Cluster Server Agent for Documentum Connection Broker Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

5.1

September 2015

Cluster Server Agent for Connection Broker 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.

Agent Version: 5.1

Document version: 5.1 Rev 0

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

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 support 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/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

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

www.symantec.com/business/support/contact_techsupp_static.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/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as 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 support 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

Support agreement resources

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

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

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec website.

https://sort.symantec.com/documents

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

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Technical Support 3

Chapter 1 Introducing the agent for Connection Broker 8

- About the Cluster Server agent for Connection Broker 8
- Supported software 9
- How the agent makes Connection Broker highly available 9
- Connection Broker agent functions 9
  - Online 9
  - Offline 10
  - Monitor 10
  - Clean 11

Chapter 2 Installing and configuring Connection Broker for high availability 12

- About Connection Broker 12
- Uniquely identifying Connection Broker instances 14
- About configuring Connection Broker for high availability 15
- Configuring the Connection Broker for high availability 15
  - Synchronizing accounts and services 15
  - Removing physical host dependencies 15

Chapter 3 Installing, upgrading, and removing the agent for Connection Broker 17

- Before you install the Cluster Server agent for Connection Broker 17
- About the ACC library 18
- Installing the ACC library 18
  - Installing the ACC library IPS package on Oracle Solaris 11 systems 19
  - Installing the ACC library package on Solaris brand non-global zones 20
- Installing the agent in a VCS environment 21
  - Installing the agent IPS package on Oracle Solaris 11 systems 22
  - Installing agent packages on Solaris brand non-global zones 23

Uninstalling the agent in a VCS environment 24
Removing the ACC library 25

Chapter 4 Configuring the agent for Connection Broker 26

About configuring the Cluster Server agent for Connection Broker 26
Importing the agent types files in a VCS environment 26
Documentum Connection Broker agent attributes 28
Executing a customized monitoring program 32

Chapter 5 Configuring the service groups for Connection Broker using the CLI 33

About configuring service groups for Connection Broker 33
Before configuring the service groups for Connection Broker 34
Configuring service groups for Connection Broker 34
Generating the environments file for Connection Broker 36

Chapter 6 Troubleshooting the agent for Connection Broker 38

Using the correct software and operating system versions 38
Meeting prerequisites 38
Configuring ConnectionBroker resources 39
Starting the ConnectionBroker instance outside a cluster 39
Verifying Second Level Monitor command 40
Reviewing error log files 40
    Using ConnectionBroker log files 40
    Reviewing cluster log files 41
    Using trace level logging 41

Appendix A Sample Configurations 43

About sample configurations for the agents for Connection Broker 43
Sample agent type definition 43
Sample configuration 44
Sample service group configuration 46
Sample service group dependency for Content Server and Connection Broker 47

Index 49

Chapter 1

Introducing the agent for Connection Broker

This chapter includes the following topics:

- About the Cluster Server agent for Connection Broker
- Supported software
- How the agent makes Connection Broker highly available
- Connection Broker agent functions

About the Cluster Server agent for Connection Broker

Cluster Server (VCS) agents monitor specific resources within an enterprise application. They determine the status of resources and start or stop them according to external events.

The Cluster Server agent for Connection Broker provides high availability for a Connection Broker in a clustered environment. The agent brings specific instances of the Connection Broker online, monitors the instance, and brings it offline. The agent monitors the processes of the Connection Broker instance and shuts down the Connection Broker in case of a failure.

See the Agent Pack Release Notes for the latest updates or software issues for this agent.

Supported software

For information on the software versions that the Cluster Server agent for Connection Broker supports, see the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

How the agent makes Connection Broker highly available

The Cluster Server agent for Connection Broker continuously monitors the Connection Broker processes to verify that they function properly.

The agent provides the following levels of application monitoring:

- Primary or Basic monitoring
  This mode has Process check and Health check monitoring options. With the default Process check option, the agent verifies that the Connection Broker processes are present in the process table. Process check cannot detect whether processes are in the hung or stopped states.
- Secondary or Detail monitoring
  In this mode, the agent runs a utility to verify the status of the Connection Broker. The agent detects application failure if the monitoring routine reports an improper function of the Connection Broker processes. When this application failure occurs, the Connection Broker service group fails over to another node in the cluster.
  Thus, the agent ensures high availability for Connection Brokers.

Connection Broker agent functions

The agent consists of resource type declarations and agent executables. The agent executables implement online, offline, monitor, and clean operations.

Online

The online operation performs the following tasks:

- Performs the preliminary check to ensure that the Connection Broker instance is not online on the specified node in the cluster.
- Uses the Connection Broker start script `dm_launch_`*`CBName`* to start the Connection Broker instance using the name of Connection Broker and Connection Broker initialization file. The online function sources a shell script or a program that the `EnvFile` attribute specifies. The script or program ensures

that the required shell environment variables are properly set before it executes the start script.

- Ensures that the Connection Broker instance is up and running successfully. The operation uses the wait period that the OnlineTimeout attribute specifies, to enable the Connection Broker instance to initialize completely before it allows the monitor function to probe the resource.

Offline

The offline operation performs the following tasks:

- Verifies that the Connection Broker instance is not already offline.
- Uses the Connection Broker stop script `dm_stop_`*`CBName`* to stop the Connection Broker instance using the name of the repository the Connection Broker manages. The offline function also sources a shell script or a program that the `EnvFile` attribute specifies. This script or program ensures that the required shell environment variables are properly set before it executes the stop script.
- Ensures that the Connection Broker instance is given enough time to go offline successfully. The operation uses a wait period that the OfflineTimeout attribute specifies, to allow the Connection Broker instance to complete the offline sequence before it allows further probing of the resource.

Monitor

The monitor function monitors the state of the Connection Broker instance running on all nodes within the cluster.

The monitor operation performs following tasks:

- The first level check scans the system process table and searches the processes that must be running for Connection Broker instance. If the first level check does not find these processes running on the node, the check exits immediately, and reports the Connection Broker instance as offline.
- If the SecondLevelMonitor attribute is set to greater than 0, the monitor function performs a second-level check to determine the status of the Connection Broker instance. The second-level check runs the `dmqdocbroker` utility or program, provide by Connection Broker installation to ensure that the processes are truly available for Connection Broker instance.
- Depending upon the MonitorProgram attribute, the monitor function can perform a customized check using a user-supplied monitoring utility.
  See “Executing a customized monitoring program” on page 32.

