Veritas™ Cluster Server Agent for DB2 Installation and Configuration Guide

Solaris

5.0 Maintenance Pack 3

symantec™

Veritas Cluster Server Agent for DB2 Installation and Configuration Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 5.0 MP3

Document version: 5.0MP3.0

Legal Notice

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's maintenance offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

For information about Symantec's Maintenance Programs, you can visit our Web site at the following URL:

www.symantec.com/techsupp/

Contacting Technical Support

Customers with a current maintenance agreement may access Technical Support information at the following URL:

http://www.symantec.com/business/support/assistance_care.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system

- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and maintenance contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Documentation feedback

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions to clustering_docs@symantec.com. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting.

Maintenance agreement resources

If you want to contact Symantec regarding an existing maintenance agreement, please contact the maintenance agreement administration team for your region as follows:

Asia-Pacific and Japan	contractsadmin@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Introducing the Veritas agent for DB2

This chapter includes the following topics:

- About the Veritas High Availability Agent for DB2

- How the agent makes DB2 highly available

- What's new in this release

- Supported software for DB2

- DB2 agent functions

- Typical DB2 configuration in a VCS cluster

- Road map for setting up a DB2 UDB cluster

About the Veritas High Availability Agent for DB2

The Veritas High Availability agent, version 5.0 MP3, 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 the agent makes DB2 highly available

The agent for DB2 monitors DB2 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. For more information on configuring the different functions that the agent has:

See "DB2 agent functions" on page 12.

What's new in this release

This release introduces support for DB2 9.5.

Supported software for DB2

The Veritas High Availability agent, version 5.0 MP3 for DB2 UDB, supports the DB2 Universal Database Enterprise Server Edition (ESE).

For the ESE multi-partition instance, it supports both of the following configurations:

- Symmetric Multiprocessing (SMP) hardware configuration
- Massively Parallel Processing (MPP) hardware configuration

The agent for DB2 ESE must support the following platforms or single and multi-partition instances.

See the *Veritas Cluster Server Release Notes* for information on supported operating systems and versions.

The memory requirements vary for different versions of DB2 being used. Check the relevant IBM DB2 guide for information about memory requirements.

DB2 agent functions

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 *Veritas Cluster Server Agent Developer's Guide.*

Online agent function

The agent uses 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
```

Offline agent function

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
```

Monitor agent function

The agent executes the 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.

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/sample_db2udb directory.

Note: When you run the DB2 instance inside of a Solaris 10 non-global zone, create the monitor_custom_$db2instance_$nodenum file inside the non-global zone. Create the file after doing a "zlogin *zonename*" command to log on to the zone. You then change to the /etc/VRTSagents/ha/bin/Db2udb directory, which is relative to the local zone.

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. For error handling information:

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.

Clean agent function

The agent uses the db2gcf program to kill a DB2 database partition. The command is:

```
su $DB2InstOwner -c "$InstHome/sqllib/bin/db2gcf -k -i
    $DB2InstOwner -p $nodenum"
```

Info agent functions

The agent for DB2 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:

- *Veritas Cluster Server User's Guide*
- *Veritas Cluster Server Agent Developer's Guide*

For an example of the info agent function retrieving database information:

Action agent function

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] [-rclus 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 (db2udb0): 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
```

Running the info agent function

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 processes' information 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.

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 2005
```

Typical DB2 configuration in a VCS cluster

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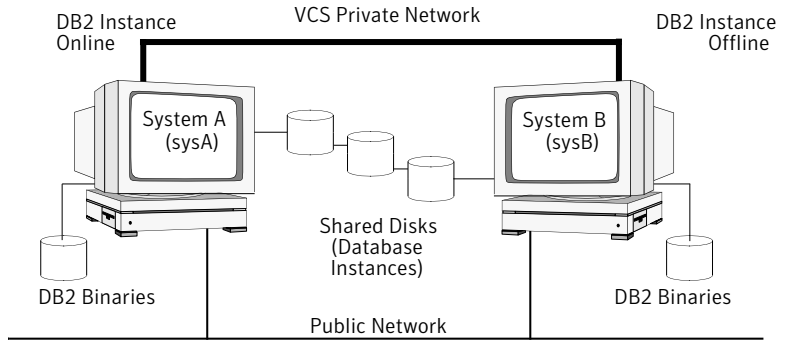
In the case of the MPP configuration, a database partition can run on each system and each system can become a failover system.

