 appropriate action.

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

Supported software for VCS agent for DB2

Refer to the *Cluster Server Release Notes* for information on supported software and platform versions.

Refer to the support matrix at https://www.veritas.com/support/en_US/article.000126342.

About agent functions for VCS Agent for DB2

The agent can perform different operations or functions on the database. These functions are online, offline, monitor, clean, info, and action. With the action agent function, you can add custom actions for the agent.

For information on how to define custom actions, refer to the *Cluster Server Agent Developer's Guide*.

About the online agent function for VCS agent for DB2

The agent uses the db2gcf program to start a DB2 instance or database partition. The command is:

```
# su $DB2InstOwner -c "$InstHome/sqllib/bin/db2gcf -u -i\
    $DB2InstOwner -p $nodenum"
```

This agent can also use the `db2start` command to start all DB2 partitions.

Note: The agent uses the `db2start` command when the **UseDB2start** attribute is set to 1 and the partition is of non-MPP type.

About the offline agent function for VCS agent for DB2

The agent uses the db2gcf program to stop a DB2 database partition. For a database with only one partition, it stops the instance. The command is:

```
# su $DB2InstOwner -c "$InstHome/sqllib/bin/db2gcf -d -i\
    $DB2InstOwner -p $nodenum"
```

The agent can also use the `db2stop` command to stop all DB2 partitions.

Note: The agent uses the `db2stop` command when the **UseDB2start** attribute is set to 1 and the partition is of non-MPP type.

About the monitor agent function for VCS agent for DB2

The agent executes the `su $DB2InstOwner -c "$InstHome/sqllib/bin/db2gcf -s -i $DB2InstOwner -p $nodenum"` command to check the status of the database partition or node number. If the exit status of the `db2gcf` command is 0, the monitor returns the exit code 110. Otherwise, the monitor returns an exit code of 100 and the resource is taken offline. The agent then restarts or fails over the resource. This action depends on other type-independent attributes, such as RestartLimit or ToleranceLimit.

Note: In case of single partition, the monitor checks the process of the respective partition using ps -p command. And in case of multiple partions, db2gcf -s command is used

---

Set the IndepthMonitor attribute to 1 for in-depth monitoring. The agent looks for the monitor_custom_$db2instance_$nodenum file in the /opt/VRTSagents/ha/bin/Db2udb directory. It executes this customized in-depth monitor file, if the file exists and is executable. You can find samples of custom monitor scripts in the /etc/VRTSagents/ha/conf/Db2udb/sample_db2udb directory.

If the custom monitor has any errors or problems, it checks the value of the WarnOnlyIfDBQueryFailed attribute of the Db2udb agent. If you have a db2error.dat file in the /opt/VRTSagents/ha/bin/Db2udb directory, the agent checks this file, and handles the error according to the error configuration.

See "Handling DB2 error codes during in-depth monitoring" on page 51.

If you set the WarnOnlyIfDBQueryFailed attribute to 1 (its default), and you have configured the Notifier resource, the agent performs the following actions:

- Sends a notification
- Returns the exit code 110

If you set the WarnOnlyIfDBQueryFailed attribute to 0, it performs error handling in the db2error.dat file. Note that the file needs to exist to perform error handling. If it does not exist, it returns the exit code 100, which is the default.

About the clean agent function for VCS agent for DB2

The agent first uses the `db2stop` command with the force option. If the command is unable to clean the agent function, the agent then uses the db2_kill script to kill the instance. If the db2_kill script fails, the `kill -9` command is invoked. The command `ipclean` is used to ensure all the DB2 interprocess communications are cleaned for the particular instance.

```
# su $DB2InstOwner -c "$InstHome/sqllib/adm/db2stop force
    nodenum $nodenum"
```

About the info agent function for VCS agent for DB2

The agent supports the info function, which provides static and dynamic information about the database partition and its critical processes.

For more information about the Info agent function, see the following guides:

- *Cluster Server Administrator's Guide*
- *Cluster Server Agent Developer's Guide*

See "About running the info agent function to get database information for VCS agent for DB2" on page 12.

About the action agent function for VCS agent for DB2

The agent for DB2 supports the action function, which enables you to perform predefined actions or custom 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]
```

The agent supports the following predefined actions:

- The VRTS_GetInstanceName token retrieves the DB2 instance name of the configured Db2udb resource.
- The VRTS_GetRunningServices token retrieves the list of processes that the agent monitors for the Db2udb resource.

For example:

```
# hares -action db2udb1 VRTS_GetInstanceName -sys systemName

VCS NOTICE V-16-13323 Resource (db2udb1): action
(VRTS_GetInstanceName) completed successfully. Output is:

db2inst1

# hares -action db2udb0 VRTS_GetRunningServices -sys systemName

VCS NOTICE V-16-13323 Resource (db2udb0): action
(VRTS_GetRunningServices) completed successfully. Output is:

PARTITION: 0
```

```
PID TTY TIME CMD

9800 ? 0:06 db2sysc
```

About IMF Integration functions for VCS Agent for DB2

The following sections describe the IMF Integration functions for VCS Agent for DB2.

About the imf_init function for VCS agent for DB2

Initializes the agent to interface with the AMF kernel driver, which is the IMF notification module for DB2 agent. This function runs when the agent starts up.

About the imf_register function for VCS agent for DB2

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).

About the imf_getnotification function for VCS agent for DB2

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.

About running the info agent function to get database information for VCS agent for DB2

You can run the info agent function to return database information. In this example, the info agent function retrieves the database information.

To retrieve database information

1. Make the configuration writable:

```
# haconf -makerw
```

2. Specify the periodic interval in seconds that the info agent function is invoked. The default value of 0 means info agent function is not invoked.

```
# hatype -modify Db2udb InfoInterval 300
```

   In this command, Db2udb is the name of the DB2 resource type. InfoInterval 300 is the duration (in seconds) after which the info agent function executes the info script. The script gets the information of the processes that the agent monitors.

3. Show the requested ResourceInfo value. The following example output shows the processes that the agent monitors for the DB2 resource.

   The value needs to be overridden at the resource level using the following command:

```
#  hares -override db2udb3_2 InfoInterval
```

   Note that ResourceInfo refreshes every 300 seconds (five minutes), since you set the InfoInterval to 300 in the previous step.

```
# hares -value db2udb1 ResourceInfo
State Valid
Msg
PARTITION: 0
    PID    TTY  TIME CMD
  413924    -   0:00 db2sysc