Clean

In case of a failure or after an unsuccessful attempt to be online or offline, the clean function removes any Connection Broker processes remaining in the system.

The clean operation performs following tasks:

- Attempts to gracefully shut down the Connection Broker instance.
- If a graceful shutdown fails, the clean function looks for all the processes running for the Connection Broker instance, and cleans the processes by killing them.

Chapter 2

Installing and configuring Connection Broker for high availability

This chapter includes the following topics:

- About Connection Broker
- Uniquely identifying Connection Broker instances
- About configuring Connection Broker for high availability
- Configuring the Connection Broker for high availability

About Connection Broker

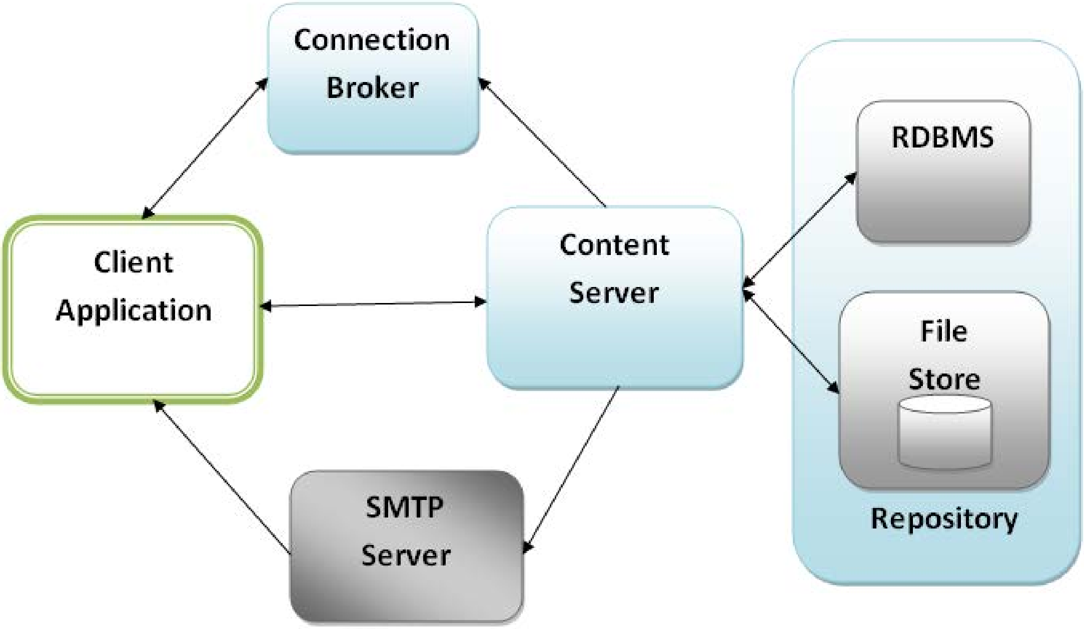
The Documentum connection broker is a process that provides client sessions with connection information. To establish a connection, a client session must know where to find a server that accesses the requested repository. When a client session is opened, the client contacts the connection broker and requests the information it needs to connect with a server for the requested repository. The connection brokers that can handle a client's connection request are defined in the client's dfc.properties file.

The connection broker sends back the IP address for the host where the Content Server resides and the port number that the Content Server uses. If there are multiple servers for the repository, the connection broker returns connection information for all of them. The client session uses the information to choose a server and open the connection. Clients can request a connection through a particular server, any server on a particular host, or a particular server on a specified host. Each connection broker has information about servers and the repositories they access.

A connection broker keeps track of when it last heard from a server and when it expects to hear from that server. If it does not receive a broadcast from a server at the expected time, it sets the server's status to presumed down. It keeps the entry for a non-broadcasting server for a certain number of hours. At the end of the allotted time (the keep entry interval), if the connection broker has not heard from the server, it removes the server from its registry. When a system administrator shuts down a server gracefully, the server sends out a message informing the connection brokers that it is going down. The connection brokers that hear that message set the server's status to stopped. Later, when the server is restarted, it rebroadcasts its information and the connection brokers update their entries for the server and reset its status.

A client is anything requesting a connection to a server. For example, a client can be Webtop, an external application, or a user working with Documentum Administrator. In each case, at the start of a new client session, a connection must be established with a repository. To establish the initial connection, the client application first sends a message to a connection broker asking for the service information that it needs to make the requested connection. Each client session has a default connection broker. Generally, clients also have backup connection brokers to ensure continuous service. The default connection brokers and backup connection brokers for clients are defined in the dfc.properties file. Each EMC Documentum web-based client application contains a dfc.properties file that is packaged with the application's WAR file.

Figure 2-1 A simple deployment topology



In simple configuration, connection broker can be installed and started on the Content Server host or the Content Server can project one or more connection brokers that are located on a different host.

Uniquely identifying Connection Broker instances

For multiple Connection Broker instances running concurrently on a single node, the Cluster Server agent must be able to uniquely identify each of the Connection Broker instance on that system. Each Connection Broker instance has a unique configuration or an initialization file. The agent uses the `InitFile` attribute value to identify the Connection Broker instance uniquely.

Differentiating the Connection Broker instances is important to identify each Connection Broker uniquely. When the Cluster Server agent kills the processes of a non-responsive or failed Connection Broker instance, in the absence of an unique `InitFile` for each Connection Broker instance, the agent may kill processes for more than one Connection Broker instance during a clean operation.

About configuring Connection Broker for high availability

To configure the Connection Broker instance for high availability ensure that you install it on all the nodes in the cluster and associate it to a virtual host/IP.

During the service group configuration ensure that you create a service group dependency between the Content Server and Connection Broker agent service groups. The dependency type between the Connection Broker service group and service group containing the resource for Content Server must be 'Online Global Soft'.

In a typical configuration when the Connection Broker goes offline the Content Server cannot establish any new client connection. However, the existing client connections remain unaffected. Thus, the service group containing the Content Server should not fault or fail over in case the Connection Broker service group faults. The Connection Broker can be running on any node and the Content Server configuration would be aware of the Connection Broker hostname and the port to which it has to broadcast the information. Once the Connection Broker is online again, the Content Server broadcasts the information to the Connection Broker and the session is resumed. This eliminates the need to restart the Content Server.

Hence the dependency type between two service group is 'Online Global Soft' with the Content Server group as the parent and the Connection Broker group as the child.