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

Figure 1-2   DB2 installation with an MPP configuration
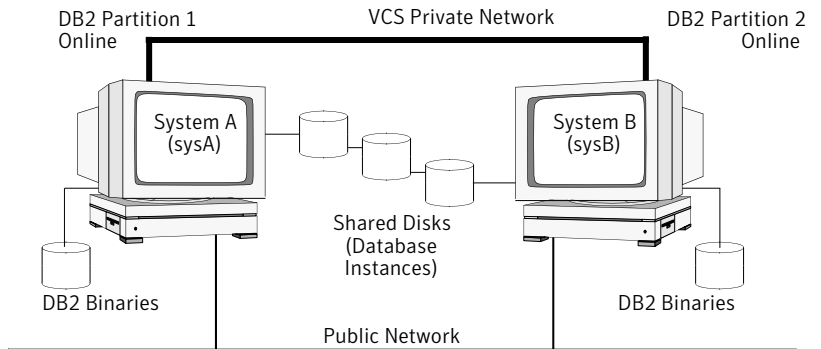
Road map for setting up a DB2 UDB cluster

Table 1-1 is a road map to set up the cluster.

Table 1-1   Setting up a DB2 UDB cluster

Task	Reference
Review the following: ■ The supported software ■ The agent functions ■ The agent requirements	■ See "Supported software for DB2" on page 12. ■ See "DB2 agent functions" on page 12. ■ See "VCS requirements for installing DB2" on page 19.
Install and set up DB2.	See "Installing DB2 UDB in a VCS environment" on page 24. See "Setting up the DB2 UDB configuration" on page 25.
Install the VCS agent for DB2.	See "Installing the agent for DB2" on page 30.
Configure the service groups for the VCS agent for DB2. Optionally, configure in-depth monitoring and any automated actions.	See "Configuring the DB2 UDB agent from Cluster Manager (Java Console)" on page 38. See "Setting up in-depth monitoring of DB2 UDB instance" on page 46.
Bring the service group online.	See "Bringing the service group online for DB2" on page 51.

Installing and configuring DB2

This chapter includes the following topics:

VCS requirements for installing DB2

Review the following requirements and prerequisites for installing DB2.

Prerequisites for installing DB2 UDB, non-MPP versions

Perform the following prerequisites before you install the non-MPP versions of DB2:

- Verify that all systems have enough resources to run DB2 UDB. An example of this kind of resource is shared memory. Check the DB2 memory requirements, which vary due to the version and the hardware configuration of DB2.

- Install the DB2 UDB system binaries locally.

- Install the DB2 UDB database instances on shared storage.

- Install and configure VCS version 5.0 MP3 on all nodes in the cluster. For installation instructions:
  See the *Veritas Cluster Server Installation Guide*.

- Before installing DB2 UDB, define DB2 UDB user and group accounts as follows:

- Defining DB2 user and group accounts
  See "Defining DB2 user and group accounts" on page 23.

- For your particular version of DB2, refer to the appropriate DB2 UDB guide.

Prerequisites for installing DB2 UDB, MPP version

Perform the following prerequisites before you installing the MPP versions of DB2:

- Verify that all systems have enough resources to run DB2 UDB. An example of this kind of resource is shared memory. Check the DB2 memory requirements, which vary due to the version and the hardware configuration of DB2.

- Install the DB2 UDB system binaries on the local file systems on each system.

- Install the DB2 UDB database instances on shared storage.

- The MPP configuration requires the Storage Foundation Cluster File System software. This software includes the following:

  - Veritas Cluster Server (VCS)

  - Veritas Volume Manager with cluster functionality enabled (CVM)

  - Veritas File System with cluster functionality enabled (CFS)

  For information and installation instructions for these products refer to the following guides, see the:

  - *Veritas Storage Foundation Cluster File System Installation Guide.*

  - *Veritas Storage Foundation Cluster File System Administration Guide.*

- Before installing DB2 UDB, define DB2 UDB user and group accounts and review the following:

  - Defining DB2 user and group accounts
    See "Defining DB2 user and group accounts" on page 23.

  - The relevant DB2 UDB guide

Creating file systems for DB2 instances

The following sections describe examples of creating disk groups for the DB2 database instances.