TS Fri Jan 14 18:11:52 2011
```

4. Make the configuration permanent:

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

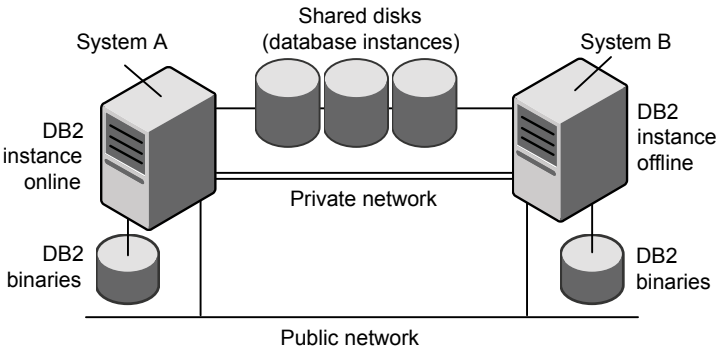
Typical DB2 configuration in a Cluster Server

DB2 UDB is configured on a two-system cluster. DB2 UDB system binaries are installed identically on local file systems on System A and System B. The instance home directory, instance binaries, and the database reside on shared storage, available to either node.

In the case of the non-MPP configuration, an instance is online on only one system at a time. The other system is the failover system.

Figure 1-1 shows a DB2 installation with a non-MPP configuration.

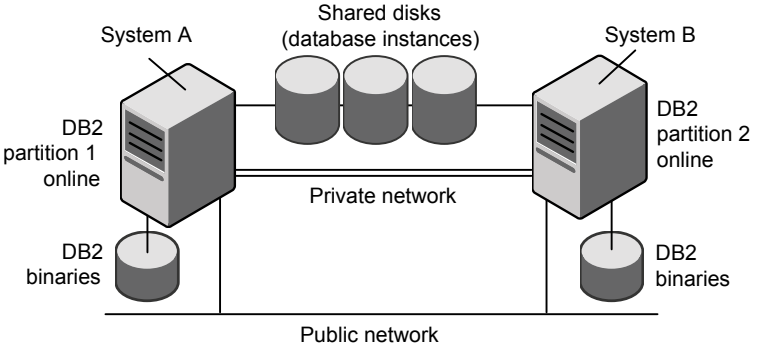
Figure 1-1 DB2 installation with a non-MPP configuration



In the case of the MPP configuration, a database partition can run on each system and each system can become a failover system.

Figure 1-2 shows a DB2 installation with an MPP configuration.

Figure 1-2 DB2 installation with an MPP configuration



Road map for setting up DB2 UDB in a VCS environment

Review the following tasks and information:

- Check the supported software.

- Review the agent functions.
  See “About agent functions for VCS Agent for DB2” on page 9.
- Check the agent requirements.
  See “VCS requirements for installing DB2” on page 16.
- Install and set up DB2.
  See “Installing DB2 in a VCS environment” on page 21.
  See “Setting up the DB2 configuration” on page 21.
- Install the VCS agent for DB2.
  See “Installing the VCS agent for DB2 software” on page 29.
- Configure the service groups for the VCS agent for DB2.
  See “Configuring the VCS Agent for DB2 from Cluster Manager (Java Console)” on page 39.
- Optionally, configure in-depth monitoring and any automated actions.
  See “Setting up in-depth monitoring of a DB2 instance” on page 47.
- Bring the service group online.

Chapter 2

Installing and configuring DB2

This chapter includes the following topics:

- VCS requirements for installing DB2
- Installing DB2 in a VCS environment
- Setting up the DB2 configuration

VCS requirements for installing DB2

Perform the following tasks before you install:

- Review preparing to install DB2 UDB non-MPP versions.
  See "Preparing to install DB2 UDB non-MPP versions" on page 16.
- Review preparing to install DB2 UDB MPP versions.
  See "Preparing to install DB2 UDB MPP versions" on page 17.
- Create the file systems for non-MPP instances.
  See "Creating the file systems for non-MPP instances for DB2" on page 18.
- Create the shared file systems for MPP instances.
  See "Creating the shared file systems for MPP instances for DB2" on page 18.
- Define user and group accounts.
  See "Defining user and group accounts for DB2" on page 19.

Preparing to install DB2 UDB non-MPP versions

Review the following list:

- Verify that all nodes in the cluster have adequate resources to run DB2 and VCS.
- Make sure that you meet the VCS requirements to install DB2.
- Define DB2 UDB user and group accounts.
  See "Defining user and group accounts for DB2" on page 19.
  For your particular version of DB2 UDB, refer to the appropriate DB2 UDB guide.
- Install the DB2 UDB system binaries locally.
- Install the DB2 UDB database instances on shared storage.
- Install and configure VCS version 7.3.1 on all nodes in the cluster. For installation instructions, refer to the *Cluster Server Installation Guide*.

Preparing to install DB2 UDB MPP versions

Review the following list:

- Verify that all systems have enough resources to run DB2 UDB. An example of this kind of resource is shared memory. Check the memory requirements, which vary due to the version and the hardware configuration of DB2.
- Define DB2 UDB user and group accounts.
  See "Defining user and group accounts for DB2" on page 19.
  For your particular version of DB2 UDB, refer to the appropriate DB2 UDB guide.
- Install the DB2 UDB system binaries on the local file systems on each system.
- Install the DB2 UDB database instances on shared storage.
- Install and configure VCS version 7.3.1 on all nodes in the cluster. For installation instructions, refer to the *Cluster Server Installation Guide*.
- The MPP configuration requires the Storage Foundation Cluster File System software.

  This software includes the following:

  - Cluster Server (VCS)
  - Volume Manager with cluster functionality enabled (CVM)
    Storage Foundation Cluster File System High Availability Installation Guide.
  - File System with cluster functionality enabled (CFS)
    Storage Foundation Cluster File System Administration Guide.

Creating the file systems for non-MPP instances for DB2

To create a file system for non-MPP instances, you first create a disk group on the physically shared disk. You then create a volume of sufficient size within the disk group.

To create a file system for non-MPP instances on Linux

1. `# vxdg init db2db_dg sdc`

   Deport and import the disk group:

   ```
   # vxdg deport db2db_dg
   # vxdg import db2db_dg
   ```

2. Create a volume using the `vxassist` command:

   ```
   # vxassist -g db2db_dg make db2db_vol 3g
   ```

3. Create the file system:

   ```
   # mkfs -t vxfs /dev/vx/dsk/db2db_dg/db2db_vol
   ```

4. Create the mount point directory and mount the file system. Make sure that the mount point exists on all nodes in the cluster on the local file system—not on shared storage.

   ```
   # mkdir /db2_mnt/db2inst1
   # mount -t vxfs /dev/vx/dsk/db2db_dg/db2db_vol \
     /db2_mnt/db2inst1
   ```

Creating the shared file systems for MPP instances for DB2

Install the Storage Foundation Cluster File System software. To create a shared file system for MPP instances, create a shared disk group on the physically shared disk. You then create a volume of sufficient size within the disk group.

To create a shared file system for MPP instances on Linux

1. From one of the nodes, create the disk group. List the disks using the `vxdisk list` command.
2. Create a shared disk group. In this case, the group consists of one disk. In this example the disk is sdc:

   ```
   # vxdg -s init db2db_dg sdc
   ```

3 Deport and import the disk group:

```
# vxdg deport db2db_dg
# vxdg -s import db2db_dg
```

4 Use the `vxassist` command to create a volume:

```
# vxassist -g db2db_dg make db2db_vol 7g
```

5 Create the file system:

```
# mkfs -t vxfs -o largefiles /dev/vx/rdsk/db2db_dg/db2db_vol
```

6 Create the mount point directory and mount the file system.

```
# mkdir /db2_mnt/db2inst1
# mount -t vxfs -o cluster /dev/vx/dsk/db2db_dg/db2db_vol \
  /db2_mnt/db2inst1
```

Defining user and group accounts for DB2

Before you install DB2 UDB binaries and create instances, you must define DB2 UDB user and group accounts for each instance on each system.

Note the following requirements:

- The IDs for DB2 users and groups must be exactly the same across all cluster nodes.
- The DB2 instance owner's home directory must be accessible from each node. This directory is the mount point that the DB2 instance uses. The database that you want to mount must be on shared storage. Create the mount point directory locally on each node, if it does not already exist.
- All DB2 user accounts must exist on the local systems. Veritas does not recommend the use of NIS or NIS+ for users, because these services are not highly available. If their service is interrupted, VCS may not be able to work correctly.

Creating user group accounts for DB2

Three user group accounts are required on each node in the cluster.

To create the group accounts on each node in the cluster on Linux

1. Create a group for the DB2 UDB instance owner. For example, enter:

```
# groupadd -g 999 db2iadm1
```

2. Create a group for the user to execute fenced user-defined functions (UDFs) or store procedures. For example, enter:

```
# groupadd -g 998 db2fadm1
```

3. Create a group for the database administration server. For example, enter:

```
# groupadd -g 997 dasadm1
```

Adding user accounts for DB2 on Linux

In the following examples that show creating user accounts, you can use the following options:

- The -g option specifies the group
- The -u option specifies the user ID
- The -d option specifies the home directory
- The -m option creates the home directory if it doesn't exist
- The -s option is the user's logon shell
- The final expression is the user's logon.

Create the user accounts on each node in the cluster.

This example shows how to create the user, db2inst1, who is the DB2 UDB instance owner. The instance's home directory is also the mount point: /db2_mnt/db2inst1. The DB2 UDB instance home directory must exist on every node. For example:

```
# useradd -g db2iadm1 -u 1004 -d /db2_mnt/db2inst1 -m -s \
/bin/ksh db2inst1
```

The following examples show how to create user accounts for db2fenc1 and dasusr1. These users' home directories are under /home in the local file system on each node.

```
# useradd -g db2fadm1 -u 1003 -d /home/db2fenc1 -m -s \
/bin/ksh db2fenc1
# useradd -g dasadm1 -u 1002 -d /home/dasusr1 -m -s /bin/ksh dasusr1
```

For DB2 configuration in MPP mode the file system that uses the mount point: /db2_mnt/db2inst1 uses Cluster File System to hosts the DB2 UDB instance home directory.

Installing DB2 in a VCS environment

For installing DB2 in a VCS environment, Veritas recommends that you follow the installation procedure in the relevant IBM DB2 UDB guide.

Install binaries on local disks of each node, and the database instances on shared storage, accessible by each cluster node.

To install DB2 in VCS environment

1. Set shared memory parameters. Refer to the relevant IBM DB2 UDB guide to make sure that memory requirements are met.
2. Install the binaries. Install the DB2 UDB system binaries on local disks on each node (mirrored disks are recommended) not on shared storage. You can use IBM's db2setup tool.
3. Install the DB2 license. Install the DB2 license on each node. For example, enter:

   ```
   # /opt/IBM/db2/V9.7/adm/db2licm -a db2ese.lic
   ```

4. Install the instances. Install the database instances on the shared storage only on the one node where the instance's home directory is currently mounted. You can choose to install single-partition instance or multi-partition instance. You can use IBM's db2setup tool.

   Keep in mind the following when you install:

   - When you use the db2setup, do not select the option to Auto start DB2 instance at system boot in the DB2 Instance Properties window. Note that this option does not exist on all DB2 versions. VCS needs to bring up the resources for the DB2 instances in a specific order before it brings the instance online.
   - The instance's home directory is a mount point on the local system.

Note: For more information on DB2 installation, refer IBM documentation.

Setting up the DB2 configuration

Use the following procedures to configure DB2 UDB in a VCS environment.

You need to do the following tasks:

- Check /etc/services.
  See “Checking /etc/services for DB2” on page 22.
- Enable password less communication for DB2 user among all cluster nodes.
  See “Enabling password less communication for DB2 users among all cluster nodes ” on page 23.
- Modify the $DB2InstHome/sqllib/db2nodes.cfg file for DB2.
  See “Modifying the $DB2InstHome/sqllib/db2nodes.cfg file for DB2” on page 23.
- Confirm the setup of the DB2 installation.
  See “Confirming the DB2 installation” on page 26.

Checking /etc/services for DB2

On each system in the cluster, use the `more` command to check the file /etc/services.

Remember to perform the following:

- Make sure each partition has a port number assigned. The number of reserved ports depends on the number of partitions.
- Make sure that no other services use the ports. Manually assign new numbers if necessary.
- Make sure all systems in the cluster have the same entries in the /etc/services file.

The following is an example for two DB2 instances: db2inst1 and db2inst2. Both instances have two partitions each. Each instance requires two ports plus one port per partition, hence four lines per instance.

