Veritas High Availability Agent 5.0 for Siebel Server Installation and Configuration Guide

Solaris

symantec™

N19602F

Veritas High Availability Agent 5.0 for Siebel Server Installation and Configuration Guide

Veritas High Availability Agent 5.0 for Siebel Server

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

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

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

Third-party legal notices

Technical support

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

Contents

Introducing the Veritas Agent for Siebel Server

The Veritas High Availabilty Agents monitor specific resources within an enterprise application, determine the status of these resources, and start or stop them according to external events. The Veritas High Availability Agent for Siebel Server provides high availability for all Siebel Servers in a cluster.

This guide assumes that the user understands the primary components and basic functionality of a cluster. The guide also assumes that the user has a basic understanding of the Siebel applications architecture and the administration tasks required to start, stop, monitor and operate these components.

Wherever applicable, the Siebel Server and the Siebel Gateway Server are referred as Siebel Server.

The chapter includes:

What's new in this agent

- Merged the previous agents for Siebel Server and Seibel Gateway Server, to form a single Veritas High Availability Agent for Siebel Server.
- Integrated with an enhanced version of ACC library, that is compliant with VAD and VCS.
- Added support for the Siebel Enteprise Server instance to check the database server by verifying ODBC connectivity. If the database server is down, the agent can flag the server resource state as UNKNOWN.
- Added support for monitoring for each component in the specified Siebel Server component group, if the Sadmin attribute is specified. This ability can alert the administrator if a problem occurs.
- Added support for the Siebel Enterprise Server to detect an unreachable Siebel Gateway Server.
- The agent can now detect old `.osdf` and `.shm` files, and can ensure that these files do not exist before bringing a Siebel Server instance online. Also, the agent can re-create the service file, if the old file is empty or the file does not exist.
- The agent now backs up the critical `siebns.dat` file during shutdown and clean operations.
- The agent no longer requires an Activation Key.
- Removed these attributes:
  - FirstMonitorDelay
  - SecondLevelTimeout
- Added the MonitorProgram attribute.
- Added support for Solaris 10 zones in the VCS 5.0 environment.

Supported software

The Veritas High Availability Agent for Siebel Server support the following environments:

Environment	Supported Versions
Veritas Cluster Server	4.1, 5.0
ACC Library	5.0.01.0
Operating Systems	Solaris—8, 9, 10
Siebel Server	7.7, 7.8

About the Siebel Servers

A Siebel Server is a middle-tier platform application that provides back end processes and interactive processes for all Siebel Server clients. These processes are present with the Siebel Server architecture and provide functions such as:

- Mobile Web client synchronization
- Operation of business logic for Siebel Web clients, as well as connectivity and access to the Siebel Database Server and Siebel File System
- Integration with legacy or third-party data
- Automatic assignment of new accounts, opportunities, service requests, and other records
- Workflow management

The Siebel Server supports both multi-process and multi-threaded components. The Server can operate components in background, batch, and interactive modes. Many of the Siebel Server components can operate on multiple Siebel Servers simultaneously to support an increased number of users or larger batch workloads.

About the Seibel Gateway Server

A Siebel Gateway Server is a logical entity that consists of a Siebel Name Server. The Name Server is the primary service of the Siebel Gateway Name Server. This Name Server coordinates the Siebel Enterprise Servers and the Siebel Servers. A single Siebel Gateway Name Server can support Siebel Enterprise Servers.

About the Siebel Name Server

The Siebel Name Server provides support for the Siebel Enterprise Server configuration information, that includes:

- Operational parameters
- Connectivity information
- Definition and assignment of component groups and components

This configuration information is dynamic and may change during a Siebel Server installation or configuration. These changes are logged in the `siebns.dat` file on the Name Server. At startup, the Siebel Server obtains the configuration information from this `siebns.dat` file.

The Siebel Name Server also serves as the dynamic registry for the Siebel Server and for keeping information about component availability. At startup, a Siebel Server within the Siebel Enterprise Server notifies the Name Server of its availability. The Siebel Server stores the connectivity information such as network addresses in the non-persistent store of the Name Server.

Periodically, the Name Server also flushes its current state to the `siebns.dat` file. Enterprise components, such as the Server Manager query the Name Server for the Siebel Server availability and connectivity information. When a Siebel Server shuts down, the relevant information is cleared from the Name Server.

About the Agent for Siebel Server

This Agent for Siebel Server manages the Siebel Servers in a clustered environment. The agent can bring a specific Siebel Server instance online and monitors the state of the Siebel Server. The agent can also detect failures and can shut down the instance in case of a failure.

Agent Operations

The operations that the Veritas Agent for Siebel Server can perform are described below.

Online

The online operation performs the following tasks:

- Verifies that the required attributes are set correctly.
- Verifies whether the Siebel Server instance is not already online. If the instance is online, the online operation exits immediately.

- If any Siebel Server processes remain, the operation kills these processes using the user name associated with the specific resource.
- Removes any existing IPC resources that the user specified in the SiebelUser attribute owns.
- If the ServerType attribute is **GTWY**:
  The online operation executes the `start_ns` script as the user specified in the SiebelUser attribute.
  If the ServerType attribute is **SRVR**:
  - a The online operation checks whether old `.shm` and `.osdf` files are present. If these files exist, the operation deletes these files.
  - b The online operation then checks if the service file is present.
    If the service file is an empty file or does not exist, the online operation re-creates the service file.
  - c The online operation executes the `start_server` script as the user specified in the SiebelUser attribute.

The online operation ensures that the `siebsvc` process for the Siebel Server instance starts successfully.

For the Siebel Enterprise Server, the operation ensures that the percentage CPU utilization falls below 5 for the `siebsvc` process. This fall in the percentage ensures that the Siebel Enterprose Server instantiates completely.

Offline

The offline operation performs the following tasks:

- Verifies that the required attributes are set correctly.
- Verifies that the Siebel Server instance is not offline. If the instance is already offline, the operation kills any existing processes that belong to the user specified in the SiebelUser attribute and the exits.
- If the ServerType attribute is **GTWY**, the operation executes the `stop_ns` script as the user specified in the SiebelUser attribute.
  If the ServerType atrribute is **SRVR**, the operation executes the `stop_server` script as the user specified in the SiebelUser attribute.
- After the offline script is executed, the offline operation kills any existing processes that belong to the user specified in the SiebelUser attribute
- The operation also removes any existing IPC resources that the user specified in the SiebelUser attribute owns.