Configuring the Connection Broker for high availability

This section provides the information about the tasks you must perform to configure Connection Broker for high availability.

Synchronizing accounts and services

Ensure that you synchronize accounts and services in the following ways:

- Synchronize the Documentum user accounts user name, UNIX uid, group name, and UNIX gid across all nodes in the cluster.
- The /etc/services entries should be consistent on all cluster nodes.

Removing physical host dependencies

Perform the following tasks to remove the physical host dependencies:

- Modify the host parameter in the script `dm_launch_`*`CBName`* and pass the parameter to the start command `dmdocbroker`.

  `host =` *connection broker virtual hostname*

  Example:

  ```
  host=CBHost

   ./dmdocbroker -host $host -port 1489

  -init_file/documentum/dba/$CBName.ini $@ >> $tlogfile 2>&1 &
  ```

- Edit the configuration file and modify the parameter in the script.

  Edit the connection broker configuration file (`$DOCUMENTUM/dba/`*`$CBName`*`.ini`) to set the virtual host and the port on which the connection broker listens. Change the host, port, and service from the [DOCBROKER_CONFIGURATION] section, present in the configuration file.

  `host =` *Connection Broker virtual hostname*

  `service =` *service_name*

  `port =` *port_number*

- Modify the —T parameter option to the `dmshutdown` command in dm_stop_*CBName* script.

  Example:

  ```
  ./dmshutdown docbroker -B —T virtual_hostname -P -N1489 $@
  ```

Chapter

3

Installing, upgrading, and removing the agent for Connection Broker

This chapter includes the following topics:

- Before you install the Cluster Server agent for Connection Broker
- About the ACC library
- Installing the ACC library
- Installing the agent in a VCS environment
- Uninstalling the agent in a VCS environment
- Removing the ACC library

Before you install the Cluster Server agent for Connection Broker

You must install the Cluster Server agent for Connection Broker on all the systems that will host ConnectionBroker service groups.

Before you install the agent for Connection Broker, ensure that the following prerequisites are met.

- Install and configure Cluster Server.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Install the latest version of ACC Library.

To install or update the ACC Library package, locate the library and related documentation in the Agent Pack tarball,

See "Installing the ACC library" on page 18.

About the ACC library

The operations of a Cluster Server agent depend on a set of Perl modules known as the ACC library. The library must be installed on each system in the cluster that runs the agent. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

Instructions to install or remove the ACC library on a single system in the cluster are given in the following sections. The instructions assume that the ACCLib tar file has already been extracted.

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the ACC library

Install the ACC library on each system in the cluster that runs an agent that depends on the ACC library.

To install the ACC library

1. Log in as a superuser.
2. Download ACC Library.

   You can download either the complete Agent Pack tar file or the individual ACCLib tar file from the Symantec Operations Readiness Tools (SORT) site (https://sort.symantec.com/agents).
3. If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX	*cd1*/aix/vcs/application/acc_library/*version*_library/pkgs
HP-UX	*cd1*/hpux/generic/vcs/application/acc_library/*version*_library/pkgs
Linux	*cd1*/linux/generic/vcs/application/acc_library/*version*_library/rpms
Solaris	*cd1*/solaris/*dist_arch*/vcs/application/acc_library/*version*_library/pkgs where *dist_arch* is sol_sparc

4. If you downloaded the individual ACCLib tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).
5. Install the package. Enter **Yes**, if asked to confirm overwriting of files in the existing package.

AIX	`# installp -ac -d VRTSacclib.bff VRTSacclib`
HP-UX	`` # swinstall -s `pwd` VRTSacclib ``

Linux	`# rpm -i \`
	`VRTSacclib-`*`VersionNumber`*`-GA_GENERIC.noarch.rpm`

Solaris	`# pkgadd -d VRTSacclib.pkg`

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the ACC library IPS package on Oracle Solaris 11 systems

To install the ACC library IPS package on an Oracle Solaris 11 system

1. Copy the `VRTSacclib.p5p` package from the pkgs directory to the system in the `/tmp/install` directory.
2. Disable the publishers that are not reachable as package install may fail, if any, of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

3. Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSacclib.p5p Symantec
```

4. Install the package.

```
# pkg install --accept VRTSacclib
```

5. Remove the publisher from the system.

```
# pkg unset-publisher Symantec
```

6. Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher name>
```

Installing the ACC library package on Solaris brand non-global zones

With Oracle Solaris 11, you must install the ACC library package inside non-global zones. The native non-global zones are called Solaris brand zones.

To install the ACC library package on Solaris brand non-global zones

1. Ensure that the SMF service `svc:/application/pkg/system-repository:default` and `svc:/application/pkg/zones-proxyd:default` are online on the global zone.

   ```
   # svcs svc:/application/pkg/system-repository:default
   ```

   ```
   # svcs svc:/application/pkg/zones-proxyd:default
   ```

2. Log on to the non-global zone as a superuser.

3. Ensure that the SMF service `svc:/application/pkg/zones-proxy-client:default` is online inside the non-global zone:

   ```
   # svcs svc:/application/pkg/zones-proxy-client:default
   ```

4. Copy the `VRTSacclib.p5p` package from the pkgs directory to the non-global zone (for example, at the `/tmp/install` directory).

5. Disable the publishers that are not reachable, as package install may fail if any of the already added repositories are unreachable.

   ```
   # pkg set-publisher --disable <publisher name>
   ```

6. Add a file-based repository in the non-global zone.

   ```
   # pkg set-publisher -g/tmp/install/VRTSacclib.p5p Symantec
   ```

7. Install the package.

   ```
   # pkg install --accept VRTSacclib
   ```

8. Remove the publisher on the non-global zone.

   ```
   # pkg unset-publisher Symantec
   ```

9. Clear the state of the SMF service, as setting the file-based repository causes the SMF service `svc:/application/pkg/system-repository:default` to go into the maintenance state.

   ```
   # svcadm clear svc:/application/pkg/system-repository:default
   ```

10. Enable the publishers that were disabled earlier.

    ```
    # pkg set-publisher --enable <publisher>
    ```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent in a VCS environment

Install the agent for Connection Broker on each node in the cluster.

Note: The agent package VRTSvcsdctm includes the Cluster Server agents for Connection Broker and Content Server. So, the following procedure to install the agent for ConnectionBroker also installs the agent for ContentServer.