```
# more /etc/services
DB2_db2inst1      60000/tcp
DB2_db2inst1_1    60001/tcp
DB2_db2inst1_2    60002/tcp
DB2_db2inst1_END  60003/tcp
DB2_db2inst2      60004/tcp
DB2_db2inst2_1    60005/tcp
DB2_db2inst2_2    60006/tcp
DB2_db2inst2_END  60007/tcp
```

Inspect the file and verify that no duplicate port numbers exist.

Enabling password less communication for DB2 users among all cluster nodes

By default, DB2 database system uses rsh as the communication protocol when starting remote database partitions. Configure passwordless rsh so that it does not prompt for passwords while starting a remote partition. If system communication uses ssh, you must configure it so that it operates without requests for passwords or passphrases.

Modifying the $DB2InstHome/sqllib/db2nodes.cfg file for DB2

DB2 uses the $DB2InstHome/sqllib/db2nodes.cfg file during failover from one node to another.

To modify the $DB2InstHome/sqllib/db2nodes.cfg file for non-MPP versions

1. Open the $DB2InstHome/sqllib/db2nodes.cfg file.
2. Add an entry for database partition.
3. For IPv4, assign the virtual IP address as the hostname, for example for this step and the previous step:

   ```
   0 virtualhostname 0
   1 virtualhostname 1
   ```

   Note that the *virtualhostname* corresponds to the virtual IP address in the /etc/hosts file. Make sure that the virtual IP address is up and works.
4. For IPv6, DB2 has a limitation that the hostname entries in db2nodes.cfg need to correspond to the hostname of the node. For example, if the hostname is sysA-v6, db2nodes.cfg should contain sysA-v6.

   With VCS, use virtual hostnames. When the service group fails over to another node, the virtual IP address is brought up on the target node. As a workaround for the limitation where the hostname entries in db2nodes.cfg need to correspond to the hostname of the node, you can use something similar to the following configuration.

   On each node, create a /local/$DB2InstHome/db2nodes.cfg file with same permission as that of $DB2InstHome and create a symbolic link to this file from $DB2InstHome/sqllib/db2nodes.cfg.

   ```
   # mkdir -p /local/$DB2InstHome/
   # ln -s /local/$DB2InstHome/db2nodes.cfg \
   $DB2InstHome/sqllib/db2nodes.cfg
   ```

The path /local/$DB2InstHome/db2nodes.cfg is local to each system. Enter the IPv6 hostname in the/local/$DB2InstHome/db2nodes.cfg on each node.

For example, on nodes sysA-v6 and sysB-v6, enter the following in their respective db2nodes.cfg files.

- On sysA-v6:

```
0 sysA-v6 0
```

- On sysB-v6:

```
0 sysB-v6 0
```

To modify the $DB2InstHome/sqllib/db2nodes.cfg file for MPP versions

1 For MPP versions, modify the file $DB2InstHome/sqllib/db2nodes.cfg with the hostname that you want each database partition to start on. DB2 automatically changes and updates the db2nodes.cfg file to enable the database partitions to fail over from one node to another. DB2 adds a fourth column for the "netname". Netname is an IP address of the high speed interconnect for the communication of Fast Communication Manager (FCM). Netname is required only if you are using a high speed interconnect for database partition communications. If the netname is not specified, it is by default the hostname. The virtual IP is not used in the db2nodes.cfg file for MPP configurations when we configure more than one database partition.

For example when the high speed interconnect (FCM switch) is not used:

```
0 sysA 0
1 sysB 0
2 sysC 0
3 sysD 0
```

For example when the high speed interconnect (FCM switch) is used for inter-partition communication:

```
0 sysA 0 switchA
1 sysB 0 switchB
2 sysC 0 switchC
3 sysD 0 switchD
```

Make sure that the relative port number in the third column is unique for each partition on a host. For example when the high speed interconnect (FCM switch) is not used:

```
0 sysA 0
1 sysA 1
2 sysB 0
3 sysC 0
4 sysD 0
```

For example when the high speed interconnect (FCM switch) is used for inter-partition communication:

```
0 sysA 0 switchA
1 sysA 1 switchA
2 sysB 0 switchB
3 sysC 0 switchC
4 sysD 0 switchD
```

2 If a high speed communication switch/netname is configured for inter partition communication, add entry of the switchname/netname in the db2 global registry using db2greg command before bringing the partitions online using db2start or db2gcf commands. Execute db2greg command, after sourcing the db2profile from DB2 Instance's home directory as mentioned below

```
# source /db2_mnt/db2inst1/sqllib/db2profile

# db2greg -addvarrec variable=switchname, value=$switch
```

To verify the entry of switchname is the db2 global registry use the below command

```
# db2greg -dump
```

Confirming the DB2 installation

On the host where the shared file system is mounted, check whether you can start and stop each instance. Do this procedure to verify the DB2 installation.

To check if a DB2 instance can start and stop

1 Log on as the instance owner:

```
# su - db2inst1
```

2 If you are configuring DB2 in an IPv6 environment, apply the setting for variable DB2FCMCOMM:

```
# db2set DB2FCMCOMM=TCPIP6
```

3 Attempt to start the instance:

```
$ db2start
```

DB2 should start on the partitions in the db2nodes.cfg file. If DB2 does not start, check the error codes.

4 Assuming that the previous command is successful, stop the instance:

```
$ db2stop
```

5 If the application does not stop correctly on each node, check for configuration errors. Review the DB2 UDB documentation for error codes.

6 Create a database.

```
$ db2 create database dbname
```

7 List the database directory

```
$ db2 list database directory
```

To check the rest of the DB2 configuration in the cluster

1 For each node in the VCS cluster, import the disk group and start all the volumes in the disk group.

2 Mount the file system for the volume that contains the DB2 instance and database.

3 Unmount and deport the disk group.

4 Repeat this procedure for each node in the cluster.

Chapter 3

Installing and removing the Cluster Server Agent for DB2

This chapter includes the following topics:

- Before you install or upgrade the VCS Agent for DB2
- Installing the VCS agent for DB2 software
- Upgrading the VCS Agent for DB2
- Disabling the VCS Agent for DB2
- Removing the VCS Agent for DB2

Before you install or upgrade the VCS Agent for DB2

Meet the following prerequisites to install or upgrade the Veritas High Availability Agent for DB2:

- Meet VCS requirements for installing DB2.
  See "VCS requirements for installing DB2" on page 16.
- Install DB2 in a VCS environment.
  See "Installing DB2 in a VCS environment" on page 21.
- Make sure that VCS is installed in the cluster.
- Verify that DB2 is installed and configured.

- VCS Single Cluster Manager (Java Console) is no longer packaged with VCS. Veritas recommends use of the Veritas InfoScale Operations Manager to manage, monitor and report on multi-cluster environments. You can download Veritas InfoScale Operations Manager at https://sort.veritas.com/vom . If you wish to continue using the VCS Single Cluster Manager, you can get it at no charge at the https://www.veritas.com/product/storage-management/infoscale-operations-manager website.

Installing the VCS agent for DB2 software

The agent binaries for Cluster Server Agent for DB2 is part of 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.

You can install the Cluster Server Agent for DB2 from the product disc. You must install the VCS Agent for DB2 enterprise agent on all nodes that will host the DB2 service group.

To install the agent on a Linux node

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 DB2, you can skip this procedure.
3. Insert the software disc into a drive that is connected to the node.

   The software automatically mounts the disc as /mnt/cdrom.
4. If the disc does not automatically mount, then you must mount the disc manually. For example:

   ```
   # mount -t iso9660 -o ro /dev/cdrom /mnt/cdrom
   ```
5. Install the agent software.
   - For RHEL6 or supported RHEL-compatible distributions on x86_64:

     ```
     # cd /mnt/cdrom/rhel6_x86_64/rpms

     # rpm -i VRTSvcsea-7.3.1.0.000-RHEL6.x86_64.rpm
     ```
   - For RHEL7 or supported RHEL-compatible distributions on x86_64:

     ```
     # cd /mnt/cdrom/rhel5_x86_64/rpms
     ```

```
# # rpm -i VRTSvcsea-7.3.1.0.000-RHEL7.x86_64.rpm
```

- For SLES11 on x86_64:

```
# cd /mnt/cdrom/sles11_x86_64/rpms

# rpm -i VRTSvcsea-7.3.1.0.000-SLES11.x86_64.rpm
```

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

6 Repeat step 1 through step 4 on each node that you want to be part of the DB2 service group.

Upgrading the VCS Agent for DB2

To upgrade without saving previous configuration information you must disable the agent, remove it, and reinstall it.

The steps to upgrade and re-use previous configuration information is as follows:

To upgrade from DB2 agent

1 Disable and remove the agent.

For more information see the following:

- Disabling the agent for DB2
  See “Disabling the VCS Agent for DB2” on page 31.
- Removing the agent for DB2 UDB
  See “Removing the VCS Agent for DB2” on page 32.

2 From the disc that has the HA agent for DB2, add the new package.

See “Installing DB2 in a VCS environment” on page 21.

3 Make sure that you have copied the Db2udbTypes.cf file from the /etc/VRTSagents/ha/conf/Db2udb/ directory to the /etc/VRTSvcs/conf/config directory.

Make sure to update the newly copied Db2udbTypes.cf file with all the type-level changes that you had made to the older Db2udbTypes.cf file. For example, if you had changed the value of the MonitorInterval attribute from the default 60 to 120 seconds, the Db2udbTypes.cf file gets updated. You must apply these type-level changes manually to the newly copied Db2udbTypes.cf file.

4 To continue to use in-depth monitoring, use the custom monitoring sample script or any user-defined scripts.

See "Handling DB2 error codes during in-depth monitoring" on page 51.

Disabling the VCS Agent for DB2

To disable the Cluster Server Agent for DB2, you must change the VCS Agent for DB2 service group to an OFFLINE state. You can stop the application completely or switch the agent to another system.

To disable the agent

1 To remove a system from the service group's SystemList, check if the service group is online:

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

2 If the service group is online, take it offline. Use one of the following commands:

- To take the service group offline on one node and online it on another node, you can use the -switch option:

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

- To take the service group offline without bringing it online on any other node in the cluster, enter:

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

3 Stop the agent on the node:

```
# haagent -stop Db2udb -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 confirm the agent is stopped.