- If the ServerType attribute is **GTWY**, the operation backs up the siebns.dat file to the /var/tmp/.VRTSSiebel/*<ResourceName>*/BACKUP directory.

The offline operation exits either after all the processes stop successfully, or after the timeout period specified in the OfflineTimeout attribute expires.

Monitor

The monitor operation monitors the states of the Siebel Servers on all nodes within the cluster. The operation performs the following tasks:

- The monitor operation conducts a first level check to determine that the Siebel Server processes that the user specified in the SiebelUser attribute owns, are running on the system in the cluster.
  - If the ServerType attribute is **SRVR**, the operation determines whether the siebsvc process is running for the Siebel Server instance.
  - If the ServerType attribute is **GTWY**, the operation determines whether the siebsvc process that is configured for the *<SiebelRoot>*/sys/siebns.dat file is running for the Siebel Gateway Server instance.

  If the first level check does not find these processes running on the node, the check exits immediately, and reports the instance as OFFLINE.
- If the SecondLevelMonitor attribute is set to greater than 0, the monitor operation conducts a second level check.
  - If the ServerType attribute is **GTWY**, the operation executes the srvredit command to verify the state of the Siebel Server instance. The srvredit command uses the values of the SiebelRoot, SiebelGWHost, and SiebelGWPort attributes for execution. If the command returns 0, the Siebel Server instance is reported as ONLINE.
  - If the ServerType attribute is **SRVR**:
    - The monitor operation attempts an ODBC connection and determines whether the database can be queried.
    - If the ODBC connection is successful, the monitor operation executes the Siebel supplied utility, siebctl to connect to the Siebel Server instance. If the operation is unable to connect to the instance, the instance is flagged as OFFLINE.
    - If the optional attributes, CompGrps, Sadmin, and SadminCrPasswd are defined, the monitor operation executes the Siebel supplied utility, srvrmgr to connect to the Siebel Server instance. The operation uses this utility to query all the CompGrps components in the component groups that are listed in attribute.

      If any of the components in the component groups are not in either Running or Online states, the operation flags the state of the Siebel

Server instance as UNKNOWN. Such a check ensures that the Agent for Siebel Server does not fault the resource and the administrator is sufficiently warned to attempt to correct the issue.

- Depending upon the MonitorProgram attribute, the monitor operation can perform a customized check using a user-supplied monitoring utility. For details about executing a custom monitor program:
  See "Executing a custom monitor utility" on page 24.

Clean

In case of a failure or after an unsuccessful attempt to online or offline an Siebel Server instance, the clean operation performs the following tasks:

- Attempts to gracefully shut down the Siebel Server instance.
- If the instance does not shut down normally, the clean operation kills the remaining Siebel Server processes that the user name specified in the SiebelUser attribute owns.
- Removes any existing IPC resources that the user specified in the SiebelUser attribute owns.
- If the ServerType attribute is **SRVR**, the clean operation checks whether the `.shm` and `.osdf` files are present. If these files are present, the operation deletes these files.
  If the ServerType attribute is **GTWY**, the clean operation takes a backup copy of the `<SiebelRoot>/sys/siebns.dat` file. The backup copy is stored as `/var/tmp/.VRTSSiebel/<ResourceName>/BACKUP/siebns.dat`.

Chapter 2

Installing the Veritas Agent for Siebel Server

This chapter describes how to install the Veritas High Availability Agent for Siebel Server in a cluster. You must install the agent on all systems that host a Siebel Server Service Group.

The chapter includes:

Prerequisites for installing the Agent for Siebel Server

Ensure that you meet the prerequisites before installing the Veritas High Availability Agent for Siebel Server.

Prerequisites for installing the agent in a VCS environment

- Install and configure Veritas Cluster Server.
- Install the latest version of the ACC library.
  See "About the ACC library" on page 16.
- Remove any prior version of this agent.

Prerequisites for installing the agent for Solaris zone support

- Install Siebel to support Solaris zones. For details, refer to the Siebel user documentation.
- Install and configure the VCS 5.0 environment to support Solaris zones. Refer to the VCS user documentation for details.
- Remove any prior version of this agent.

About the ACC library

The operations for Veritas High Availability Agent for Siebel Server depend on a set of PERL modules known as the ACC library. The library must be installed on each system in the cluster that will run the Agent for Siebel Server. The ACC library contains common, reusable functions that perform tasks such as process identification, logging, and system calls.

To install or update the ACC library package, locate the library and related documentation either on the agent CD or in the compressed agent tar file.

Upgrading the Veritas Agent for Siebel Server

To upgrade the agent, first remove the older version of the agent for Siebel Server or the agent for Siebel Gateway Server. Refer the respective user documentation for details about uninstalling the software.

Then follow the instructions below to install the new agent software.

Installing the agent in a VCS environment

Perform the following steps on each system in the cluster.

To install the agent on Solaris systems

1. Log in as `root`.
2. Go to the *<cd_mount>*/solaris/sparc/application/siebel_agent/*<vcs_version>*/*<version>*_agent/pkgs directory.
3. Install the package:

   ```
   # pkgadd -d . VRTSsiebel
   ```

Configuring the Veritas Agent for Siebel Server

After installing the Agent for Siebel Server, you must import the configuration file. After importing the configuration file, you can create and configure a Siebel Server resource. Before you configure a resource, review the attributes table that describes the Siebel Server resource type and its attributes.

The chapter includes:

To view sample agent type definition files and configuration Service Groups:

See "Sample Configurations" on page 43.

Importing the agent types files

To use the Agent for Siebel Server, you must import the `SiebelTypes.cf` file into the cluster.

To import the agent types files to work with VCS

Perform the following steps using the VCS graphical user interface.

1. Start the Veritas Cluster Manager GUI 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:

Version	Directory structure
VCS 4.x	`/etc/VRTSvcs/conf/sample_Siebel/SiebelTypes.cf`
VCS 5.0	`/etc/VRTSagents/ha/conf/Siebel/SiebelTypes.cf`
For Solaris zone support	`/etc/VRTSagents/ha/conf/Siebel/SiebelTypes_zones.cf`

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

The Siebel Server configuration is now imported to the VCS engine. You can now create Siebel Server resources. For additional information about using the VCS GUI, refer to the *Veritas Cluster Server User's Guide*.

For a sample agent definition:

See “Sample agent type definition” on page 44.