Creating the file system for the DB2 non-MPP instances

To create a file system, 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 on Solaris systems for non-MPP instances

1 Create a disk group on the shared disk. List the disks using the `vxdisk list` command. In this case the group consists of one disk, c4t0d0s2. For example:

```
# vxdg init db2db_dg c4t0d0s2
```

Deport and import the disk group:

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

2 Create a volume of three GB using the `vxassist` command:

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

3 Create the file system:

```
# mkfs -F vxfs /dev/vx/rdsk/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 -F vxfs /dev/vx/dsk/db2db_dg/db2db_vol \
  /db2_mnt/db2inst1
```

Creating the shared cluster file system for the DB2 MPP instances

To create a shared file system, you first create a shared disk group on the physically shared disk. You then create a volume of sufficient size within the disk group. You must have installed the Storage Foundation Cluster File System software.

To create a shared file system on Solaris for MPP instances

1   You must issue the commands to create a shared disk group from the CVM master node. To determine whether a node is the master or the slave, enter the command:

    ```
    # vxdctl -c mode
    ```

    In the output, look for:

    ```
    cluster active - MASTER
    ```

    Or

    ```
    cluster active - SLAVE
    ```

2   From the master node, create the disk group. List the disks using the `vxdisk list` command.

3   Create a shared disk group. In this case, the group consists of one disk. In this example the disk is c5t0d0s2:

    ```
    # vxdg -s init db2db_dg c5t0d0s2
    ```

4   Deport and import the disk group:

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

5   Use the `vxassist` command to create a 7-GB volume:

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

6   Create the file system:

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

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

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

Defining DB2 user and group accounts

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 systems.

■ The DB2 instance owner's home directory must exist locally on 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. Symantec 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

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

To create the group accounts on each node in the cluster for Solaris systems

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 db2asgrp
```

Adding user accounts for Solaris systems

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 file system that hosts the DB2 UDB instance home directory on shared storage uses this mount point. 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 db2as. 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 db2asgrp -u 1002 -d /home/db2as -m -s /bin/ksh db2as
```

Installing DB2 UDB in a VCS environment

For installing DB2 UDB in a VCS environment, Symantec 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.

Setting shared memory parameters

Refer to the relevant IBM DB2 UDB guide to make sure that memory requirements are met.

On Solaris systems, set the memory parameters in the /etc/system file.

Installing 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.

Install the DB2 license

Install the DB2 license on each node. For example, enter:

```
# /opt/IBM/db2/V8.1/adm/db2licm -a db2ese.lic
```

Installing 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.

Setting up the DB2 UDB configuration

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

Checking /etc/services

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 UDB 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.

Creating $DB2InstHome/.rhosts

On each system, create a file named $DB2InstHome/.rhosts, and place a "+" character within it. This file permits a system to access the database without the use of a password.

If security is a concern, put the hostname and user ID inside the .rhosts file, as shown in the following examples:

```
dbmach01    db2inst1
dbmach02    db2inst1
dbmach03    db2inst1
dbmach04    db2inst1
```

Or

```
+    db2inst1
```

With the `rsh` *system_name* command, you can test password free remote logon. From one system in the cluster to another, the command tests that you can remotely log on with the DB2 instance (for example, db2inst1) account. You should not be prompted for a password. Test this command from each system in the cluster to all other systems.

Modifying the $DB2InstHome/sqllib/db2nodes.cfg file

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

Non-MPP versions

For each DB2 UDB instance (non-MPP) database partition, modify the file $DB2InstHome/sqllib/db2nodes.cfg. You need to create an entry for each database partition, and to assign the virtual IP address as the hostname. For example:

```
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.

MPP versions

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". "The netname" is by default the hostname. The virtual IP is not used in the db2nodes.cfg file for MPP configurations.

For example:

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

Make sure that the relative port number in the third column is unique for each partition on a host. For example:

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

Confirming the setup of DB2 MPP and non-MPP installations

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 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.

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

```
$ db2stop
```

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

5 Create a database.