To install the agent in a VCS environment

1 Download the agent from the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

You can download either the complete Agent Pack tar file or an individual agent tar file.

2 Uncompress the file to a temporary location, say /tmp.

3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX	cd1/aix/vcs/application/documentum_agent/ *vcs_version*/*version*_agent/pkgs
HP-UX	cd1/hpux/generic/vcs/application/documentum_agent/ *vcs_version*/*version*_agent/pkgs
Linux	cd1/linux/generic/vcs/application/documentum_agent/ *vcs_version*/*version*_agent/rpms
Solaris	cd1/solaris/*dist_arch*/vcs/application/documentum_agent/ *vcs_version*/*version*_agent/pkgs where, *dist_arch* is sol_sparc.

If you downloaded the individual agent tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).

4 Log in as a superuser.

5. Install the package.

   AIX

   ```
   # installp -ac -d
   VRTSvcsdctm.rte.bff VRTSvcsdctm.rte
   ```

   HP-UX

   ```
   # swinstall -s ‘pwd‘ VRTSvcsdctm
   ```

   Linux

   ```
   # rpm -ihv \
   VRTSvcsdctm-AgentVersion-GA_GENERIC.noarch.rpm
   ```

   Solaris

   ```
   # pkgadd -d . VRTSvcsdctm
   ```

6. After installing the agent package, you must import the agent type configuration file.

Installing the agent IPS package on Oracle Solaris 11 systems

To install the agent IPS package on an Oracle Solaris 11 system

1. Copy the `VRTSvcsdctm.p5p` package from the pkgs directory to the system in the `/tmp/install` directory.
2. Disable the publishers that are not reachable as package install may fail if any of the already added repositories are unreachable.

   ```
   # pkg set-publisher --disable <publisher name>
   ```

   where the publisher name is obtained using the `pkg publisher` command.
3. Add a file-based repository in the system.

   ```
   # pkg set-publisher -g /tmp/install/VRTSvcsdctm.p5p Symantec
   ```

4. Install the package.

   ```
   # pkg install --accept VRTSvcsdctm
   ```

5. Remove the publisher from the system.

   ```
   # pkg unset-publisher Symantec
   ```

6. Enable the publishers that were disabled earlier.

   ```
   # pkg set-publisher --enable <publisher name>
   ```

Installing agent packages on Solaris brand non-global zones

With Oracle Solaris 11, you must install the agent package inside non-global zones. The native non-global zones are called Solaris brand zones.

To install the agent package on Solaris brand non-global zones

1. Ensure that the SMF service `svc:/application/pkg/system-repository:default` and `svc:/application/pkg/zones-proxyd:default` are online on the global zone.

   ```
   # svcs svc:/application/pkg/system-repository:default
   ```

   ```
   # svcs svc:/application/pkg/zones-proxyd:default
   ```

2. Log on to the non-global zone as a superuser.
3. Ensure that the SMF service `svc:/application/pkg/zones-proxy-client:default` is online inside non-global zone:

   ```
   # svcs svc:/application/pkg/zones-proxy-client:default
   ```

4. Copy the `VRTSvcsdctm.p5p` package from the pkgs directory to the non-global zone (for example, at the `/tmp/install` directory).
5. Disable the publishers that are not reachable, as package install may fail if any of the already added repositories are unreachable.

   ```
   # pkg set-publisher --disable <publisher name>
   ```

6. Add a file-based repository in the non-global zone.

   ```
   # pkg set-publisher -g/tmp/install/VRTSvcsdctm.p5p Symantec
   ```

7. Install the package.

   ```
   # pkg install --accept VRTSvcsdctm
   ```

8. Remove the publisher on the non-global zone.

   ```
   # pkg unset-publisher Symantec
   ```

9. Clear the state of the SMF service, as setting the file-based repository causes the SMF service `svc:/application/pkg/system-repository:default` to go into the maintenance state.

   ```
   # svcadm clear svc:/application/pkg/system-repository:default
   ```

10. Enable the publishers that were disabled earlier.

    ```
    # pkg set-publisher --enable <publisher>
    ```

Note: Perform steps 2 through 10 on each non-global zone.

Uninstalling the agent in a VCS environment

You must uninstall the agent for Connection Broker from a cluster while the cluster is active.

Warning: The agent package VRTSvcsdctm includes the agents for Connection Broker and Content Server. So, the following procedure to remove the agent for ConnectionBroker also removes the agent for ContentServer.

To uninstall the agent in a VCS environment

1. Log in as a superuser.
2. Set the cluster configuration mode to read/write by running the following command from any node in the cluster:

   ```
   # haconf -makerw
   ```

3. Remove all ConnectionBroker resources from the cluster. Run the following command to verify that all resources have been removed:

   ```
   # hares -list Type=ConnectionBroker
   ```

4. Remove the agent type from the cluster configuration by running the following command from any node in the cluster:

   ```
   # hatype -delete ConnectionBroker
   ```

   Removing the agent’s type file from the cluster removes the include statement for the agent from the `main.cf` file, but the agent's type file is not removed from the cluster configuration directory. You can remove the agent’s type file later from the cluster configuration directory.

5. Save these changes. Then set the cluster configuration mode to read-only by running the following command from any node in the cluster:

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

6. Use the platform's native software management program to remove the agent for Connection Broker from each node in the cluster.

   Run the following command to uninstall the agent:

AIX	`# installp -u VRTSvcsdctm.rte`
HP-UX	`# swremove VRTSvcsdctm`
Linux	`# rpm -e VRTSvcsdctm`

Solaris	`# pkgrm VRTSvcsdctm`
	Note: To uninstall the agent IPS package on a Solaris 11 system, run the following command:
	`# pkg uninstall VRTSvcsdctm`

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

1. Ensure that all agents that use ACC library are removed.
2. Run the following command to remove the ACC library package.

AIX	`# installp -u VRTSacclib`
HP-UX	`# swremove VRTSacclib`
Linux	`# rpm -e VRTSacclib`

Solaris	`# pkgrm VRTSacclib`
	Note: To uninstall the ACCLib IPS package on a Solaris 11 system, run the following command:
	`# pkg uninstall VRTSacclib`

Chapter

4

Configuring the agent for Connection Broker

This chapter includes the following topics:

- About configuring the Cluster Server agent for Connection Broker
- Importing the agent types files in a VCS environment
- Documentum Connection Broker agent attributes
- Executing a customized monitoring program

About configuring the Cluster Server agent for Connection Broker

After installing the Cluster Server agent for Connection Broker, you must import the agent type configuration file. After importing this file, review the attributes table that describes the resource type and its attributes, and then create and configure ConnectionBroker resources.