Agent attributes

The required and optional attributes required for configuring a Siebel Server instance are described below.

Required attributes

Required attributes	Description
EnvFile *String*	Full path to the file that the agent sources to set the environment before executing any Siebel programs. Typically, this file is the Siebel supplied shell script, `$SIEBEL_ROOT/siebenv.sh`. Symantec recommends storing this file on a shared disk. The supported shell environments are: `ksh`, `sh`, and `csh`. Example: **`/siebel/v78/srv1/siebsrvr/srv1.env`** Default: ""
HostName *String*	The IP address or host name of the virtual host which is configured for the Siebel Server instance. Example 1: **`sbl78sv1sol`** Example 2: **`10.212.98.240`** Default: ""
ResLogLevel *String*	The logging detail performed by the agent for the resource. Valid values are: ■ **`ERROR`**: Only logs error messages. ■ **`WARN`**: Logs above plus warning messages. ■ **`INFO`**: Logs above plus informational 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. Example: **`TRACE`** Default: `INFO`
ServerName *String*	Name of the Siebel Server in the Siebel Enterprise. This attribute is not required for a Siebel Gateway Server. Example: **`siebsrv1`** Default: ""
ServerType *String*	Type of Siebel Server that the agent must support. Valid values are: ■ **`SRVR`**—for a Siebel Enterprise Server instance ■ **`GTWY`**—for a Siebel Gateway Server instance **Note:** In a Siebel Enterprise, you can configure one Siebel Gateway Server instance *only*. Example: **`GTWY`** Default: `SRVR`

Required attributes	Description
SiebelEnterprise *String*	Name of the Enterprise to which the Siebel Server instance belongs. Example: **sbl78sol** Default: ““
SiebelGWHost *String*	The virtual hostname of the machine that hosts the Siebel Gateway Server instance. You must cluster this instance on one of the nodes in the cluster. For a Siebel Gateway Server instance, the values of the HostName and SiebelGWHost attributes must be the same. Example: **sbl78gtwsol** Default: ““
SiebelGWPort *Integer*	The port number on which the Siebel Gateway Server listens. Default: 2320
SiebelRoot *String*	Full path to the installation or root directory of the Siebel Server. Symantec recommends storing this directory on the shared storage device. Example: **/siebel/v78/srv1/siebsrvr** Default: ““
SiebelUser *String*	User name that the Agent for Siebel Server uses to execute the programs for managing a Siebel Server instance. This user name must be unique within all the systems in a cluster. The shell specified for the user names must also be the same on all the systems in the cluster. Example: **sbl78sv1** Default: ““

Optional attributes

Optional attribute	Description
CompGrps *String*	A comma separated list of the name aliases (CG_ALIAS) for Siebel component groups that the Agent for Siebel Server must monitor. These components must be enabled in the Siebel Enterprise Server. You must specify this attribute if you have specified the Sadmin, SadminCrPasswd, and SecondLevelMonitor attributes. Example 1: **CallCenter, Sales, SiebAnywhere, Remote** Example 2: **EAI,System** Default: ““

Optional attribute	Description
MonitorProgram *String*	Absolute path name of an external, user-supplied monitor executable. For information about setting this attribute: See "Executing a custom monitor utility" on page 24. Example 1: **`/siebel/v78/srv1/siebsrvr/chk_gendb.sh`** Example 2: **`/siebel/v78/srv1/siebsrvr/comm_inbound.pl arg1 arg2`** Default: ""
Sadmin *String*	The administrative login that is defined for the Siebel Enterprise. You must use this login if you want to perform component level monitoring during second level check for the Siebel Enterprise Server. Default: `sadmin`
SadminCrPasswd *String*	Password for the administrator specified in the Sadmin attribute. The password is encrypted using the VCS encrypt utility, `vcsencrypt(1m)`. You must specify this attribute if you have specified the CompGrps, Sadmin, and SecondLevelMonitor attributes. Example: **`EshQfqIqrQnqS`** Default: ""
SecondLevelMonitor *Integer*	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the Siebel Server. 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, and so on. **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. Example: **`5`** Default: `0`

Executing a custom monitor utility

You can configure the monitor operation to execute a custom monitor utility to perform a user-defined Siebel Server state check. The utility is executed in the context of the UNIX user that is defined in the SiebelUser attribute. The environment is set by sourcing the file specified in the EnvFile attribute.

The monitor operation executes MonitorProgram if:

- The MonitorProgram attribute value is set to a valid executable utility.
- The first level process check indicates that the Siebel Server instance is online.
- The SecondLevelMonitor attribute is set to 1 and the second level check returns the server state as ONLINE,
  or the SecondLevelMonitor attribute is set to greater than 1, but the second level check is deferred for this monitoring cycle.

The monitor operation interprets the program exit code as follows:

Exit code	Interpretation
110 or 0	Siebel Server instance is ONLINE
100 or 1	Siebel Server instance is OFFLINE
99	Siebel Server instance is UNKNOWN
Any other value	Siebel Server instance is UNKNOWN

To ensure that the custom monitor utility is always available to the agent application, Symantec recommends storing the file in the directory that the SiebelRoot attribute specifies on the shared storage device.

Configuring Siebel resources for Solaris zone support

To enable the Agent for Siebel Server to support Solaris 10 zones, Symantec recommends these configuration steps:

- Install each Siebel Server on a dedicated Solaris zone.
- Symantec recommends installing zones on the shared disk for convenient configuration, failover, and maintenance.
- The name of the Solaris zone must be the same as the virtual hostname that you use to install and configure the Siebel Server.

For sample Service Groups that depict Solaris zone support:

See “Sample Service Group configurations for Solaris zone support” on page 51.

Chapter 4

Clustering Siebel Servers

You can cluster Siebel Server instances in a clustered environment, and you can use the Agent for Siebel Server to manage these entities.

The following sections cover the entities that you must configure for a Siebel Server instance to function in a clustered environment and the recommended directory structures for the Siebel resources in a Siebel enterprise. The chapter includes:

- "Siebel entities in a clustered environment" on page 28
- "Installing the Siebel Server instance" on page 29
- "Setting Siebel Server parameters after installation" on page 31
- "Configuring unique port numbers for Siebel remote servers" on page 33

Siebel entities in a clustered environment

A Service Group is a logical setup containing all resources that can support a Siebel Server instance in a clustered environment. The required resources are described as below.

- *Disk Group*

  A disk group contains a volume and a file system, which is a mount resource containing the Siebel Server installation files.