5 You can now remove the service group, the resource type, or both from the VCS configuration after disabling the agent on all nodes.

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

Removing the VCS Agent for DB2

Before you remove the agent you must disable it on all nodes. Note that the agent package VRTSvcsea includes the VCS agents for DB2, Oracle, and Sybase. The following procedure removes all the VCS agents.

See "Disabling the VCS Agent for DB2" on page 31.

To remove the agent from a node

- On each system that has the agent, type:

```
# rpm -e VRTSvcsea
```

Chapter 4

Configuring VCS service groups for DB2

This chapter includes the following topics:

- About configuring service groups for DB2
- About DB2 configurations in VCS
- Before configuring the service group for DB2
- Configuring the VCS Agent for DB2 from Cluster Manager (Java Console)
- Configuring the VCS Agent for DB2 by editing the main.cf file
- Setting up in-depth monitoring of a DB2 instance

About configuring service groups for DB2

Configuring the DB2 service group involves creating the DB2 service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the agent using the following methods:

- Use the VCS Cluster Manager (the Java Console).
  You can use the Cluster Manager (Java Console) to edit a service group template for the Cluster Server Agent for DB2.
  See "Configuring the VCS Agent for DB2 from Cluster Manager (Java Console)" on page 39.
- Use the Veritas InfoScale Operations Manager.
  Refer to the Veritas InfoScale Operations Manager documentation for details.
- Use VCS commands.

From the command line, you can configure the agent. For information about configuring service groups from the command line, see the *Cluster Server Administrator's Guide*

- Edit the main.cf file.
  You can edit the main.cf file. This method requires that you stop and restart VCS before the new configuration takes effect.

  For more information on the main.cf file refer to the following:

  - You can use the `hacf -verify` command to ensure that the modified main.cf file works. For more information on this command, refer to the *Cluster Server Administrator's Guide*.
  - Configuring the Cluster Server Agent for DB2 by editing the main.cf file
    See "Configuring the VCS Agent for DB2 by editing the main.cf file" on page 43.

About DB2 configurations in VCS

You can configure DB2 service groups in MPP and non-MPP configurations.

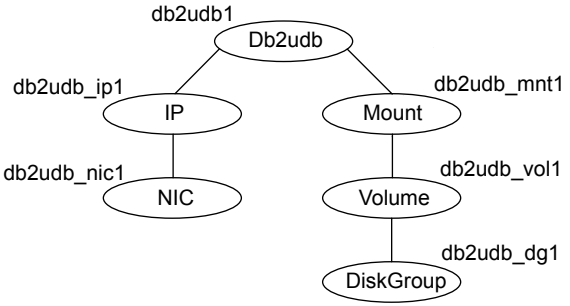
DB2 Installation Mode	Virtual Host configured and has entry in db2nodes.cfg	Contents of Nodes.cfg	Attribute value of UseDB2Start
NON-MPP (Single Partition)	Yes. Entries in the existing db2nodes.cfg is modified by adding virtual hostname. Example: db2_server	Example: 0 db2_server 0	Set UseDB2start = 1. When UseDB2start is set to 1, use `db2start` command to start the partition. After the failover or switchover happens the entry in db2nodes.cfg remains the same. **Note:** In this configuration the password less rsh must be set for the DB2 Instance Owner.

DB2 Installation Mode	Virtual Host configured and has entry in db2nodes.cfg	Contents of Nodes.cfg	Attribute value of UseDB2Start
NON-MPP (Single Partition)	No	Example: 0 sys1 0 sys1 DB2 Installation automatically creates the above entry in db2nodes.cfg file during its installation. **Note:** sys1/sys2 is the hostname of the physical machine.	Set UseDB2start = 0. When UseDB2Start is set to 0, use `db2gcf` command to start the partition. After the failover/switchover happens the entry in db2nodes.cfg is updated by the hostname of the physical nodes on which the failover/switchover has happened. For example, when failover/switchover happens from machine sys1 to machine sys2. The existing entry in db2nodes.cf [0 sys1 0 sys1] gets changed to [0 sys2 0 sys2].
MPP (Multiple Partitions)	No	Example: ■ 0 sys1 0 ■ 1 sys1 1 ■ 2 sys2 0	Set UseDB2start = 0. Since the number of partitions are greater that one, always use `db2gcf` command to start the partitions.

Non-MPP configuration service groups for DB2

Figure 4-1 illustrates the dependencies among the resources that are configured for a non-MPP DB2 instance resource group.

Figure 4-1 Dependency graph for a DB2udb resource



This configuration shows a service group for a Db2udb resource. The db2udb1 resource (the database) requires the IP resource and the Mount resource. The service group IP address for the DB2 server is configured using the IP resource (db2udb_ip1) and the NIC resource (db2udb_nic1). The mount resource (db2udb_mnt1) requires the Volume resource (db2udb_vol1), which in turn requires the DiskGroup resource (db2udb_dg1). You can start the service group after each of these resources is available.

MPP configuration service groups for DB2

The DB2 agent uses two service groups to support MPP configuration.

These service groups are as follows:

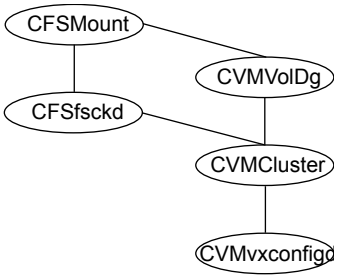
- Parallel CVM
- DB2 failover

Parallel CVM service group for MPP configuration for DB2

One CVM/Infrastructure group per cluster node exists. This service group has the CVM resource and the necessary resources for support of CFS. This group also contains all common components that DB2 needs, such as the instance's home directory, which is shared on all the cluster nodes.

Figure 4-2 illustrates the parallel CVM service group, which is the first of two service groups that support the MPP configuration.

Figure 4-2 Parallel CVM service group

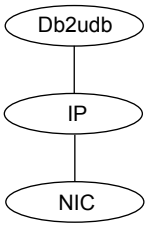


Failover service group for MPP configuration for DB2

This service group monitors one database partition with MPP configuration. The failover DB2 service group depends on the parallel CVM service group with online local firm dependency.

Figure 4-3 illustrates the DB2 failover service group, which is the second of two service groups that support the MPP configuration.

Figure 4-3 DB2 failover service group



Before configuring the service group for DB2

Before you configure the DB2 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 installing VCS see the *Cluster Server Installation Guide*.
- Verify that DB2 is installed and configured identically on all nodes in the cluster. See “Installing DB2 in a VCS environment” on page 21.

- Verify that the VCS agent for DB2 is installed on all nodes in the cluster.
  See "Installing DB2 in a VCS environment" on page 21.

Importing the Db2udbTypes.cf file

Before you use the DB2 UDB templates, use the Cluster Manager (Java Console) to import the Db2udbTypes.cf file to the VCS engine.

Note: Before you start the following procedure, for more instructions on Java GUI, refer to the *Cluster Server Installation Guide*.

To import the Db2udbTypes.cf file using the Cluster Manager (Java Console)