To view the sample agent type definition and service groups configuration:

See “About sample configurations for the agents for Connection Broker” on page 43.

Importing the agent types files in a VCS environment

To use the agent for Connection Broker, you must import the agent types file into the cluster.

You can import the agent types file using the VCS graphical user interface or using the command line interface.

To import the agent types file using the VCS graphical user interface

1. Start the Cluster Manager (Java Console) and connect to the cluster on which the agent is installed.
2. Click **File > Import Types**.
3. In the **Import Types** dialog box, select the following file:

VCS 4.x	■ AIX ■ HP-UX ■ Linux ■ Solaris	`/etc/VRTSvcs/conf/sample_ConnectionBroker/` `DocumentumTypes.cf`
VCS 5.x or later	■ AIX ■ HP-UX ■ Linux	`/etc/VRTSagents/ha/conf/ConnectionBroker/` `DocumentumTypes.cf`
VCS 5.0	Solaris	`/etc/VRTSagents/ha/conf/ConnectionBroker/` `DocumentumTypes50.cf`
VCS 5.1 or later	Solaris	`/etc/VRTSagents/ha/conf/ConnectionBroker/` `DocumentumTypes51.cf`

4. Click **Import**.
5. Save the VCS configuration.

   The ConnectionBroker agent type is now imported to the VCS engine.

   Note: The Documentum.cf file contains the agent type definition for ContentServer and ConnectionBroker. Hence, the above procedure will import the agent type definition for both ContentServer and ConnectionBroker agent.

   You can now create ConnectionBroker resources. For additional information about using the VCS GUI, refer to the *Cluster Server Administrator's Guide*.

To import the agent types file using the command line interface (CLI):

1. If VCS is running, run the `/etc/VRTSagents/ha/conf/ConnectionBroker/DocumentumTypes.cmd` file from the command line.
2. If VCS is not running, perform the following steps:

1. Copy the agent types file from the `/etc/VRTSagents/ha/conf/ConnectionBroker/<AgentTypes_file>` directory to the `/etc/VRTSvcs/conf/config` directory.

   Where, <AgentTypes_file> is chosen according to the following table:

VCS 4.x	AIX HP-UX Linux Solaris	`/etc/VRTSvcs/conf/sample_ConnectionBroker/DocumentumTypes.cf`
VCS 5.x or later	AIX HP-UX Linux	`/etc/VRTSagents/ha/conf/ConnectionBroker/DocumentumTypes.cf`
VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/ConnectionBroker/DocumentumTypes50.cf`
VCS 5.1 or later	Solaris SPARC	`/etc/VRTSagents/ha/conf/ConnectionBroker/DocumentumTypes51.cf`

2. Include the agent types file in the main.cf file.
3. Start HAD.

Documentum Connection Broker agent attributes

Refer to the required attributes and optional attributes while configuring the agent for Connection Broker.

Table 4-1 lists the required attributes for the ConnectionBroker agent.

Table 4-1 Required attributes

Required attributes	Description
InitFile	Specifies the configuration or Initialization file for the Connection Broker instance. The attribute value is used as the parameter to start script and also in identifying the Connection Broker instance uniquely. Type and Dimension: string-scaler Default: "" Example: /documentum/dba/DBroker.ini

Table 4-1 Required attributes *(continued)*

Required attributes	Description
CBName	Specifies the connection broker name. The attribute value is used to uniquely identify, start, and stop the connection broker. Type and Dimension: string-scaler Default: "" Example: DBroker
DMBase	Specifies the absolute path of the directory where the Connection Broker binaries `dm_launch_CBName, dm_stop_CBName` reside. Type and Dimension: string-scaler Default: "" /Documentum/dba
Password	Specifies the encrypted password for the Documentum user. Refer to the VCS documentation for more information about VCSEncrypt. Type and Dimension: string-scaler Default: "" Example: codOeoFidleiFig
DMUser	Specifies the user name that the Symantec agent uses to execute the connection broker commands. The user name must be synchronized across the systems in the cluster. The user name must resolve to the same UID and have the same default shell on each system in the cluster. The Symantec Agent entry points use the getpwnam (3c) function call to obtain UNIX user attributes. Hence, the user can be defined locally or can be defined in a common repository (NIS, NIS+, or LDAP). If the user is defined to a repository, the agent will fail if the access to the repository fails. The supported shell environments are: ksh, sh, and csh. Type and Dimension: string-scaler Default: "" Example: cvradm

Table 4-1 Required attributes *(continued)*

Required attributes	Description
EnvFile	Specifices the absolute path to the file that must be sourced with the UNIX shell. Source this file to set the environment before executing Connection Broker scripts for online, offline, monitor, and clean operations. The shell environments supported are: ksh, sh, and csh. **Note:** Ensure that the syntax of this file is in accordance with the user shell that the DMUser attribute specifies. Review the information about how to generate environments file for Documentum ConnectionBroker. See "Generating the environments file for Connection Broker" on page 36. Type and Dimension: string-scaler Default: "/dev/null" Example: /documentum/envfile
ResLogLevel	Specifies the logging detail performed by the agent for the resource. The valid values are as follows: ■ ERROR: Only logs error messages. ■ WARN: Logs above plus warning messages. ■ INFO: Logs above plus warning messages. ■ TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. Type and Dimension: string-scaler Default: INFO Example: TRACE

Table 4-2 lists the optional attributes for the ConnectionBroker agent.