  Use the **`DiskGroup`** resource type to create this resource. Create the disk group from the shared disk so that you can import the group into any system in the cluster.

- *Mount*

  The mount resource mounts, monitors, and unmounts the file system that is dedicated to the Siebel Server installation files.

  Use the **`Mount`** resource type to create this resource.

- *Network Interface*

  The Network Interface resource monitors the network interface card through which the Siebel Server instance communicates with other services.

  Use the **`NIC`** resource type to create this resource.

- *Virtual IP*

  The Virtual IP resource configures the virtual IP address dedicated to the Siebel Server instance. The external services, programs, and clients use this address to communicate with this instance.

  Use the **`IP`** resource type to create this resource.

- *NFS Mount*

  If the Siebel File System is shared among all the Siebel Server instances using NFS services, the file system must be NFS mounted on each node in the cluster. The file system must also be configured for the Siebel Enterprise Server.

  Placing the remote mount under cluster control ensures effective communication among the resources that are required to bring the Siebel Server instance online.

  Use the **`VProNFSMount`** resource type to create this resource.

  Configuring an NFS Mount resource is optional.

- *Siebel Server*

  The Siebel Server resource starts, stops, and monitors the Siebel Server instance.

  Use the **`Siebel`** resource type to create this resource.

Installing the Siebel Server instance

While installing a Siebel Server instance, ensure that you take care of these tasks:

- "Recommended directory structure"
- "Specifying virtual Siebel Server names during installation"
- "Installing the Database client"
- "Using the Siebel installation properties file"

Recommended directory structure

Typically, a Siebel enterprise consists of these core application services:

- a Siebel Gateway Server
- one or more Siebel Application Servers
- a Siebel File System
- a database server
- one or more Web servers

Symantec recommends a directory structure for these services for a simplified cluster configuration.

Note: This example shows directory structure for two Siebel Servers and two Web servers. But the naming structure supports an unlimited number of each type of servers.

- A well designed directory structure also creates a storage environment that is more intuitive and easier to manage. Symantec recommends a directory structure as described below.

Recommended directory structure	Description
`/siebel/v78/gtw`	The mount point of the Siebel Gateway Server.
`/siebel/v78/srv1`	The mount point of the first Siebel Server.
`/siebel/v78/srv1/ora92`	Location for the database client for the first Siebel Server.
`/siebel/v78/srv2`	The mount point of the second Siebel Server.

Recommended directory structure	Description
`/siebel/v78/srv2/ora92`	Location for the database client for the second Siebel Server.
`/siebel/v78/web1`	The mount point of the first web server that is required for SWSE (Siebel Web Extension).
`/siebel/v78/web2`	If needed, the mount point of the second web server that is required for SWSE.

- If all the Siebel Servers use a common enterprise level Siebel file system, you can create the `/siebel/v78/sfs78` directory on all the configured nodes in the cluster.
  This directory is the mount point for the Siebel file system. This file system can be NFS mounted on all nodes in the cluster, that are configured for the Siebel Enterprise Servers. This file system can be configured as a parallel Service Group using the VProNFSMount resource.
- If the Siebel Servers use dedicated file systems, then create dedicated mount points for each Siebel file system. For example, `/siebel/v78/sfs78_srv1`, `/siebel/v78/sfs78_srv2`, and so on.
  Ensure that you create VProNFSMount resources for each filesystem. These resources can be a part of the failover Service Group that is configured for the respective Siebel Server.

Specifying virtual Siebel Server names during installation

While installing a Siebel Server instance, the program prompts you for a logical name to assign to the Siebel Server. Symantec recommends assigning a name that does not include the host name of the system in the cluster. Therefore, the name of a Siebel Server must not imply the system in the cluster on which the instance is allowed to run.

Installing the Database client

While installing the Siebel Server, you must also install the appropriate database client software, for example, Oracle client. Symantec recommends installing the database client on the file system that is dedicated to the program and data files of the Siebel Server. This ensures that the Siebel Server instance can access the client, even when the Service Group switches among the nodes in the cluster.

Using the Siebel installation properties file

The Siebel EBusiness Applications, version 7.5 introduces the use of InstallShield and a Configuration GUI for installation and configuration of the Siebel application on UNIX platforms. During server installation, the installer creates a `vpd.properties` file that records installation information. This file is located in `/var/adm/Siebel` directory on the system in which the server was installed.

Note: This is an internal system directory, and not the SiebelRoot directory that is located on shared disk.

This file is not referenced for normal server operations such as startup or shutdown. But the installer may use data from this file to perform other tasks, such as checking versions, applying patches, and adding language packs.

Therefore, when you perform software upgrades to a Siebel server, ensure that you switch the Siebel Service Group to the system on which the Group was originally installed, so that the installer is able to access the `vpd.properties` file. Refer to the Siebel product documentation for more information about this file.

Setting Siebel Server parameters after installation

After installing a Siebel Server, you must set some parameters so that you can cluster the instance. Follow these steps to configure the parameters.

- "Setting the host address parameter for a Siebel Server"
- "Setting the host parameter for a Siebel Server"
- "Configuring the Siebel Server file system"
- "Setting the SIEBEL_GATEWAY environment variable"
- "Disabling the autostart option"

Setting the host address parameter for a Siebel Server

Siebel versions 7.5 and above introduce a new parameter, *ServerHostAddress*. This parameter specifies the virtual host name of a Siebel Server instance. After installing each Siebel Server, perform these steps to set the ServerHostAddress parameter:

1. Log in to the Siebel Server Manager utility, `srvrmgr`.
2. Change the value of the ServerHostAddress parameter:

```
srvrmgr> change param ServerHostAddress=<HostName> for server
<ServerName>
```

Setting the host parameter for a Siebel Server

For Siebel versions 7.5 and above, you must set the *Host* parameter to enable a Siebel Server instance to run on any system in the cluster. You must set the parameter as equal to the virtual IP host name assigned to the instance. After installing each Siebel Server, perform these steps to set the ServerHostAddress parameter:

1. Log in to the Siebel Server Manager utility, `srvrmgr`.
2. Change the value of the Host parameter:

```
srvrmgr> change param Host=<HostName> for server <ServerName>
```

Configuring the Siebel Server file system

Each Siebel Server can either have a dedicated file system, or all Siebel Servers can use a single filesystem created in the Siebel Enterprise.

For details about this configuration, refer to the Siebel documentation or contact your Siebel administrator.