1. On one of the nodes in the cluster, start the Cluster Manager (Java Console). Type:

   ```
   # hagui&
   ```

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/Db2udb/Db2udbTypes.cf
   ```

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

   When the Db2udb types are imported to the VCS engine, the Db2udb agent can be configured.

   Refer to the following:

   - If you use the DB2 UDB MPP configuration:
     See "Adding service group for DB2 MPP using the Cluster Manager (Java Console)" on page 39.
   - If you use the DB2 UDB non-MPP configuration:
     See "Adding a service group for DB2 non-MPP with the Cluster Manager (Java Console)" on page 41.

To import the Db2udbTypes.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 Db2udbTypes.cf file.

   ```
   # include "Db2udbTypes.cf"
   ```

   The DB2 types definition is imported to the VCS engine. The agent for DB2 can be configured without interrupting or stopping VCS.

Configuring the VCS Agent for DB2 from Cluster Manager (Java Console)

Templates for the DB2 resource groups were automatically installed when you installed the agent for DB2. Using the Cluster Manager (Java Console), you can use the template (/etc/VRTSagents/ha/Templates/Db2udbGroup.tf) to configure the DB2 service group, its resources, and their attributes. You can also use the Cluster Manager (Java Console) to dynamically modify the attributes' values as necessary for your configuration.

Adding service group for DB2 MPP using the Cluster Manager (Java Console)

If you have imported the Db2udbTypes.cf file, you can use the template (/etc/VRTSagents/ha/Templates/Db2udbGroup.tf) to configure a service group.

See "Importing the Db2udbTypes.cf file" on page 38.

After you log into the Cluster Manager, the **Status** tab indicates that the CVM service group is online on each system in the cluster. The CVM service group is automatically configured when you complete the installation of the Storage Foundation Cluster File System (SFCFS) software.

To start adding the service group for the DB2 MPP database

1. In the Cluster Explorer window, click the **Add Service Group** icon on the toolbar.
2. In the Add Service Group window, enter the name of the service group you want to create for the service group. For example, enter `db2mpp_grp1`. Do not press Return or Enter.
3. In the Available Systems box, double-click the systems that you want in your configuration.
4. Select the system where you want to automatically start the service group. In the window that shows the systems in the configuration, select the check box next to the system.
5. Click the **Failover** radio button to specify the Service Group Type.

To select the template for the DB2 MPP database

1. Click the Templates button.
2. In the Select Templates window, select db2udb_mpp_grp from the list in the Templates box. The Dependency graph information and the Types information should change to reflect the template choice. Click **OK**. In the Add Service Group window, the name of the template is now shown as selected.
3. Click **OK** on at the bottom of the Add Service Group window. The group is added. On the left pane of the Cluster Manager (Java Console) window, the service group is under the CVM service group. On the Status tab, the group is shown Offline on each system.

To configure the resources for the DB2 MPP database

1. In the left pane, double-click the db2mpp_grp1 service group. The types of resources that you can configure for the group appear: Db2udb, IP, and NIC.
2. Double-click the Db2udb resource type. Select the resource, db2udb, which is under the Db2udb type. Click the **Properties** tab.
3. On the Properties tab for the db2udb resource, a list of Type Specific Attributes is shown. click the **Edit** icon for each attribute you want to configure. In the Edit Attribute window, enter the necessary attribute value information. For example, enter the db2inst1 as the value for DB2InstOwner.

4 Assign values for the IP and the NIC resources in the same manner as you assigned values to the db2udb resource: double-click the type to display the resource and select the resource. With the Properties tab visible, you can edit the Type Specific Attributes for each resource.

5 Right-click the db2mpp_grp1 service group in the left pane. Click Link in the drop-down menu.

The Link Service Groups window shows the following:

- The Parent Group as db2mpp_grp1
- The Child group as CVM
- The Relationship as "online local"
- The Dependency Type as "firm"

6 Click **OK** to create the dependency link.

To finalize the service group for the DB2 MPP database

1 Click the **Save Configuration** icon.

2 Enable the db2udb and IP resources. Right-click a resource and select **Enabled** in the drop-down menu. If necessary, make the configuration read and write.

3 Click the **Online Service Group** icon.

4 In the window, select the service group and the system that you want to bring online. Click **OK**.

Adding a service group for DB2 non-MPP with the Cluster Manager (Java Console)

If you have imported the Db2udbTypes.cf file, you can use the Db2udb_Group template to configure a service group.

See “Importing the Db2udbTypes.cf file” on page 38.

To configure a service group if you have imported the Db2udbTypes.cf file

1 In the Cluster Explorer window, answer **No** when prompted to use the configuration wizard. Note that if you choose to use the wizard, the steps that follow are similar.

2 In the Cluster Explorer window, click the **Add Service Group** icon on the toolbar.

3 In the Add Service Group window, enter the name of the service group you want to create for the service group. For example, enter db2_group1. Do not press Return or Enter.

4 From the systems in the Available Systems box, double-click those that you want in your configuration.

5 Select the system where you want to automatically start the service group. In the window that shows the systems in the configuration, select the check box next to the system.

6 Click the **Failover** radio button to specify the Service Group Type.

To configure templates if you have imported the Db2udbTypes.cf file

1 Click the **Templates** button.

2 In the Select Templates window, select db2udb_grp from the list in the Templates box. The Dependency graph information and the Types information should change to reflect the template choice. Click **OK**. In the Add Service Group window, the name of the template is now shown as selected.

To finalize the service group if you have imported the Db2udbTypes.cf file

1 Click **OK** on at the bottom of the Add Service Group window. The group is added. On the left pane of the Cluster Manager (Java Console) window, the service group is shown under the cluster name. On the Status tab, the group is shown Offline on each system.

2 In the left pane, double-click the **db2_group1** service group. The types of resources that you can configure for the group appear: Db2udb, DiskGroup, IP, Mount, NIC, and Volume.

3 Double-click the **Db2udb** resource type. Select the resource **db2udb**, which is under the Db2udb type. Click the **Properties** tab.

4 On the Properties tab for the db2udb resource, a list of Type Specific Attributes is shown. Click the **Edit** icon for each attribute you want to configure. In the Edit Attribute window, enter the necessary attribute value information. For example, enter the db2inst1 as the value for DB2InstOwner.

DB2InstOwner and DB2InstHome are required attributes. You must edit these attributes.

5 Assign values for the DiskGroup, IP, Mount, NIC, and Volume resources in the same manner as you assigned values to the db2udb resource: double-click the type to display the resource and select the resource. With the **Properties** tab visible, you can edit the Type Specific Attributes.

For a list of the required attributes and their descriptions, see the *Cluster Server Bundled Agents Reference Guide*.

6 Enable the resources in db2_group1. Right-click each resource and select **Enabled** in the drop-down menu. If necessary, make the configuration read and write.

7 Click the **Save and Close Configuration** icon.

8 Click the **Online Service Group** icon.

9 In the window, select the service group and the system that you want to bring online. Click the system where you want to bring it online. Click **OK**. Click **Yes** at the confirmation question.

Note: VCS Single Cluster Manager (Java Console) is no longer packaged with VCS. Veritas recommends use of Veritas InfoScale Operations Manager to manage, monitor and report on multi-cluster environments. You can download VIOM at https://sort.veritas.com/vom .

Configuring the VCS Agent for DB2 by editing the main.cf file

The VCS agent for DB2 comes with three sample VCS configuration files, which are in the /etc/VRTSagents/ha/conf/Db2udb/sample_db2udb directory.

The samples are as follows:

- A single-partition instance configuration
- A multi-partition instance SMP configuration
- A multi-partition instance MPP configuration.

The appropriate file can be used as reference to directly modify your present main.cf configuration file. When you use this method, you must stop and restart VCS to implement the configuration.

After you complete these steps, go to one of the following sections that describe configuring the agent.

To prepare to edit the main.cf file

1 Log on to System A as root.

2 Save your existing configuration to prevent any changes while you modify the main.cf file:

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

3 Make sure to stop VCS while you edit main.cf. Use the `hastop` command to stop the VCS engine on all systems and leave the resources available:

```
# hastop -all -force
```

4 Make a backup copy of the main.cf file:

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.cf.orig
```

Configuring the agent to use the DB2 MPP configuration

Edit the main.cf file. Use /etc/VRTSagents/ha/conf/Db2udb/sample_db2udb/main.cf.MPP for reference. Notice that CVM service group is present in the configuration file.

To configure the agent to use the DB2 MPP configuration

1 Include the Db2udbTypes.cf file.

```
include "Db2udbTypes.cf"
```

2 Create service groups for the Db2udb resources. Refer to the sample configuration file. The example shows four DB2 MPP service groups and a CVM service group as follows:

- MPP main.cf configuration for Linux
  See "MPP main.cf configuration for VCS agent for DB2 on Linux" on page 69.

3 In the DB2 MPP service groups, include the definitions for the Db2udb, IP, and NIC resources. Assign values to the attributes for the resources to match the parameters of your configuration.

For more information review the following information:

- Db2udb resource type attributes
  See "About the Db2udb resource types and attribute definitions" on page 60.
- Sample configuration files.
- For information about IP and NIC resources refer to the *Cluster Server Bundled Agents Reference Guide*.

4 Assign the online local firm service group dependency of the db2udb service group for the cvm service group. For example:

```
requires group cvm online local firm
```

5 Immediately following the service group dependency, assign dependencies for the newly created resources. Refer to the appropriate sample configuration file. For the group `db2mpp_grp0`, enter:

```
db2udb0 requires Db2_IP0
Db2_IP0 requires Db2_NIC0
```

For more information on assigning dependencies refer to the *Cluster Server Administrator's Guide*.

6 Save and close the file.

Configuring the agent to use the DB2 non-MPP configurations

Edit the main.cf file. Use /etc/VRTSagents/ha/conf/Db2udb/sample_db2udb/main.cf.EE or /etc/VRTSagents/ha/conf/Db2udb/sample_db2/main.cf.EEE for reference.

To configure the agent to use the DB2 non-MPP configurations

1 Include the Db2udbTypes.cf file.

```
include "Db2udbTypes.cf"
```

2 Create a service group for the DB2 resources.

This example that shows a group named "db2_grp1" in which two partitions are defined.

3 Include all resources in the service groups. These resources include the Db2udb, DiskGroup, IP, Mount, NIC, and Volume resources. Assign values to the attributes for the resources to match the parameters of your configuration.

For more information:

- Db2udb resource type attributes
  See "About the Db2udb resource types and attribute definitions" on page 60.
- The sample configuration files

For information about the DiskGroup, IP, Mount, NIC, and Volume resources, refer to the *Cluster Server Bundled Agents Reference Guide*.