```
$ db2 create database dbname
```

6 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.

Installing, upgrading, and removing the agent for DB2 UDB

This chapter includes the following topics:

- Before you install or upgrade the agent

- Installing the agent for DB2

- Upgrading the agent for DB2 UDB

- Removing the agent for DB2 UDB

Before you install or upgrade the agent

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

- VCS requirements for installing DB2
  See "VCS requirements for installing DB2" on page 19.

- Installing DB2 UDB in a VCS environment
  See "Installing DB2 UDB in a VCS environment" on page 24.

- Make sure that VCS is installed in the cluster.

- Verify that DB2 is installed and configured.

- Symantec recommends installing the VCS graphical user interface, the Cluster Manager (Java Console).
  See the *Veritas Cluster Server Installation Guide*.

Installing the agent for DB2

Mount the disc and install the agent.

To install the agent

1   Log on as superuser.

2   Create a temporary directory for installation:

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

3   Insert the disc into a system drive. Note the following:

■   If you are running Solaris volume-management software, the software automatically mounts the disc as /cdrom/cdrom0. Type the following command to go to the location:

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

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

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

Where, in this example, /dev/dsk/c0t6d0s2 is the default for the CD drive. After the disc is mounted, type the following commands to go to the location:

```
# cd /cdrom
```

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

```
# cp -r db2_agent/pkgs/* /tmp/install
```

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

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

6   Go to the temporary directory and unzip the compressed package file:

```
# cd /tmp/install
# gunzip VRTS*.gz
```

7 Extract the compressed file from the tar file:

```
# tar -xvf VRTSvcsdb.tar
```

8 Install the package:

```
# pkgadd -d . VRTSvcsdb
```

To install the Japanese language pack on Solaris systems

1 After you have installed the agent, insert the language disc into the system's drive. Type the command:

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

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

```
# cp -r ja/cluster_server_agents/db2_agent/pkgs/* /tmp/install
```

3 Go to the temporary directory and unzip the compressed package file:

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

4 Extract the compressed file from the tar file:

```
# tar -xvf VRTSjacsb.tar
```

5 Install the Japanese package:

```
# pkgadd -d . VRTSjacsb
```

Upgrading the agent for DB2 UDB

You can only upgrade the HA agent for DB2 manually. The installvcs program does not automatically upgrade the VRTSvcsdb package.

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 4.1 or 5.0

1 Disable and remove the agent.

For more information see the following:

■ Disabling the agent for DB2
See "Disabling the agent for DB2" on page 52.

■ Removing the agent for DB2 UDB
See "Removing the agent for DB2 UDB" on page 32.

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

See "Installing the agent for DB2" on page 30.

3   Copy the new "Db2udbTypes.cf" from the /etc/VRTSagents/ha/conf directory
to the /etc/VRTSagents/ha/conf/config directory.

4   Update the location of the Db2udbTypes.cf file in your main.cf file's include
statement. For example, change this statement:

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

To read:

```
include "/etc/VRTSagents/ha/conf/config/DB2udbTypes.cf"
```

5   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 47.

Removing the agent for DB2 UDB

Before you remove the agent, you must disable it.

See "Disabling the agent for DB2" on page 52.

Perform the following steps to remove the agent:

To remove the agent from Solaris systems

◆   On each system that has the agent, type:

```
# pkgrm VRTSvcsdb
```

To remove the Japanese language pack from Solaris systems

◆   On each system that has the agent, type:

```
# pkgrm VRTSjacsb
```

Configuring VCS service groups for DB2

This chapter includes the following topics:

- About configuring service groups for DB2 UDB

- About DB2 UDB configurations in VCS

- Before configuring the service group for DB2

- Configuring the DB2 UDB agent from Cluster Manager (Java Console)

- Configuring the DB2 UDB agent by editing the main.cf file

- Setting up in-depth monitoring of DB2 UDB instance

About configuring service groups for DB2 UDB

This chapter describes how you can configure the DB2 UDB agent.

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 DB2 UDB agent.
  See "Configuring the DB2 UDB agent from Cluster Manager (Java Console)" on page 38.

- Use the VCS Management Console.
  You can use the VCS Management Console to configure service groups. For more information on the VCS Management Console, refer to the *Veritas Cluster Server User's Guide*.

- Use VCS commands.

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

  - Configuring the DB2 UDB agent by editing the main.cf file
    See "Configuring the DB2 UDB agent by editing the main.cf file" on page 42.

About DB2 UDB configurations in VCS

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

Solaris Zones are also available for use with service groups.

DB2 UDB service group for non-MPP configuration

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

Figure 4-1   Dependency tree for a DB2udb resource
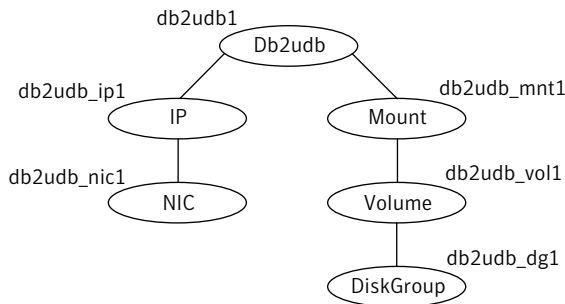
This configuration shows a service group for a Db2udb resource. The db2udb1 resource (the database) requires the IP resource and the Mount resource. The service group IP address for the DB2 UDB server is configured using the IP resource (db2udb_ip1) and the NIC resource (db2udb_nic1). The mount resource (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.

DB2 UDB service groups for MPP configuration

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

These service groups are as follows:

- Parallel CVM
- DB2 failover

Parallel CVM service group

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
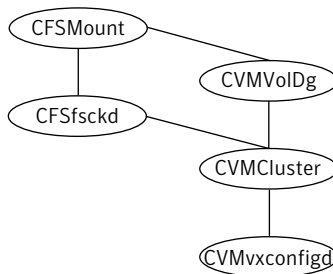
DB2 failover service group

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
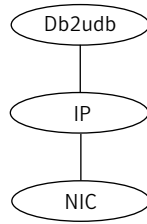
DB2 UDB instances that are configured in Solaris zones

The following examples describe a service group that monitors the state of a DB2 instance in a Solaris zone.

Figure 4-4 illustrates the dependencies among the resources that are configured for a Db2udb resource that is configured in a Solaris zone.

Figure 4-4 Db2udb resource that is configured in a Solaris zone
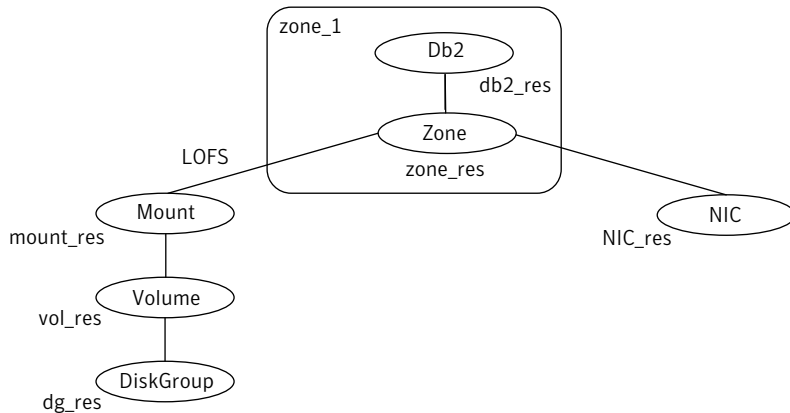
The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The Solaris Zone is monitored through a Zone resource, which is dependent on the Mount and NIC resources. The DB2 server can be started after each of these resources is brought online.

The DB2 instance's home directory is mounted in the global zone. To make this file system available to the non-global zone, you must execute the following command on the global zone.

The lines in the following example specify that you mount /zones/db2data in the global zone as /db2inst1 in the non-global zone named zone1. The file system

type to use is LOFS. The /db2inst1 directory in this example is the home directory for the DB2 instance.