Table 4-2 Optional attributes

Optional attributes	Description
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the Connection Broker. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval. This interpretation may be extended to other values. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then the second level check is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Also, verify the second level monitoring utility before enabling second level monitor. See "Verifying Second Level Monitor command" on page 40. Type and Dimension: integer-scaler Default: 0 Example: 5
MonitorProgram	Specifies the absolute path of an external, the user supplied monitor executable. See "Executing a customized monitoring program" on page 32. Type and Dimension: string-scaler Default: "" Example: /documentum/myMonitor.sh
DMHome	Specifies the absolute path of the Documentum installation directory. The Cluster Server Agent uses the attribute to locate the `dmqdocbroker` command. Type and Dimension: string-scaler Default: "" Example:/documentum/product/6.5/

Table 4-2 Optional attributes *(continued)*

Optional attributes	Description
CBHost	Specifies the virtual host name by which the connection broker is running. Type and Dimension: string-scaler Default: "" Example: DBHost
CBPort	Specifies the port number by which the connection broker is listening. Type and Dimension: integer-scaler Default; "1489" Example: 1489

Executing a customized monitoring program

The monitor function executes a custom monitor program to perform an additional Connection Broker state check. The monitor function executes the utility specified in the MonitorProgram attribute, if:

- The specified utility is a valid executable file.
- The first-level process check indicates that the Connection Broker instance is online.
- The SecondLevelMonitor attribute is either set to 0 or 1, and the second-level check indicates that the Connection Broker instance is online.
- The SecondLevelMonitor attribute is set to greater than 1, but the second-level check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

110 or 0	ConnectionBroker server instance is online
100 or 1	ConnectionBroker server instance is offline
99	ConnectionBroker server instance is unknown
Any other value	ConnectionBroker server instance is unknown

To ensure that the customized utility is always available to the agent, Symantec recommends storing the file in a shared directory that is available on an online node.

Configuring the service groups for Connection Broker using the CLI

This chapter includes the following topics:

- About configuring service groups for Connection Broker
- Before configuring the service groups for Connection Broker
- Configuring service groups for Connection Broker
- Generating the environments file for Connection Broker

About configuring service groups for Connection Broker

Configuring the Connection Broker service group involves creating the ConnectionBroker 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 service groups using one of the following:

- The Cluster Manager (Java console)
- Veritas Infoscale Operations Manager
- The command line

See "Configuring service groups for Connection Broker" on page 34.

Before configuring the service groups for Connection Broker

Before you configure the ConnectionBroker service group, you must:

- Verify that Cluster Server is installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Verify that Connection Broker is installed and configured identically on all nodes in the cluster.
  See "About configuring Connection Broker for high availability " on page 15.
- Verify that the Cluster Server agent for Connection Broker is installed on all nodes in the cluster.
  See "Installing the agent in a VCS environment" on page 21.
- Verify that the type definition for the Cluster Server agent for Connection Broker is imported into the VCS engine.
  See "Importing the agent types files in a VCS environment" on page 26.

Configuring service groups for Connection Broker

While setting up a cluster, you must ensure that the cluster has some spare capacity to handle the Connection Broker failover scenarios.

The cluster should be able to provide application failover by encapsulating the resources required for an application into a service group. A service group is a virtualized application that can switch between the cluster nodes. It contains a set of dependent resources, such as IP addresses, NIC cards, and dependent application processes. It also includes logic about the dependencies between the application components.

These service groups should thus be configured such that the cluster can start, stop, monitor, and switch the service groups between the nodes, depending upon the server faults or resource faults. An administrator should also be proactively able to move a service group between cluster nodes to perform preventative maintenance or apply patches.

Perform the following steps to add a service group for Connection Broker

1. Create a service group for Connection Broker.

   ```
   # hagrp -add DCM652-CB
   ```

   For more details on creating a service group refer to, *Cluster Server Administrator's Guide*.

2. Modify the SystemList attribute for the group, to add systems.

   For example,

   ```
   # hagrp -modify DCM652-CB SystemList systemA 0 systemB 1
   ```

3. Create resources for NIC and IP in the service group.

   For example,

   ```
   # hares -add DCM652-CB_nic NIC DCM652-CB
   ```

   ```
   # hares -DCM652-CB_ip IP DCM652-CB
   ```

   For more details on creating and modifying resource attributes for NIC, and IP refer to, *Cluster Server Bundled Agents Reference Guide*.

4. Create links between the resources.

   For example,

   ```
   # hares -link DCM652-CB_ip DCM652-CB_nic
   ```

5. Create the resource for the Connection Broker.

   For example,

   ```
   # hares -add DCM652-CB_broker ConnectionBroker DCM652-CB
   ```

   Based on the Connection Broker instance you cluster, modify the resource attributes.

   See “Documentum Connection Broker agent attributes” on page 28.

6. Create resource dependencies for Connection Broker resource.

   The Connection Broker resource depends on the IP and Mount resources.

   ```
   # hares -link DCM652-CB_broker DCM652-CB_ip
   ```

7. Verify the final resource dependencies for DCM652-CB server group.

   For example,

   ```
   # hares -dep

   Group        Parent              Child
   DCM652-CB    DCM652-CB_broker   DCM652-CB_ip
   DCM652-CB    DCM652-CB_ip       DCM652-CB_nic
   ```

Generating the environments file for Connection Broker

Note: The application itself provides the environment file with the recent versions of Documentum. For example, with Documentum version 7.1, the following files are available:

`$DOCUMENTUM/product/version_number/bin/dm_set_server_env.sh` or

`$DOCUMENTUM/product/version_number/bin/dm_set_server_env.csh`

You can edit the environment file and and use it to ensure that VCS agent for Connection Broker works correctly. You should also ensure that the ORACLE_HOME variable is set correctly in the environment file.

To generate the environments file for ConnectionBroker

1. Login as Documentum user using the following command.

   ```
   su - dmadmin
   ```

2. Capture the environment with the following command.

   ```
   env > /home/dmadmin/dmadmin.env
   ```

3. Modify the file according to the Documentum user shell environment.

   For example, if the generated file contains environments for bash shell and Documentum user shell is C shell, convert the file to C shell environments.

   - Edit the dmadmin.env file to add string 'setenv' at the beginning of each line.

- Replace the '=' with space " " in the file.

4 Copy the dmadmin.env file to shared directory and use it as the ConnectionBroker instance environments file in EnvFile attribute. Ensure that the permissions are set properly for user Documentum user.

```
chmod 755 dmadmin.env
```

Note: Before generating the EnvFile, verify the successful execution of start, stop, and second level monitor command with Documentum user environment.

Chapter 6

Troubleshooting the agent for Connection Broker

This chapter includes the following topics:

- Using the correct software and operating system versions
- Meeting prerequisites
- Configuring ConnectionBroker resources
- Starting the ConnectionBroker instance outside a cluster
- Verifying Second Level Monitor command
- Reviewing error log files

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

For information on the software versions that the agent for Connection Broker supports, see the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

Meeting prerequisites

Before installing the agent for Connection Broker, ensure that the following prerequisites are met.