4 Assign dependencies for the newly created resources. For the group db2udb_grp1, enter:

```
db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1
```

And for group db2udb_grp3, enter:

```
db2udb3 requires db2udb_ip3
db2udb3 requires db2udb_mnt3
db2udb_ip3 requires db2udb_nic3
db2udb_mnt3 requires db2udb_vol3
db2udb_vol3 requires db2udb_dg3
```

For more information, refer to the appropriate sample configuration file.

For more information on assigning dependencies, refer to the *Cluster Server Administrator's Guide*.

5 Save and close the file.

Verifying the configuration for VCS agent for DB2

Check the configuration after you edit the main.cf.

To check the configuration

1 Verify the syntax of the file /etc/VRTSvcs/conf/config/main.cf:

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

2 Start the VCS engine on System A:

```
# hastart
```

3 Type the `hastatus` command:

```
# hastatus
```

4 When "LOCAL_BUILD" is listed in the message column, start VCS on System B:

```
# hastart
```

5 Verify that all DB2 service group resources are brought online on System A:

```
# hagrp -display
```

6 Take the service groups offline on System A and verify that all resources are stopped:

```
# hagrp -offline db2udb_grp1 -sys sysa
# hagrp -offline db2udb_grp3 -sys sysa
# hagrp -display
```

7 Bring the service groups online again on System A and verify that all resources are available:

```
# hagrp -online db2udb_grp1 -sys sysa
# hagrp -online db2udb_grp3 -sys sysa
# hagrp -display
```

8 Switch the DB2 service group to System B:

```
# hagrp -switch db2udb_grp1 -to sysb
# hagrp -switch db2udb_grp3 -to sysb
```

9 Verify that all DB2 service group resources are brought online on System B:

```
# hagrp -display
```

10 On all the systems, look at the following log files for any errors or status:

```
/var/VRTSvcs/log/engine_A.log
/var/VRTSvcs/log/Db2udb_A.log
```

Setting up in-depth monitoring of a DB2 instance

Use the VCS commands to dynamically reconfigure the Cluster Server Agent for DB2. The following description of configuration changes to include in-depth monitoring shows the use of VCS commands from the command line. For information on reconfiguring VCS from the command line refer to the *Cluster Server Administrator's Guide*.

Enabling in-depth monitoring of a DB2 instance

Shallow monitoring of a DB2 instance involves checking the exit status of the `db2gcf` command.

In contrast, in-depth monitoring provides a higher level of confidence in the availability of the instance or partition and its database. It makes additional queries to the database to verify whether the database is available.

Enabling in-depth monitoring from the command line for DB2

You can dynamically configure in-depth monitoring. Veritas recommends that you successfully run DB2 with the agent's default (shallow) monitoring before you start the in-depth monitoring. In the MPP configuration, make sure the database can be accessible locally by the database partition.

You need to have custom monitoring scripts. Refer to the following table for information on the Indepth Monitor attribute:

See "About the Db2udb resource types and attribute definitions" on page 60.

To start the in-depth monitor for a given instance

1. Make the VCS configuration writable:

   ```
   # haconf -makerw
   ```

2. Freeze the service group so VCS does not perform actions automatically based on an incomplete reconfiguration:

   ```
   # hagrp -freeze db2udb_group
   ```

3. Enable in-depth monitoring using the command:

   ```
   # hares -modify resource DatabaseName name

   # hares -modify resource IndepthMonitor 1
   ```

   For example:

   ```
   # hares -modify db2udb DatabaseName SAMPLE
   # hares -modify db2udb IndepthMonitor 1
   # haconf -dump -makero
   # hagrp -unfreeze db2udb_group
   ```

Enabling and disabling intelligent resource monitoring for Db2udb resource

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

To enable intelligent resource monitoring for Db2udb resource

1. Make the VCS configuration writable by running the command

   ```
   # haconf -makerw
   ```

2. To enable intelligent resource monitoring for all the resources of type Db2udb, run the following command:

   ```
   # hatype -modify Db2udb IMF -update Mode 2
   ```

3. Change the values of the MonitorFreq key and the RegisterRetryLimit key of the IMF attribute.

   See “About the Db2udb resource types and attribute definitions” on page 60.

4. Save the VCS configuration

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

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

   ```
   # /opt/VRTS/bin/amfconfig
   ```

   Configure the AMF driver if the command output returns that the AMF driver is not loaded or not configured.

6. Restart the Db2udb agent. Run the following commands on each node.

   ```
   # haagent -stop Db2udb -force -sys sys_name
   ```

   ```
   # haagent -start Db2udb -sys sys_name
   ```

To disable intelligent resource monitoring for Db2udb resource

1. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

2. To disable intelligent resource monitoring for all the resources of type Db2udb, run the following command:

   ```
   # hatype -modify Db2udb IMF -update Mode 0
   ```

3 To disable intelligent resource monitoring for a specific Db2udb 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: `haimfconfig` utility can also be used to enable or disable IMF. Please refer *VCS Admin Guide* for more information on `haimfconfig`.

Administering the AMF kernel driver

Review the following procedures to start or stop the AMF kernel driver.

To start the AMF kernel driver

1 Check that the value of the AMF_START variable is set to 1 in the following file:

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

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

For RHEL 7, SLES 12, and supported RHEL distributions:

```
systemctl start amf
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

```
/etc/init.d/amf start
```

To stop the AMF kernel driver

1 Check that the value of the AMF_STOP variable is set to 1 in the following file:

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

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

For RHEL 7, SLES 12, and supported RHEL distributions:

```
systemctl stop amf
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

```
/etc/init.d/amf stop
```

Handling DB2 error codes during in-depth monitoring

The agent for DB2 comes with enhanced ability to handle DB2 errors during in-depth monitoring. The agent classifies DB2 errors according to their severity and associates predefined actions with each error code.

You can create a custom error handling file, db2error.dat. The file lists the DB2 errors and the associated actions that you want the agent to take when it encounters an error.

The file stores information in the following format:

```
SQL_error_string:action_to_be_taken
```

For example:

```
SQL1034N: IGNORE
SQL1039N: WARN
SQL1234N: FAILOVER
```

Table 4-1 shows the available actions for in-depth monitoring.

Table 4-1 Available actions for in-depth monitoring

Action	Description
`IGNORE`	Ignores the error.
`UNKNOWN`	Marks the resource state as UNKNOWN and sends a notification if the Notifier resource is configured. For more information about VCS notification, refer to the *Cluster Server Administrator's Guide*. This action is typically associated with configuration errors.
`WARN`	Marks the resource state as ONLINE and sends a notification if the Notifier resource is configured. This action is typically associated with low-severity errors.
`FAILOVER` (Default)	Marks the resource state as OFFLINE. This faults the service group, which fails over to the next available system. This action is the agent's default behavior. If the DB2 error code that the agent encounters does not exist in the db2error.dat file, then the agent assumes this default behavior.

Table 4-1 Available actions for in-depth monitoring *(continued)*

Action	Description
NOFAILOVER	Freezes the service group temporarily and marks the resource state as OFFLINE. The agent also sends a notification if the Notifier resource is configured. This action is typically associated with the errors that are not system-specific. For example a failover to another node does not help a corrupt database, .

Disabling in-depth monitoring for DB2

You can dynamically disable in-depth monitoring.

To dynamically disable in-depth monitoring

1. Make the VCS configuration writable:

```
# haconf -makerw
```

2. Freeze the service group so VCS does not perform actions automatically based on an incomplete reconfiguration:

```
# hagrp -freeze db2udb_group
```

3. Assign the InDepthMonitor attribute a null value to disable in-depth monitoring. Use the command:

```
# hares -modify resource IndepthMonitor 0
```

For example:

```
# hares -modify db2udb IndepthMonitor 0
# haconf -dump -makero
# hagrp -unfreeze db2udb_group
```

Chapter 5

Administering VCS service groups for DB2

This chapter includes the following topics:

- About administering service groups for DB2
- Bringing the DB2 service group online
- Taking the DB2 service group offline
- Switching the DB2 service group
- Disabling the agent for DB2

About administering service groups for DB2

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 DB2 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 DB2 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 DB2 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.

Disabling the agent for DB2

To disable the agent on a system, you must first change the DB2 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 Db2udb -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*.

Appendix A

Troubleshooting Cluster Server Agent for DB2

This appendix includes the following topics:

- Creating a db2profile for environment variables for DB2
- Setting the RestartLimit attribute for VCS agent for DB2
- Partition/Node switch-over scenarios for DB2 V9.5, V9.7, V10.1, and V10.5 in MPP mode

Creating a db2profile for environment variables for DB2

You can create a profile file for each instance of DB2 and place environment variables in the profile file. You can use this profile to create unique variables for each database user. Each DB2 instance has a home directory that is associated with the instance's log on ID, for example:

`$InstHome/sqllib/db2profile`

Where $InstHome is the name of the logon ID.

Place the variables that you are interested in using (for example the TimeZone variable) in the profile file. When you then issue a `su - db2instX` command (where X is the instance's name), the environment variables are sourced.

Setting the RestartLimit attribute for VCS agent for DB2

VCS starts multiple partitions simultaneously, which can lead to a race condition. The agent's RestartLimit attribute is set to a value of three to help avoid this condition. You can alleviate the potential for this condition by building resource dependencies for each partition. For example, within a service group you can have the Db2udb resource 4 (where nodenum=1) depend on Db2udb resource 3 (where nodenum=2) etc. With the partitions built in a dependency tree, you can set the value of the RestartLimit to zero.