Setting the SIEBEL_GATEWAY environment variable

The Siebel Server installation program generates two Unix scripts. The Siebel administrators typically use one of the two scripts to set the environment prior to starting or stopping a Siebel Server instance.

The `siebenv.sh` script sets the environment for a Bourne or Korn shell. The `siebenv.csh` script sets the environment for a C shell. Normally, one of these two scripts is specified in EnvFile attribute. In both scripts, ensure that the SIEBEL_GATEWAY environment variable is set to the virtual host name of the Siebel Gateway Server. This variable should not be set to the IP address or the host name of the system.

Disabling the autostart option

Ensure that you disable the *autostart* option while configuring the Siebel Server instance for clustering.

For details about the autostart option, refer to the Siebel documentation or contact your Siebel administrator.

Configuring unique port numbers for Siebel remote servers

The Siebel Remote component group provides data synchronization support between Siebel Mobile Web Clients and the Siebel Database Server. To perform synchronization, Siebel Mobile Web Client users must be able to connect to a Siebel Remote Server using TCP/IP.

By default, Remote Servers listen for client requests on the `40400` port. If multiple Remote Servers are deployed, and if these Servers are configured within the cluster such that two or more Servers may run simultaneously on the same system, ensure that you configure each Remote Server to listen on a unique port.

Such an arrangement is necessary to avoid port number conflicts, since a Remote Server listens on its configured port on all IP addresses active on the system. Therefore the Server listens promiscuously on its port for all active IP addresses. Refer to the Siebel documentation for instructions to configure a port number for a Remote Server.

Uninstalling the Veritas Agent for Siebel Server

Follow the steps below to remove the Veritas High Availability Agent for Siebel Server from a cluster. These steps must be performed while the cluster is active.

To uninstall the agent in a VCS environment

1. Log in as `root`.
2. Set the cluster configuration mode to read/write by typing the following command from any system in the cluster:

   ```
   # haconf -makerw
   ```
3. Remove all Siebel Server resources from the cluster. Use the following command to verify that all resources have been removed.

   ```
   # hares -list Type=Siebel
   ```
4. Remove the agent from the cluster configuration by typing the following command from any system in the cluster.

   ```
   # hatype -delete Siebel
   ```

   Removing the type file of the agent from the cluster removes the include statement for the agent from the `main.cf` file, but the type file is not removed from the cluster configuration directory. You can remove the type file later, from the cluster configuration directory.
5. Save these changes. Then set the cluster configuration mode to read-only by typing the following command from any system in the cluster:

   ```
   # haconf -dump -makero
   ```
6. Use the platform's native software management program to remove the Agent for Siebel Server from each node in the cluster:

   ```
   # pkgrm VRTSsiebel
   ```

Chapter 6

Troubleshooting the Veritas Agent for Siebel Server

This chapter covers tips and pointers on using the Agent for Siebel Server, with Veritas high availability products. To resolve issues effectively, follow the steps in the order presented below. You may come across unique issues, but make sure that you follow these steps in the presented order to avoid unnecessary issues.

These troubleshooting tips and pointers are applicable to the agent while working in a VCS environment.

The chapter includes:

Using correct software and operating system versions

Ensure that no issues arise due to incorrect software and operating system versions. For the correct versions of operating system and software to be installed on the resource systems:

See "Supported software" on page 9.

Meeting prerequisites

Before installing the Agent for Siebel Server, double check that you meet the prerequiste requirements. For example, you must install the ACC library on VCS before installing the agent. For a list of prerequisites:

See "Prerequisites for installing the Agent for Siebel Server" on page 16.

Configuring Siebel Server resources

Before using an Siebel Server resource, ensure that you configure the resource properly. For a list of resource types with which you can configure all Siebel Server resources:

See "Agent attributes" on page 20.

Configuration checks for Solaris zone support

If you have configured the Agent for Siebel Server to support Solaris zones, ensure that you have followed all the configuration steps:

- Prerequisites for enabling Solaris zone support:
  See "Prerequisites for installing the agent for Solaris zone support" on page 16.
- Importing the `types.cf` file for Solaris zone support:
  See "Importing the agent types files" on page 20.
- Configuring the SAP resources for Solaris zone support:
  See "Configuring Siebel resources for Solaris zone support" on page 25.

Starting the Siebel Server outside a cluster server

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 Siebel Server independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the Siebel Server outside the cluster framework.

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

A sample procedure to start a Siebel Server outside the cluster framework, is illustrated below.

To start the Siebel Gateway Server instance outside the framework

1 Start the Siebel Gateway Server instance using these commands:

```
# su <SiebelUser>
$ . <EnvFile>
$ <SiebelRoot>/bin/start_ns -p <SiebelGWPort>
```

2 Execute this command to check whether the `siebsvc` process of the Siebel Gateway Server instance is present in the processes table of the system:

```
$ /bin/ps -ef | grep siebsvc | grep -v grep
```

For example:

```
<SiebelUser> 23804 25691   0 12:03:36 ?          1:37 siebsvc -s
gtwyns -a /f <SiebelRoot>/sys/siebns.dat /t <SiebelGWPort> /t
<SiebelGWPort>
```

If this line appears in the processes table, the `siebsvc` process has started. The user specified in the SiebelUser attribute owns this process, and the repository file is the `<SiebelRoot>/sys/siebns.dat` file.

To stop the Siebel Gateway Server instance outside the framework

If the Siebel Gateway Server instance starts successfully, attempt to shut down the instance using these commands:

```
# su <SiebelUser>
$ . <EnvFile>
$ <SiebelRoot>/bin/stop_ns
```

If you are able to successfully stop a Siebel Gateway Server instance outside the cluster framework, attempt to use the instance inside the framework.

To start the Siebel Enterprise Server outside the framework

1 Execute these commands to start the Siebel Enterprise Server:

```
# su <SiebelUser>
% source <EnvFile>
% <SiebelRoot>/bin/start_server -e <SiebelEnterprise> \
-g <SiebelGWHost>:<SiebelGWPort> <ServerName>
```

2 Execute this command to check whether the `siebsvc` process of the Siebel Enterprise Server instance is present in the processes table of the system:

```
$ /bin/ps -ef | grep siebsvc | grep -v grep
```

For example:

```
<SeibelUser> 20783 20497   0 11:55:22 ?          0:21 siebsvc -s
siebsrvr -a /g <ServerName> /e <SiebelEnterprise> /s
<ServerName> /g <SiebelGWHost>:<SiebelGWPort>
```

If this line appears in the proc table, the `siebsvc` process of the Siebel Enterprise Server instance has started properly. The process is for the *<ServerName>* Siebel Server, belongs to the *<SiebelEnterprise>* Siebel Enterprise, and is configured with the Siebel Gateway Server that is running on the *<SiebelGWHost>* virtual host, which is listening on the *<SiebelGWPort>* port.