```
# zonecfg -z zone1
zonecfg:zone1> add fs
zonecfg:zone1:fs> set dir=/db2inst1
zonecfg:zone1:fs> set type=lofs
zonecfg:zone1:fs> set special=/zones/db2data
zonecfg:zone1:fs> end
```

For more information on setting up VCS to work with Solaris 10 local zones, see the appendix in the:*Veritas Cluster Server User's Guide.*

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 *Veritas Cluster Server Installation Guide.*

- Verify that DB2 is installed and configured identically on all nodes in the cluster.
  See "Installing DB2 UDB in a VCS environment" on page 24.

- Verify that the Veritas agent for DB2 is installed on all nodes in the cluster.
  See "Installing the agent for DB2" on page 30.

Importing the dbtypesTypes.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.

To import the dbtypesTypes.cf file using the Cluster Manager

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 UDB MPP using the Cluster Manager (Java Console)" on page 38.

- If you use the DB2 UDB non-MPP configuration:
  See "Adding a service group for DB2 UDB non-MPP with the Cluster Manager (Java Console)" on page 40.

Configuring the DB2 UDB agent from Cluster Manager (Java Console)

Templates for the DB2 UDB 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 UDB 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.

For information on the Cluster Manager (Java Console), see the *Veritas Cluster Server User's Guide*.

Adding service group for DB2 UDB 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 dbtypesTypes.cf file" on page 37.

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 UDB 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 UDB 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 UDB 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 UDB 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 UDB 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.

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 *Veritas 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.

Configuring the DB2 UDB agent 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 for DB2.

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 UDB 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 UDB MPP configuration

1   Add the fully qualified path to the Db2udbTypes.cf file.

    ```
    include "/etc/VRTSagents/ha/conf/Db2udb/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.

    ■ MPP main.cf configuration for Solaris
      See "MPP main.cf configuration for Solaris" on page 65.

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 "Db2udb resource type attributes" on page 58.

    ■ Sample configuration files.

    ■ For information about IP and NIC resources:
      See the *Veritas 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:

    See the *Veritas Cluster Server User's Guide*.

6   Save and close the file.

Configuring the agent to use the DB2 UDB, 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 UDB, non-MPP configurations

1   Add the fully qualified path to the Db2udbTypes.cf file.

```
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"
```

2   Create a service group for the DB2 UDB 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 "Db2udb resource type attributes" on page 58.

■   The sample configuration files

For information about the DiskGroup, IP, Mount, NIC, and Volume resources:

See the *Veritas 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:

See the *Veritas Cluster Server User's Guide*.

5   Save and close the file.

Verifying the configuration

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 UDB 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 UDB service group to System B:

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

9 Verify that all DB2 UDB 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 DB2 UDB instance

To dynamically reconfigure the Veritas agent for DB2, use Cluster Manager (Java Console) or the VCS command line. 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 the *Veritas Cluster Server User's Guide*.

Enabling in-depth monitoring of DB2 UDB instance

Shallow monitoring of a DB2 UDB 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

You can dynamically configure in-depth monitoring. Symantec recommends that you successfully run DB2 UDB 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.

For locales other than English to avoid resource faults, add the following lines to the $INSTHOME/sqllib/userprofile file.

The following example adds Japanese language support:

```
LANG=ja
export LANG
```

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

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_grp1**

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_grp1**

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: See the *Veritas Cluster Server User'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.
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

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_grp1
    ```

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_grp1
    ```

Administering VCS service groups for DB2

This chapter includes the following topics:

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

About administering service groups for DB2

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

Bringing the service group online for DB2

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 service group offline for DB2

Perform the following procedure from Cluster Manager (Java Console) to take the service group offline. Note that in the initial few cycles of taking a service group offline, the agent's memory usage can spike.

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 service group for DB2

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, 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 DB2 -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, see the chapter on reconfiguring VCS from the command line in:

    Veritas Cluster Server Users Guide.

Troubleshooting the DB2 agent

This appendix includes the following topics:

■ Creating a db2profile for environment variables

■ Setting the RestartLimit attribute

Creating a db2profile for environment variables

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 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 db2profile is the name of the logon ID.

Place the variables that your 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

VCS starts multiple partitions simultaneously, which can lead to a race condition. The DB2 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.

Resource type definition for DB2 UDB

This appendix includes the following topics:

- About the resource type definitions for DB2 UDB
- Db2udb resource type attributes

About the resource type definitions for DB2 UDB

This stub is content for the shared content.

See "Solaris sample configuration files" on page 63.

DB2 UDB type definition file: Db2udbTypes.cf

The following is the type definition for the DB2 agent:

```
type Db2udb (
    static str ContainerType = Zone
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Db2udb"
    static str AgentFile = "/opt/VRTSagents/ha/bin/Db2udb/
    Db2udbAgent"
    static keylist SupportedActions = { VRTS_GetInstanceName,
    VRTS_GetRunningServices }
    static int CleanTimeout = 240
    static int MonitorTimeout = 240
    static int OfflineTimeout = 240
    static int OnlineRetryLimit = 2
    static int OnlineTimeout = 180
    static int OnlineWaitLimit = 1
```