Partition/Node switch-over scenarios for DB2 V9.5, V9.7, V10.1, and V10.5 in MPP mode

To switch-over when the partitions are online on a system

1. Serially move any partition from the source machine that is not associated with port 0 on the target.
2. Move the remaining partitions one by one with partition associated with port 0 on source being last.

For more information, refer IBM's case no: IC66748.

Appendix B

Resource type information for Cluster Server Agent for DB2

This appendix includes the following topics:

- About the resource type definitions for DB2

About the resource type definitions for DB2

The resource type represents the VCS configuration definition of the agent. It also specifies how the configuration file, main.cf, defines the agent. When you configure the agent, you need to assign values to the resource type attributes.

See "Sample configuration files for VCS agent for DB2 on Linux" on page 66.

DB2 UDB type definition file: Db2udbTypes.cf on Linux

The following is the type definition for the agent.

```
Type Db2udb (
        static keylist SupportedActions = { VRTS_GetInstanceName,
        VRTS_GetRunningServices }
        static int CleanTimeout = 240
        static int OfflineTimeout = 240
        static int MonitorTimeout = 240
        static int OnlineRetryLimit = 2
        static int OnlineTimeout = 180
        static int OnlineWaitLimit = 1
        static int RestartLimit = 3
```

```
        static int ToleranceLimit = 1
        static str AgentFile = "/opt/VRTSagents/ha/bin/Db2udb/Db2udbAgent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/Db2udb"
        static str ArgList[] = { DB2InstOwner, DB2InstHome, IndepthMonitor,
        DatabaseName, NodeNumber, StartUpOpt, ShutDownOpt, AgentDebug,
        Encoding,WarnOnlyIfDBQueryFailed, LastWarningDay, UseDB2start }
        static str IMFRegList[] = { DB2InstOwner, DB2InstHome }
        static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
        str DB2InstOwner
        str DB2InstHome
        int IndepthMonitor
        str DatabaseName
        int NodeNumber
        str StartUpOpt = START
        str ShutDownOpt = STOP
        boolean AgentDebug = 0
        str Encoding
        boolean WarnOnlyIfDBQueryFailed = 1
        temp str LastWarningDay
        boolean UseDB2start = 0
)
```

Appendix C

Resource type attributes for DB2

This appendix includes the following topics:

- About the Db2udb resource types and attribute definitions

About the Db2udb resource types and attribute definitions

Review the description of the DB2 agent attributes. The agent attributes are classified as required, optional, and internal.

Table C-1 shows the required attributes for the agent for DB2, you must assign values to required attributes.

Table C-1 Required attributes for the agent for DB2

Required attributes	Description
DB2InstHome	Path to DB2 UDB instance home directory that contains critical data and configuration files for the DB2 instance. Type and dimension: string-scalar
DB2InstOwner	User ID of Instance Owner that starts a DB2 UDB instance. Each instance requires a unique user ID. Type and dimension: string-scalar **Note:** Incorrect changes to this attribute can result in DB2 entering an inconsistent state.

Table C-2 shows the optional attributes for the agent for DB2.

Table C-2 Optional attributes for the agent for DB2

Optional attributes	Description
DatabaseName	Name of the database for in-depth monitoring; required if in-depth monitor is enabled (IndepthMonitor = 1). Be careful when you change the DataBase name attribute as you can fault all the partitions in the database. Do not change the DataBaseName attribute to an invalid or an incorrect value. **Note:** Make sure the database with the provided DatabaseName has been created in the same partition for which the Db2udb agent resource has been configured. Type and dimension: string-scalar
NodeNumber	Node number or partition number of the database. Used when monitoring a specific database partition. Default: 0 Type and dimension: integer-scalar

Table C-2 Optional attributes for the agent for DB2 *(continued)*

Optional attributes	Description
StartUpOpt	Provides start up options. The allowed values are: START, ACTIVATEDB, or CUSTOM. Review the following options: ■ START (default) Starts the DB2 instance or partition. ■ ACTIVATEDB Performs activate database command after db2 processes start. ■ CUSTOM The agent leaves all the online function completely to the user when the StartUpOpt attribute is set to CUSTOM. It looks for a file named start_custom_$db2instance_$nodenum in the /opt/VRTSagents/ha/bin/Db2udb directory. If this file exists and is executable, it executes this customized online file instead. Example: To customize the online function for partition/nodenum 1 for the db2 instance named db2inst1, the agent for DB2 runs this customized file start_custom_db2inst1_1. It runs this file under the /opt/VRTSagents/ha/bin/Db2udb directory. Type and dimension: string-scalar

Table C-2 Optional attributes for the agent for DB2 *(continued)*

Optional attributes	Description
ShutDownOpt	The allowed values for this attribute are STOP and CUSTOM. Review the following options: ■ STOP (Default) Shuts the Db2 instance or partition down in the usual way. ■ CUSTOM Leaves all the offline function completely to the user when the ShutDownOpt is set to CUSTOM. It looks for a file named stop_custom_$db2instance_$nodenum in the /opt/VRTSagents/ha/bin/Db2udb directory. If this file exists and is executable, it executes this customized offline file instead. Example: You want to customize the offline function for partition/nodenum 0 for the db2 instance named db2inst1. You have the agent for DB2 run this customized file: stop_custom_db2inst1_0. The file is in the /opt/VRTSagents/ha/bin/Db2udb directory. Type and dimension: string-scalar
IndepthMonitor	Set the value of the IndepthMonitor attribute to 1 to enable in-depth monitoring. The agent now looks for the monitor_custom_$db2instance_$nodenum file in the /opt/VRTSagents/ha/bin/Db2udb directory. It executes this customized indepth monitor file if the file exists and is executable. You can find samples of custom monitor scripts in the sample_db2udb directory. Type and dimension: string-integer
Encoding	Specifies the operating system encoding corresponding to DB2 UDB encoding for display of DB2 UDB output. Type and dimension: string-scalar
AgentDebug	When the value of this attribute is 1, it causes the agent to log additional debug messages. Type and dimension: boolean-scalar

Table C-2 Optional attributes for the agent for DB2 *(continued)*

Optional attributes	Description
WarnOnlyIfDBQueryFailed	This attribute either logs SQL errors, or checks the errors to handle them specially. Set the value of the WarnOnlyIfDBQueryFailed attribute to 1 to enable it. When this attribute is enabled, it ignores all SQL errors and logs a warning message in the agent log once a day. Set the value of the WarnOnlyIfDBQueryFailed attribute to 0 to disable it. When disabled, it checks if an error code needs to be handled specially in the db2error.dat file. If the error code does not exist in the db2error.dat file, then it returns OFFLINE for monitor. Otherwise, it follows the action of that particular error code in the db2error.dat file. Type and dimension: boolean-scalar
UseDB2start	Enables you to choose an alternate way to start DB manager. The agent executes the db2start when UseDBStart is set to 1, whereas it uses db2gcf when UseDBStart is set to 0. If the setup is non-MPP, you must set the value of the UseDB2start attribute to 1. If the setup is MPP, you must set the value of the UseDB2start attribute to 0. For MPP, VCS uses the actual hostname and needs the ability to start and stop each partition. For this purpose, VCS uses the db2gcf command and requires a value of 0 for the UseDB2start attribute. Type and dimension: boolean-scalar Default: 0 For more information on using UseDB2start attribute refer : See "About DB2 configurations in VCS" on page 34.

Table C-3 shows the internal attributes for the agent for DB2.

Table C-3 Internal attribute for the agent for DB2

Internal attributes	Description
AgentDirectory	Specifies the location of other files and scripts that are related to the agent. Do not use. For internal use only.

Table C-3 Internal attribute for the agent for DB2 *(continued)*

Internal attributes	Description
IMF	This resource-type level attribute determines whether the DB2 agent must perform intelligent resource monitoring. 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: 2 ■ 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 imf_register agent function to register the resource with theAMF kernel 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 for Db2udb resource" on page 49.

Appendix D

Sample configurations

This appendix includes the following topics:

- Sample configuration files for VCS agent for DB2 on Linux

Sample configuration files for VCS agent for DB2 on Linux

This section shows example DB2 UDB configurations.

Non-MPP main.cf configuration for VCS agent for DB2 on Linux

The following main.cf configuration file reflects DB2 UDB in an ESE multi-partition instance SMP environment.

```
include "types.cf"
include "Db2udbTypes.cf"

cluster vcs (
    CounterInterval = 5
    )

system vcstc1 (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
    ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,
    NotifyTimeLimit = 0 }
    )

system vcstc2 (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
    ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,
```

```
    NotifyTimeLimit = 0 }
    )

group db2udb_grp1 (
    SystemList = { vcstc1= 0, vcstc2 = 1 }
    AutoStartList = { vcstc1 }
    )

    Db2udb db2udb1 (
        DB2InstOwner = db2inst1
        DB2InstHome = "/db2inst1"
        IndepthMonitor = 1
        DatabaseName = SAMPLE
        NodeNumber = 0
        )

    Db2udb db2udb2 (
        DB2InstOwner = db2inst1
        DB2InstHome = "/db2inst1"
        IndepthMonitor = 0
        NodeNumber = 1
        )

    DiskGroup db2udb_dg1 (
        DiskGroup = db2_dg1
        )

    IP db2udb_ip1 (
        Device = eth0
        Address = "166.98.9.163"
        NetMask = "255.255.252.0"
        )

    Mount db2udb_mnt1 (
        MountPoint = "/db2inst1"
        BlockDevice = "/dev/vx/dsk/db2_dg1/inst1_vol"
        FSType = vxfs
        MountOpt = rw
        FsckOpt = "-n"
        )

    NIC db2udb_nic1 (
        Device = eth0
```

```
    )

Volume db2udb_vol1 (
    Volume = inst1_vol
    DiskGroup = db2_dg1
    )

db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb2 requires db2udb_ip1
db2udb2 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1

// resource dependency tree
//
//          group db2udb_grp1
//          {
//          Db2udb db2udb1
//                {
//                IP db2udb_ip1
//                      {
//                      NIC db2udb_nic1
//                      }
//                Mount db2udb_mnt1
//                      {
//                      Volume db2udb_vol1
//                            {
//                            DiskGroup db2udb_dg1
//                            }
//                      }
//                }
//          Db2udb db2udb2
//                {
//                IP db2udb_ip1
//                      {
//                      NIC db2udb_nic1
//                      }
//                Mount db2udb_mnt1
//                      {
//                      Volume db2udb_vol1
//                            {
```

```
    //                              DiskGroup db2udb_dg1
    //                              }
    //                         }
    //                    }
    //               }
```

MPP main.cf configuration for VCS agent for DB2 on Linux

The following configuration file reflects DB2 UDB in an ESE multi-partition instance MPP environment. Four database partitions are shown. One partition is configured on each cluster node. Each database service group depends on the same CVM service group, which manages the shared storage in the cluster.