3 Attempt to query the database to ensure that the Siebel Enterprise Server is able to reach the data stored in the database.

For example, you can execute this command as the table owner querying the database:

```
% <SiebelRoot>/bin/odbcsql /u siebel /p siebel \
/s siebsrvr_<SiebelEnterprise>
```

If you receive the message, `Outstanding transaction committed.`, the Siebel Enterprise Server is able to reach the database.

You can also consider these options:

- Ensure that you are able to connect to the database using the database supplied client utilities. For example, for an Oracle database, you can use the `tnsping` utility to connect to the listener of the database.
- Ensure that the environment variables required for connecting to the database are set correctly. Contact the database administrator for more information.

4 Attempt to check if the component groups specified in the CompGrps attribute are up and running.

For example:

a Run this command to connect to the Siebel Server:

```
% <SiebelRoot>/bin/srvrmgr /u <Sadmin> /p <Sadmin> \
/e <SiebelEnterprise> /g <SiebelGWHost> /q \
/s <ServerName>
```

b Run this command to list the component groups with their states:

```
srvrmgr:<ServerName>> list compgrp show CG_ALIAS,
CA_RUN_STATE

CG_ALIAS      CA_RUN_STATE

------------  ------------

SiebAnywhere  Online
```

```
CallCenter      Online
Remote          Running
Sales           Online
System          Running
5 rows returned.
```

c Check the status of the components in the `CallCenter` component group:

```
srvrmgr:<ServerName>> list comp for compgrp
CallCenter show CC_ALIAS, CG_ALIAS, CP_DISP_RUN_STATE

CC_ALIAS            CG_ALIAS    CP_DISP_RUN_STATE
------------------  ----------  -----------------
SCCObjMgr_enu       CallCenter  Online
eServiceObjMgr_enu  CallCenter  Online

2 rows returned.
```

d Quit the connection:

```
srvrmgr:<ServerName>> quit
```

5 Check if the value of the SadminCrPasswd attribute is set correctly.

To stop the Siebel Enterprise Server outside the framework

Stop the Siebel Enterprise Server using these commands:

```
# su <ServerName>
% source <EnvFile>
% <SiebelRoot>/bin/stop_server -e <SiebelEnterprise> -M <ServerName>
```

If you are able to successfully start and stop the Siebel Enterprise Server outside the framework, attempt to use the Server inside the cluster framework.

Reviewing log files

If you are facing problems while using the Agent for Siebel Server or an Siebel Server, refer to the following sections to access the relevant files for information about the issue.

Using VCS log files

In case of problems while using the Agent for Siebel Server, you can also access the VCS engine log file for more information about a particular resource. The VCS engine log file is `/var/VRTSvcs/log/engine_A.log`.

Using Siebel Server log files

If a Siebel Server is facing problems, you can access the server log files to further diagnose the problem.

- For a Siebel Enterprise Server, the log files are located in the `<SiebelRoot>/log` directory and the `<SiebelRoot>/enterprises/<SiebelEnterprise>/<ServerName>/log` directory.
- For a Siebel Gateway Server, the log files are located in the `<SiebelRoot>/sys/log` directory. You can look for the `NameSrvr.log` and `siebel.log` files.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a VCS log file for each Siebel Server 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 is produced. Symantec recommends that you must localize the ResLogLevel attribute for particular resource.

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. Test the identified resource. The operation reproduces the problem that you are attempting to diagnose.
5. Set the ResLogLevel attribute back to **INFO** for the identified resource:
   ```
   # hares -modify Resource_Name ResLogLevel INFO -sys SysA
   ```
6. Review the contents of the VCS engine output log file.

You may also contact Symantec support for more help.

Appendix A

Sample Configurations

This chapter covers samples of agent type definition while working with a cluster. This chapter also covers sample service group configuration for the Siebel Server while working in a cluster.

The chapter includes:

Sample agent type definition

A sample `main.cf` file follows.

```
include "types.cf"
include "SiebelTypes.cf"

cluster siebel78sol (
    UserNames = { admin = aHIaHChEIdIIgQIcHF }
    Administrators = { admin }
    CredRenewFrequency = 0
    CounterInterval = 5
    )

system nodeA (
    )

system nodeB (
    )


group sbl78gtw (
    SystemList = { nodeA = 0, nodeB = 1 }
    AutoStartList = { nodeA }
    )

    DiskGroup sbl78gtw_dg (
        Critical = 0
        DiskGroup = sbl78gtwsol
        )

    IP sbl78gtw_ip (
        Critical = 0
        Device = bge0
        Address = "10.212.98.244"
        NetMask = "255.255.254.0"
        )

    Mount sbl78gtw_mnt (
        Critical = 0
        MountPoint = "/siebel/v78/gtw"
        BlockDevice = "/dev/vx/dsk/sbl78gtwsol/siebel"
        FSType = vxfs
        FsckOpt = "-y"
        )

    NIC sbl78gtw_nic (
        Critical = 0
        Device = bge1
        NetworkType = ether
        )
    Siebel sbl78gtw_srvr (
        EnvFile = "/siebel/v78/gtw/gtwysrvr/envfile.csh"
        HostName = sbl78gtwsol
        SiebelRoot = "/siebel/v78/gtw/gtwysrvr"
        SiebelUser = sbl78gtw
        SiebelGWHost = sbl78gtwsol
        ServerType = GTWY
        SiebelEnterprise = sbl78sol
        SecondLevelMonitor = 1
        )

    sbl78gtw_mnt requires sbl78gtw_dg
    sbl78gtw_ip requires sbl78gtw_nic
    sbl78gtw_srvr requires sbl78gtw_ip
    sbl78gtw_srvr requires sbl78gtw_mnt


    // resource dependency tree
    //
    //  group sbl78gtw
    //  {
    //  Siebel sbl78gtw_srvr
    //      {
    //      Mount sbl78gtw_mnt
    //          {
    //          DiskGroup sbl78gtw_dg
    //          }
    //      IP sbl78gtw_ip
    //      }
    //  }


group sbl78srv1 (
    SystemList = { nodeA = 0, nodeB = 1 }
    AutoStartList = { nodeB }
    )

    DiskGroup sbl78srv1_dg (
        Critical = 0
        DiskGroup = sbl78srv1sol
        )

    IP sbl78srv1_ip (
        Critical = 0
        Device = bge0
        Address = "10.212.98.240"
        NetMask = "255.255.254.0"
        )

    Mount sbl78srv1_mnt (
        Critical = 0
        MountPoint = "/siebel/v78/srv1"
    BlockDevice = "/dev/vx/dsk/sbl78srv1sol/siebel"
    FSType = vxfs
    FsckOpt = "-y"
    )

NIC sbl78srv1_nic (
    Critical = 0
    Device = bge2
    NetworkType = ether
)

Siebel sbl78srv1_srvr (
    EnvFile = "/siebel/v78/srv1/siebsrvr/envfile.csh"
    HostName = sbl78sv1sol
    SiebelRoot = "/siebel/v78/srv1/siebsrvr"
    SiebelUser = sbl78sv1
    SadminCrPasswd = ftiRgrJrsRorT
    SiebelGWHost = sbl78gtwsol
    ServerName = siebsrv1
    SiebelEnterprise = sbl78sol
    CompGrps = "CallCenter, Remote, Sales, SiebAnywhere, System"
    SecondLevelMonitor = 1
    )

requires group sbl78gtw online global soft
sbl78srv1_mnt requires sbl78srv1_dg
sbl78srv1_ip requires sbl78srv1_nic
sbl78srv1_srvr requires sbl78srv1_ip
sbl78srv1_srvr requires sbl78srv1_mnt


// resource dependency tree
//
//  group sbl78srv1
//  {
//  Siebel sbl78srv1_srvr
//      {
//      IP sbl78srv1_ip
//      Mount sbl78srv1_mnt
//          {
//          DiskGroup sbl78srv1_dg
//          }
//      }
//  }
```