For example, you must install the ACC library on VCS before installing the agent for Connection Broker.

See "Before you install the Cluster Server agent for Connection Broker" on page 17.

Configuring ConnectionBroker resources

Before using ConnectionBroker resources, ensure that you configure the resources properly. For a list of attributes used to configure all ConnectionBroker resources, refer to the agent attributes.

See "Documentum Connection Broker agent attributes" on page 28.

Starting the ConnectionBroker instance outside a cluster

If you face problems while working with a resource, you must disable the resource within the cluster framework. A disabled resource is not under the control of the cluster framework, and so you can test the ConnectionBroker instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the ConnectionBroker instance outside the cluster framework.

Note: Use the same parameters that the resource attributes define within the cluster framework while restarting the resource outside the cluster framework.

A sample procedure to start a Connection Broker instance outside the cluster framework, is described as follows.

To restart the ConnectionBroker outside the VCS framework

1. Log in to the ConnectionBroker node as an DMUser.

   ```
   # su - DMUser
   ```

2. Source the environment file.

   ```
   #. EnvFile
   ```

3. Start the Connection Broker.

   ```
   # DMBase/dm_start_CBName
   ```

If the ConnectionBroker instance works properly outside the cluster framework, attempt to implement the ConnectionBroker instance within the cluster framework.

Verifying Second Level Monitor command

If you have enabled Second Level Monitoring and are facing problems with the Connection Broker agent resource, verify whether the second level monitor command `dmqdocbroker` is working properly outside the cluster control. To ensure proper working, you must disable the resource within the cluster framework.

A disabled resource is not under the control of the cluster framework, and so you can test the ConnectionBroker instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource. Then you can verify the second level monitor command for the ConnectionBroker instance outside the cluster framework.

A sample procedure is illustrated as follows.

To verify the second level monitor command for a ConnectionBroker instance outside the cluster framework

1. Log in to theConnectionBroker node as a DMUser.

   ```
   # su – DMUser
   ```

2. Source the environment file.

   ```
   # . EnvFile
   ```

3. Execute the second level command

   ```
   # DMHome/bin/dmqdocbroker –t CBHost -p CBPort -c ping
   ```

If the command works properly outside the cluster framework, attempt to implement the ConnectionBroker within the cluster framework.

Reviewing error log files

If you face problems while using ConnectionBroker or the agent for Connection Broker, use the log files described in this section to investigate the problems.

Using ConnectionBroker log files

If Connection Broker instance is facing problems, access the Connection Broker log files to diagnose the problem. The Connection Broker log files are located in the `DMBase/log/` directory.

Reviewing cluster log files

In case of problems while using the agent for Connection Broker, you can access the engine log file for more information about a particular resource. The engine log file is located at `/var/VRTSvcs/log/engine_A.log`.

You can also access the ConnectionBroker agent log file for more detailed information. The agent log file is located at /var/VRTSvcs/log/ConnectionBroker_A.log

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each ConnectionBroker resource. You can set this attribute to TRACE, which enables very detailed and verbose logging.

If you set ResLogLevel to TRACE, a very high volume of messages are produced. Symantec recommends that you localize the ResLogLevel attribute for a particular resource.

The LogDbg attribute should be used to enable the debug logs for the ACCLib-based agents when the ACCLIB version is 6.2.0.0 or later and the VCS version is 6.2 or later.

To localize ResLogLevel attribute for a resource

1. Identify the resource for which you want to enable detailed logging.
2. Localize the ResLogLevel attribute for the identified resource:

   ```
   # hares -local Resource_Name ResLogLevel
   ```

3. Set the ResLogLevel attribute to TRACE for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel TRACE -sys SysA
   ```

4. Note the time before you begin to operate the identified resource.
5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.
7. Set the ResLogLevel attribute back to INFO for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel INFO -sys SysA
   ```

8. Save the configuration changes.

   ```
   # haconf -dump
   ```

9. Review the contents of the log file.

   Use the time noted in Step 4 and Step 6 to diagnose the problem.

   You can also contact Symantec support for more help.

To enable debug logs for all resources of type ConnectionBroker

- Enable the debug log.

  ```
  # hatype -modify ConnectionBroker LogDbg DBG_5
  ```

To override the LogDbg attribute at resource level

- Override the LogDbg attribute at the resource level and enable the debug logs for the specific resource.

  ```
  # hares -override ConnectionBroker LogDbg
  # hares -modify ConnectionBroker LogDbg DBG_5
  ```

Sample Configurations

This appendix includes the following topics:

- About sample configurations for the agents for Connection Broker
- Sample agent type definition
- Sample configuration
- Sample service group configuration
- Sample service group dependency for Content Server and Connection Broker

About sample configurations for the agents for Connection Broker

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for Connection Broker. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition

This section lists the sample agent type definition files for Connection Broker agent on different versions of VCS.

For VCS 4.x

```
type ConnectionBroker (
static str ArgList[] = { ResLogLevel, State, IState, CBName, DMHome,
DMBase, CBHost, CBPort, InitFile, DMUser, Password, EnvFile,
MonitorProgram, SecondLevelMonitor }
```

```
         str ResLogLevel = INFO
         str CBName
         str DMHome
         str DMBase
         str CBHost
         int CBPort = 1489
        str InitFile
        str DMUser
         str Password
        str EnvFile = "/dev/null"
         str MonitorProgram
         int SecondLevelMonitor = 0
 )
```

For VCS 5.x

```
type ConnectionBroker (
         static boolean AEPTimeout = 1
         static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
         static str AgentDirectory = "/opt/VRTSagents/ha/bin/ConnectionBroker
static str ArgList[] = { ResLogLevel, State, IState, CBName, DMHome,
DMBase, CBHost, CBPort, InitFile, DMUser, Password, EnvFile,
MonitorProgram, SecondLevelMonitor }
         str ResLogLevel = INFO
         str CBName
         str DMHome
         str DMBase
         str CBHost
         int CBPort = 1489
         str InitFile
         str DMUser
         str Password
         str EnvFile = "/dev/null"
         str MonitorProgram
         int SecondLevelMonitor = 0
 )
```