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"

cluster db2_mpp (
    CounterInterval = 5
    )

system vcstc1 (
    )

system vcstc2 (
    )

system vcstc3 (
    )

system vcstc4 (
    )

group cvm (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { vcstc1, vcstc2, vcstc3, vcstc4 }
    )

    CFSMount db2cfsmnt (
        MountPoint = "/db2_mnt/db2inst1"
```

```
    BlockDevice = "/dev/vx/dsk/cdb2dg1/cdb2dg1home"
    )

CFSfsckd vxfsckd (
    )

CVMCluster cvm_clus (
    CVMClustName = db2_mpp
    CVMNodeId = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3
    }
    CVMTransport = gab
    CVMTimeout = 200
    )

CVMVolDg db2dg (
    CVMDiskGroup = cdb2dg1
    CVMVolume = { cdb2dg1home }
    CVMActivation = sw
    )

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

cvm_clus requires cvm_vxconfigd
db2cfsmnt requires db2dg

db2cfsmnt requires vxfsckd
db2dg requires cvm_clus
vxfsckd requires cvm_clus

// resource dependency tree
//
//          group cvm
//          {
//          CFSMount db2cfsmnt
//                {
//                CVMVolDg db2dg
//                      {
//                      CVMCluster cvm_clus
//                            {
//                            CVMVxconfigd cvm_vxconfigd
```

```
    //                              }
    //                        }
    //                  CFSfsckd vxfsckd
    //                        {
    //                        CVMCluster cvm_clus
    //                              {
    //                              CVMVxconfigd cvm_vxconfigd
    //                              }
    //                        }
    //                  }
    //          }

group db2mpp_grp0 (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoStartList = { vcstc1 }
    AutoStart = 1
    )

    Db2udb db2udb0 (
        DB2InstOwner = db2inst1
        DB2InstHome = "/db2_mnt/db2inst1"
        IndepthMonitor = 1
        DatabaseName = SAMPLE
        NodeNumber = 0
        )

    IP Db2_IP0 (
        Device = eth0
        Address = "10.118.2.144"
        NetMask = "255.255.248.0"
        )

    NIC Db2_NIC0 (
        Device = eth0
        )

    requires group cvm online local firm
    Db2_IP0 requires Db2_NIC0
    db2udb0 requires Db2_IP0

    // resource dependency tree
    //
    //          group db2mpp_grp0
```

```
    //          {
    //          Db2udb db2udb0
    //                {
    //                IP Db2_IP0
    //                      {
    //                      NIC Db2_NIC0
    //                      }
    //                }
    //          }

group db2mpp_grp1 (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoStartList = { vcstc2 }
    AutoStart = 1
    )
    Db2udb db2udb1 (
         DB2InstOwner = db2inst1
         DB2InstHome = "/db2_mnt/db2inst1"
         IndepthMonitor = 1
         DatabaseName = TEST1
         NodeNumber = 1
         )

    IP Db2_IP1 (
         Device = eth0
         Address = "10.118.2.145"
         NetMask = "255.255.248.0"
         )

    NIC Db2_NIC1 (
         Device = eth0
         )

    requires group cvm online local firm
    Db2_IP1 requires Db2_NIC1
    db2udb1 requires Db2_IP1

    // resource dependency tree
    //
    //          group db2mpp_grp1
    //          {
    //          Db2udb db2udb1
    //                {
```

```
    //                IP Db2_IP1
    //                      {
    //                      NIC Db2_NIC1
    //                      }
    //                }
    //          }

group db2mpp_grp2 (
     SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
     AutoStartList = { vcstc3 }
     AutoStart = 1
     )

     Db2udb db2udb2 (
         DB2InstOwner = db2inst1
         DB2InstHome = "/db2_mnt/db2inst1"
         IndepthMonitor = 1
         DatabaseName = TEST2
         NodeNumber = 2
         )

     IP Db2_IP2 (
         Device = eth0
         Address = "10.118.2.146"
         NetMask = "255.255.248.0"
         )

     NIC Db2_NIC2 (
         Device = eth0
         )

     requires group cvm online local firm
     Db2_IP2 requires Db2_NIC2
     db2udb2 requires Db2_IP2

     // resource dependency tree
     //
     //          group db2mpp_grp2
     //          {
     //          Db2udb db2udb2
     //                {
     //                IP Db2_IP2
     //                      {
```

```
    //                      NIC Db2_NIC2
    //                      }
    //                }
    //          }

group db2mpp_grp3 (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoStartList = { vcstc4 }
    AutoStart = 1
    )

    Db2udb db2udb3 (

        DB2InstOwner = db2inst1
        DB2InstHome = "/db2_mnt/db2inst1"
        NodeNumber = 3
        )
    IP Db2_IP3 (
        Device = eth0
        Address = "10.118.2.147"
        NetMask = "255.255.248.0"
        )

    NIC Db2_NIC3 (
        Device = eth0
        )

    requires group cvm online local firm
    Db2_IP3 requires Db2_NIC3
    db2udb3 requires Db2_IP3

    // resource dependency tree
    //
    //          group db2mpp_grp3
    //          {
    //          Db2udb db2udb3
    //                {
    //                IP Db2_IP3
    //                      {
    //                      NIC Db2_NIC3
    //                      }
    //                }
    //          }
```

Index

Symbols

/etc/services
    checking assigned ports 22

A

agent
    configuring
        overview 33
attributes of Db2udb resource type 60

B

binaries
    DB2 UDB
        installing locally 21

C

commands
    hagrp 54
configuration files
    db2nodes.cfg
        host for partitions 23
    Db2udbTypes.cf 58
    main.cf
        example 66, 69
configuring DB2 agent
    by editing configuration files 43
CVM (infrastructure) service group 37

D

DatabaseName attribute 61
DB2
    error handling 51
DB2 UDB ESE (Enterprise Server Edition)
    sample configuration file 66, 69
DB2 UDB instances
    installing on shared storage 21
DB2 UDB type definition file 58
db2gcf -d command 9
db2gcf -u command 9
DB2InstHome attribute 60
DB2InstOwner attribute 60
db2nodes.cfg
    virtual host for partitions 23
db2setup tool
    for binaries
        instances 21
Db2udb resource
    depends on IP
        Mount resources 36
Db2udbTypes.cf
    including in main.cf file 44–45
    type definition file 58
dependency
    defining in main.cf 46
DiskGroup resource
    required by Volume resource 36

E

Encoding attribute 63
enterprise agent
    disabling 31
    installing 29

F

file system
    creating for MPP configuration 18
    creating for non-MPP instances 18
files
    db2nodes.cfg 23
    Db2udbTypes.cf 58
    main.cf (VCS configuration) 43

G

groups (user)
    adding 19

H

hagrp command 54

I

IMF Integration functions 12
in-depth monitoring
    disabling 52
    enabling 48
installing DB2 UDB
    confirming setup 26
    instances on shared storage 21
    requirements 21
    system binaries 21
instances
    installing on shared disk 21
    starting and stopping 26
IP address
    virtual 23
IP resource
    requires a NIC resource 36

M

main.cf file
    editing 43
    example for DB2 UDB EEE 66
    example for DB2 UDB ESE multi-partition 69
    examples 43
    verifying configuration 46
monitoring
    configuring in-depth monitoring 48
    shallow and indepth 48
Mount resource
    requires the Volume resource 36

N

NIC resource
    required by IP resource 36
NodeNumber attribute 61

P

parallel mode for CVM service group 37
partition number (NodeNumber) 61

R

removing 32
ResourceInfo resource attribute 13

S

service group
    creating in main.cf file 44–45
service group *(continued)*
    CVM 37
    MPP configuration
        dependencies 36
    non-MPP configuration
        dependencies 35
shallow monitoring
    described 48
stopping DB2 instance 9
supported software 9
Supporting intelligent resource monitoring
    IMF 8

T

template
    using for non-MPP service group 43
testing
    service groups are configured 47

U

uninstalling 32

V

VCS supported version 9
virtual IP address 23
Volume resource
    requires the DiskGroup resource 36

W

WarnOnlyIfDBQueryFailed attribute 10