Sample configuration file

While working in the VCS 4.x environment

```
type Siebel (
       static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
HostName, SiebelRoot, SiebelUser, Sadmin, SadminCrPasswd,
SiebelGWHost, SiebelGWPort, ServerName, ServerType,
SiebelEnterprise, CompGrps, MonitorProgram, SecondLevelMonitor }
        str ResLogLevel = INFO
        str EnvFile
        str HostName
        str SiebelRoot
        str SiebelUser
        str Sadmin = sadmin
        str SadminCrPasswd
        str SiebelGWHost
        int SiebelGWPort = 2320
        str ServerName
        str ServerType = SRVR
        str SiebelEnterprise
        str CompGrps
        str MonitorProgram
        int SecondLevelMonitor
)
```

While working in the VCS 5.0 environment

```
type Siebel (
    static str AgentFile = "/opt/VRTSvcs/bin/ScriptAgent"
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Siebel"
    static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
HostName, SiebelRoot, SiebelUser, Sadmin, SadminCrPasswd,
SiebelGWHost, SiebelGWPort, ServerName, ServerType,
SiebelEnterprise, CompGrps, MonitorProgram, SecondLevelMonitor }
    str ResLogLevel = INFO
    str EnvFile
    str HostName
    str SiebelRoot
    str SiebelUser
    str Sadmin = sadmin
    str SadminCrPasswd
    str SiebelGWHost
    int SiebelGWPort = 2320
    str ServerName
    str ServerType = SRVR
    str SiebelEnterprise
    str CompGrps
    str MonitorProgram
    int SecondLevelMonitor
)
```

While working in the VCS 5.0 environment with Solaris zone support

```
type Siebel (
    static str ContainerType = Zone
    static str AgentFile = "/opt/VRTSvcs/bin/ScriptAgent"
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Siebel"
    static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
HostName, SiebelRoot, SiebelUser, Sadmin, SadminCrPasswd,
SiebelGWHost, SiebelGWPort, ServerName, ServerType,
SiebelEnterprise, CompGrps, MonitorProgram, SecondLevelMonitor }
    str ResLogLevel = INFO
    str EnvFile
    str HostName
    str SiebelRoot
    str SiebelUser
    str Sadmin = sadmin
    str SadminCrPasswd
    str SiebelGWHost
    int SiebelGWPort = 2320
    str ServerName
    str ServerType = SRVR
    str SiebelEnterprise
    str CompGrps
    str MonitorProgram
    int SecondLevelMonitor
    str ContainerName
)
```

Sample Service Group configurations

This section includes sample Service Groups configurations in a VCS environment.

Figure A-1 shows a Service Group with a Siebel Enterprise Server instance running in a VCS environment. The Service Group also includes a DiskGroup resource, a NIC resource, and a Mount resource.