```
    static int RestartLimit = 3
    static int ToleranceLimit = 1
    static str ArgList[] = { DB2InstOwner, DB2InstHome,
    IndepthMonitor, DatabaseName, NodeNumber, StartUpOpt,
    ShutDownOpt, AgentDebug, Encoding, WarnOnlyIfDBQueryFailed,
    LastWarningDay, ContainerName }
    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
    str ContainerName
)
```

Db2udb resource type attributes

The DB2 resource has several required and optional attributes.

Table B-1 shows the required attributes for the agent for DB2.

Table B-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 **Caution:** Incorrect changes to this attribute can result in DB2 entering an inconsistent state.

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

Table B-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). **Note:** 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. 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
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 B-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 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. Before this release, IndepthMonitor performed a default SQL query to the database. In 5.0, this default query no longer exists. 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

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

Optional attributes	Description
Encoding	Specifies the operating system encoding corresponding to DB2 UDB encoding for display of DB2 UDB output.
	For example, if the environment variable LANG is set to "ja," then "eucJP" is the Solaris value for Encoding. Refer to DB2 UDB and Solaris documentation for respective encoding values. The default is "".
	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
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
ContainerName	Name of the Solaris zone (Solaris 10 only)
	Type and dimension: string-scalar

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

Table B-3   Internal attributes for the agent for DB2

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

Sample configurations

This appendix includes the following topics:

■ Solaris sample configuration files

Solaris sample configuration files

This section shows example DB2 UDB configurations for Solaris.

Non-MPP main.cf configuration for Solaris

The following main.cf configuration file reflects DB2 UDB in an ESE multi-partition instance SMP environment. Two database partitions are shown.

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_clus (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
    )

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

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

```
         NotifyThreshold = 0, NotifyTimeLimit = 0 }
          )


     group db2_grp1 (
         SystemList = { sysA = 0, sysB = 1 }
         AutoStartList = { sysA }
         )


         Db2udb db2udb1 (
             DB2InstOwner = db2inst1
             DB2InstHome = "/db2_mnt/db2inst1"
             IndepthMonitor = 1
             DatabaseName = DWCNTRL
             NodeNumber = 0
             )


         Db2udb db2udb2 (
             DB2InstOwner = db2inst1
             DB2InstHome = "/db2_mnt/db2inst1"
             IndepthMonitor = 0
             NodeNumber = 1
             )


         DiskGroup db2dg1 (
             DiskGroup = db2dg1
             )

       IP db2ip1 (
           Device = bge0
           Address = "192.2.40.21"
           )


         Mount db2mnt1 (
             MountPoint = "/db2_mnt/db2inst1"
             BlockDevice = "/dev/vx/dsk/db2dg1/db2dg1home"
             FSType = vxfs
             MountOpt = rw
             FsckOpt = "-y"
             )


         NIC db2nic1 (
             Device = bge0
             NetworkType = ether
```

```
        )

    Volume db2vol1 (
        Volume = db2dg1home
        DiskGroup = db2dg1
        )