Sample configuration

This section provides a sample configuration for Connection Broker agent. The sample configuration depicts a graphical view of the resource types, resources, and resource dependencies within the service group.

```
include "types.cf"
include "DocumentumTypes.cf"
cluster cluster1 (
        UserNames = { admin = dlmElgLimHmmKumGlj }
        ClusterAddress = "110.120.162.128"
        Administrators = { admin }
        UseFence = SCSI3
        HacliUserLevel = COMMANDROOT
        )
system systemA  (
       )
system systemB  (
       )
system systemC  (
       )

group DCM652-CB (
        SystemList = { systemA = 0,  systemB =1 }
        )
ConnectionBroker DCM652-CB_broker (
       Critical = 0
       CBName = Docbroker
       DMHome = "/documentum/product/6.5"
       DMBase = "/documentum/dba"
       CBHost = dcm652host1
       InitFile = "/documentum/dba/Docbroker.ini"
       DMUser = cvradm
       Password = ESKqHSh
       EnvFile = "/home/cvradm/.profile"
       MonitorProgram = "/documentum/monitor.sh
)

IP DCM652-CB_ip (
       Critical = 0
       Device = bge0
       Address = "110.229.222.194"
       NetMask = "255.255.255.0"
)
NIC DCM652-CB_nic (
        Device = bge0
        Network Type = ether
   )
DCM652-CB_broker requires DCM652-CB_ip
```

```
DCM652-CB_ip requires DCM652-CB_nic

 // resource dependency tree
//
//      group DCM652-CB
//      {
//      ConnectionBroker DCM652-CB_broker
//          {
//          IP DCM652-CB_ip
//              {
//              NIC DCM652-CB_nic
//              }
//          }
//      }
```

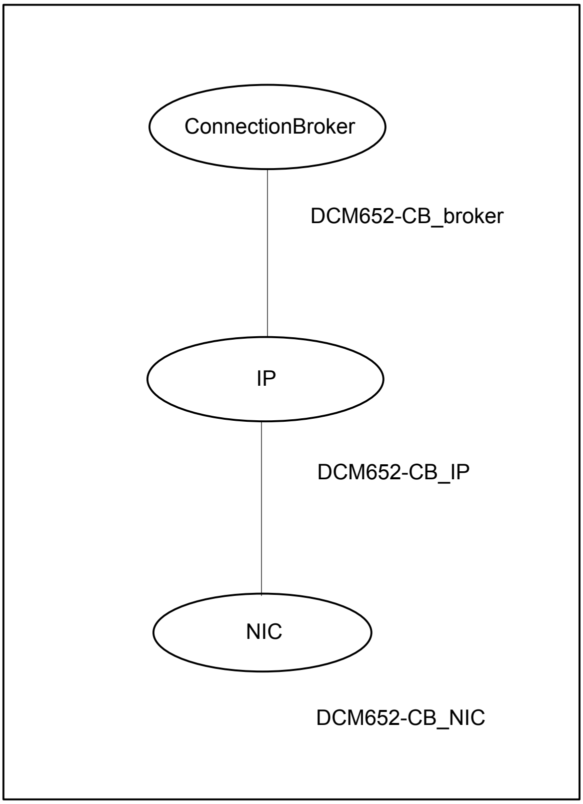
Sample service group configuration

The service group configuration in a cluster depends on some common characteristics that must be part of the configuration design.

The ConnectionBroker instance should have a separate virtual IP address assigned to facilitate network transparency.

Figure A-1 shows a sample service group configuration for ConnectionBroker instance

Figure A-1 Service group configuration for Connection Broker instance

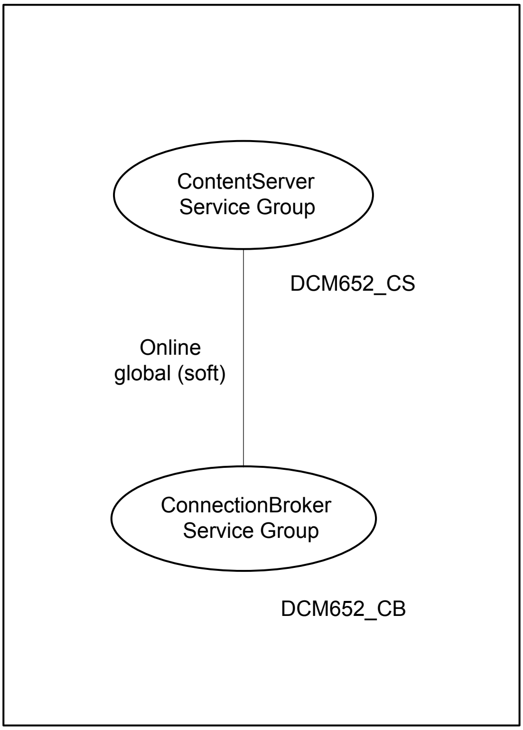


Sample service group dependency for Content Server and Connection Broker

This section includes service groups that show the group dependency between ContentServer service group and ConnectionBroker service group.

Figure A-2 shows the sample service group dependency for Documentum.

Figure A-2 Sample service group dependency



Index

A

About
  - Documentum Connection Broker 12

about
  - configuring Connection Broker for high availability 15
  - configuring service groups 33

about ACC library 18

ACC library
  - installing 18
  - removing 25

agent
  - configuring service groups 34
  - importing agent types files 26
  - installing, VCS environment 21
  - optional attributes 30
  - overview 8
  - required attributes 28
  - uninstalling, VCS environment 24

agent attribute 31–32

agent attributes 28–30, 32
  - MonitorProgram 31
  - SecondLevelMonitor 31

agent configuration file
  - importing 26

agent functions 9
  - clean 11
  - monitor 10
  - offline 10
  - online 9

agent installation
  - general requirements 17
  - steps to install 21

B

before
  - configuring the service groups 34

C

Configuring
  - Connection Broker 15

configuring monitor function 32

ConnectionBroker
  - configuring resources 39
  - starting instance outside cluster 39

E

environments file 36

executing custom monitor program 32

L

logs
  - reviewing cluster log files 41
  - reviewing error log files 40
  - using ConnectionBroker logs 40
  - using trace level logging 41

S

sample
  - service group configuration 46
  - service group dependency 47

sample agent type definition 43

sample configuration files 44

starting the ConnectionBroker instance outside a cluster 39

T

troubleshooting
  - meeting prerequisites 38
  - reviewing error log files 40
    - reviewing cluster log files 41
    - using ConnectionBroker log files 40
    - using trace level logging 41
  - using correct software 38
  - verifying second level monitor command 40

U

uninstalling agent, VCS environment 24