Figure A-1 Sample Service Group for a Siebel Enterprise Server instance

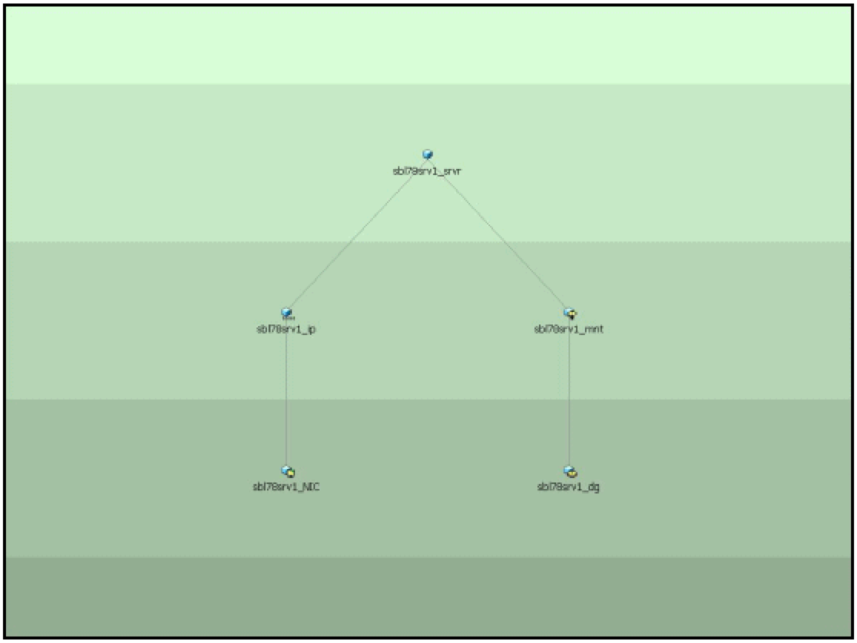
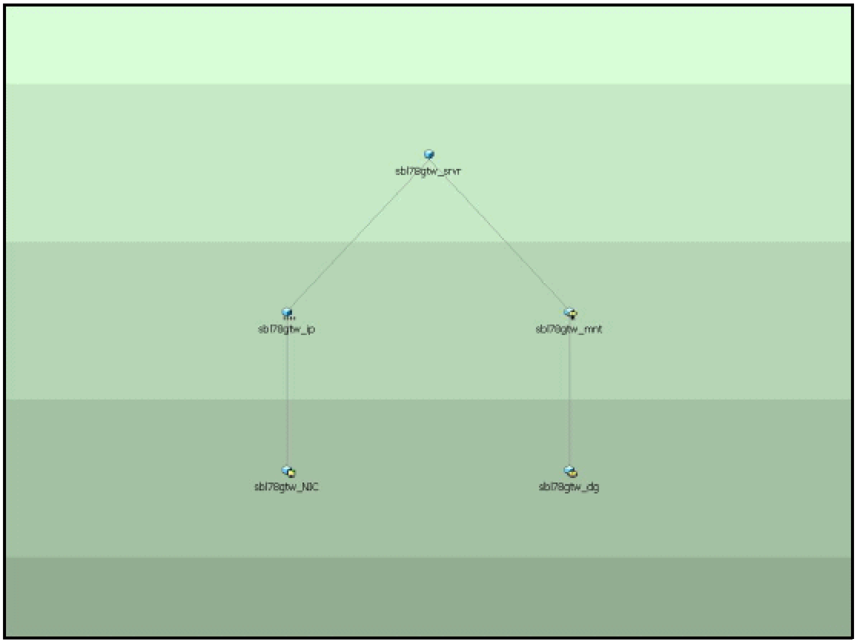


Figure A-2 shows a Service Group with a Siebel Gateway Server instance running in a VCS environment. The Service Group also includes a DiskGroup resource, a NIC resource, and a Mount resource.

Figure A-2 Sample Service Group for a Siebel Gateway Server instance



Sample Service Group configurations for Solaris zone support

This section includes sample Service Groups with Solaris zone support.

Figure A-3 shows a Service Group with a Siebel Enterprise Server instance running in a local zone, if the zone binaries are present on the shared disk. The Service Group also includes a DiskGroup resource, a NIC resource, and a Mount resource.

Figure A-3 Sample Service Group with a Siebel Enterprise Server instance

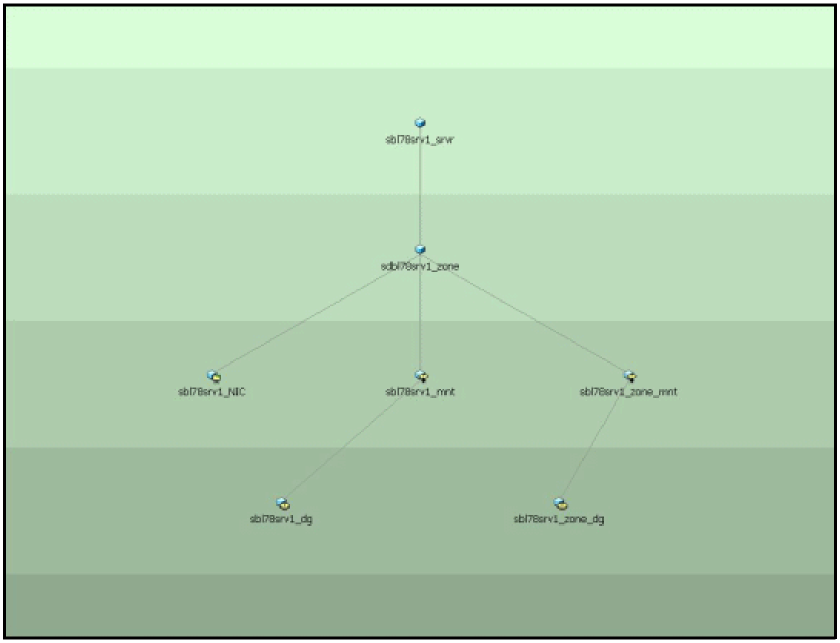
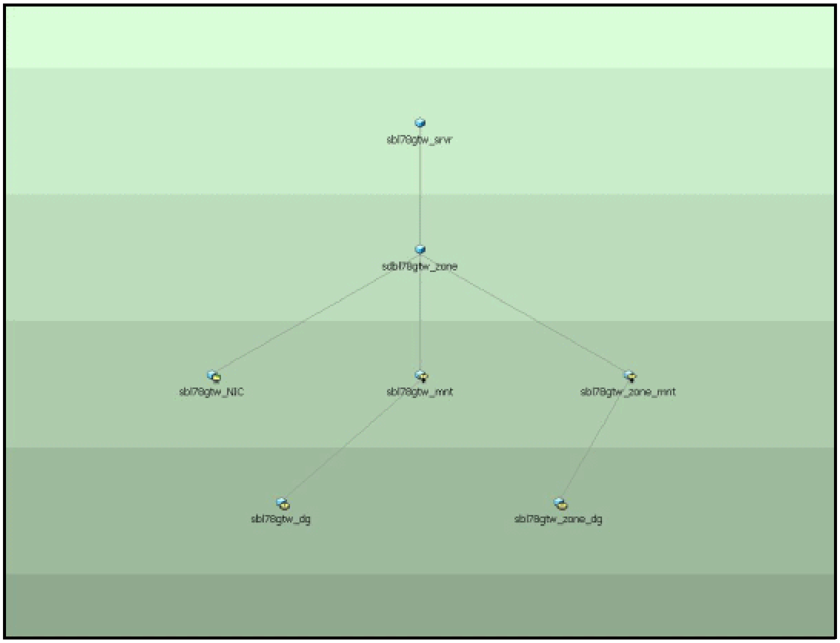


Figure A-4 shows a Service Group with a Siebel Gateway Server instance running in a local zone, if the zone binaries are present on the shared disk. The Service Group also includes a DiskGroup resource, a NIC resource, and a Mount resource.

Figure A-4 Sample Service Group with a Siebel Gateway Server instance



Index