    db2ip1 requires db2nic1
    db2mnt1 requires db2vol1
    db2udb1 requires db2ip1
    db2udb1 requires db2mnt1
    db2vol1 requires db2dg1
    db2udb2 requires db2ip1
    db2udb2 requires db2mnt1
```

MPP main.cf configuration for Solaris

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 "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_mpp (
    CounterInterval = 5
    )

system sysA (
    )

system sysB (
    )

system sysC (
    )

system sysD (
    )
```

```
group cvm (
     SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
     AutoFailOver = 0
     Parallel = 1
     AutoStartList = { sysA, sysB, sysC, sysD }
     )

    CFSMount db2cfsmnt (
        MountPoint = "/db2_mnt/db2inst1"
        BlockDevice = "/dev/vx/dsk/cdb2dg1/cdb2dg1home"
        Primary = sysD
        )

        CFSfsckd vxfsckd (
        )

    CVMCluster cvm_clus (
        Critical = 0
        CVMClustName = db2_mpp
        CVMNodeId = { sysA = 0, sysB = 1, sysC = 2,
            sysD = 3 }
        CVMTransport = gab
        CVMTimeout = 200
        )

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

    db2cfsmnt requires db2dg
    db2cfsmnt requires vxfsckd
    db2dg requires cvm_clus
    vxfsckd requires qlogckd

    // resource dependency tree
    //
    //          group cvm
    //          {
    //          CFSMount db2cfsmnt
    //              {
    //              CVMVolDg db2dg
```

```
//                  {
//                      CVMCluster cvm_clus
//                  }
//              CFSfsckd vxfsckd
//                  {
//                      CFSQlogckd qlogckd
//                  }
//              }
//          }

group db2mpp_grp0 (
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysA }
    )

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

    IP Db2_IP0 (
        Device = bge0
        Address = "10.118.2.144"
        NetMask = "255.255.248.0"
        )

    NIC Db2_NIC0 (
        Device = bge0
        NetworkHosts = { "10.118.11.90" }
        )

    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 = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysB }
    )

    Db2udb db2udb1 (
        DB2InstOwner = db2inst1
        DB2InstHome = "/db2_mnt/db2inst1"
        IndepthMonitor = 1
        DatabaseName = TEST1
        NodeNumber = 1
        )

    IP Db2_IP1 (
        Device = bge0
        Address = "10.118.2.145"
        NetMask = "255.255.248.0"
        )

    NIC Db2_NIC1 (
        Device = bge0
        NetworkHosts = { "10.118.11.90" }
        )

    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 = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysC }
    )

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

    IP Db2_IP2 (
        Device = bge0
        Address = "10.118.2.146"
        NetMask = "255.255.248.0"
        )

    NIC Db2_NIC2 (
        Device = bge0
        NetworkHosts = { "10.118.11.90" }
        )

    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 = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysD }
    )

    Db2udb db2udb3 (
        DB2InstOwner = db2inst1
        DB2InstHome = "/db2_mnt/db2inst1"
        NodeNumber = 3
        )

    IP Db2_IP3 (
        Device = bge0
        Address = "10.118.2.147"
        NetMask = "255.255.248.0"
        )

    NIC Db2_NIC3 (
        Device = bge0
        NetworkHosts = { "10.118.11.90" }
        )

    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
    //                  }
```

```
//                    }
//              }
```

DB2 instance running in a Solaris zone

This sample DB2 instance running on Solaris gives the following configuration, which reflects a DB2 UDB instance running in a Solaris 10 zone environment.

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2zone (
    UserNames = { "z_zoneres@vcs_lzs@sysA.engba.veritas.com" = Gn,
        "z_z1@vcs_lzs@sysA.engba.veritas.com" = aH }
    ClusterAddress = "10.178.6.32"
    SecureClus = 1
    CredRenewFrequency = 0
    CounterInterval = 5
    )

system sysA (
    )

system sysB (
    )

group Db2grp (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA, sysB }
    Administrators = { "z_zoneres@vcs_lzs@sysA.engba.veritas.com" }
    )

    DiskGroup z-dg (
        DiskGroup = db2dg1
        )

    IP ipres (
        Device = bge0
        Address = "10.178.6.28"
        ContainerName = "zone1"
        )

    Mount z-mnt (
```

```
            MountPoint = "/zones/db2data"
            BlockDevice = "/dev/vx/dsk/db2dg1/db2dg1data"
            FSType = vxfs
            FsckOpt = "-y"
            )

        NIC z-nic (
            Device = bge0
            NetworkType = ether
            NetworkHosts = { "10.178.2.4" }
            )

        Volume z-vol (
            Volume = db2dg1data
            DiskGroup = db2dg1
            )

        Zone zoneres (
            ZoneName = zone1
            )

        Db2udb db2udb1 (
            ContainerName = "zone1"
            DB2InstOwner = "db2inst1"
            DB2InstHome = "/db2inst1"
            )

        ipres requires z-nic
        ipres requires zoneres
        z-mnt requires z-vol
        z-vol requires z-dg
        zoneres requires z-nic
        zoneres requires z-mnt
        db2udb1 requires zoneres

        // resource dependency tree
        //
        //        group Db2grp
        //        {
        //        IP ipres
        //            {
        //            NIC z-nic
        //            }
```

```
//           Mount z-mnt
//               {
//               Volume z-vol
//                   {
//                   DiskGroup z-dg
//                   }
//               }
//           Zone zoneres
//               {
//               NIC z-nic
//               }
//           }
```

Index