# Cluster Server Agent for IBM Informix Dynamic Server Installation and Configuration Guide

AIX, Linux, Solaris

7.0

**VERITAS**™

# Veritas InfoScale™ Availability Agents

Last updated: 2020-04-07

## Legal Notice

## Technical Support

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

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

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

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

## Documentation

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

https://sort.veritas.com/documents

## Documentation feedback

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

infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

## Veritas Services and Operations Readiness Tools (SORT)

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

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

# Contents

**Chapter 6**      **Configuring the service groups for IBM Informix Dynamic Server using the CLI** ................................ 54

**Chapter 7**      **Troubleshooting the agent for IBM Informix Dynamic Server** .......................................... 61

# Introducing the agent for IBM Informix Dynamic Server

This chapter includes the following topics:

- About the Cluster Server agent for IBM Informix Dynamic Server

- Supported software

- Features of the agent

- How the agent makes IBM Informix Dynamic Server highly available

- How the agent makes Informix HDR highly available

- How the agent makes SDS highly available

- Informix Dynamic Server agent functions

- Detecting HDR Failure

- Setting up IBM Informix Dynamic Server in a VCS cluster

## About the Cluster Server agent for IBM Informix Dynamic Server

The Cluster Server (VCS) agent for Informix Dynamic Server provides high availability for all Informix Dynamic Servers in a cluster.

For the latest updates or software issues for this agent, see the *Cluster Server Agent Pack Release Notes*.

# Supported software

For information on the software versions that the Cluster Server agent for IBM Informix Dynamic Server supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

# Features of the agent

The following are the features of the Cluster Server agent for IBM Informix Dynamic Server:

- Support for validation of attributes that are based on the agent functions
  The agent can validate attributes in each agent function before the actual data processing starts.

- Support for First Failure Data Capture (FFDC)
  In case of a fault, the agent generates a huge volume of the debug logs that enable troubleshooting of the fault.

- Support for Fast First Level Monitor (FFLM)
  The agent maintains PID files based on search patterns to expedite the monitoring process.

- Support for external user-supplied monitor utilities
  The agent enables user-specified monitor utilities to be plugged in, in addition to the built-in monitoring logic. This enables administrators to completely customize the monitoring of the application.

- Support for intelligent resource monitoring and poll-based monitoring
  The agent supports the Cluster Server Intelligent Monitoring Framework (IMF) feature. IMF allows the agent to register the resources to be monitored with the IMF notification module so as to receive immediate notification of resource state changes without having to periodically poll the resources.
  See "About Intelligent Monitoring Framework" on page 47.

- Delayed agent function
  The agent manages the first monitor after online for slow initializing applications.

# How the agent makes IBM Informix Dynamic Server highly available

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 Informix instance 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 Informix instance. The agent detects application failure if the monitoring routine reports an improper function of the Informix instance processes. When this application failure occurs, the Informix instance service group fails over to another node in the cluster. Thus, the agent ensures high availability for Informix instances.

## High availability for IBM Informix Dynamic Server instances running in Solaris zones

Solaris provides a means of virtualizing operating system services, allowing one or more processes to run in isolation from other activity on the system. Such a 'sandbox' is called a 'non-global zone'. Each zone can provide a rich and customized set of services. The processes that run in a 'global zone' have the same set of privileges that are available on a Solaris system today.

VCS provides high availability to applications running in non-global zones by extending the failover capability to zones. VCS is installed in a global zone, and all the agents and the engine components run in the global zone. For applications running within non-global zones, agents run script entry points inside the zones. If a zone configured under VCS control faults, VCS fails over the entire service group containing the zone.

For more details, refer to the *Cluster Server Administrator's Guide*.

The Cluster Server agent for IBM Informix Dynamic Server is zone-aware and can monitor Informix Server instances running in non-global zones.

# How the agent makes Informix HDR highly available

When you enable the agent attribute SecondLevelMonitor, the agent runs the `onstat – ` command to determine the state of the Informix Database Server.

Following exit codes highlight if the IDS is up and functional:

| IDS Type | Exit Code | State |
|----------|-----------|-------|

| | | |
|---|---|---|
| ■ Standard database server (without HDR) | ■ 5 | Online |
| ■ Primary database server | ■ 5 | |
| ■ Secondary database server | ■ 2 | |
| ■ Standard database server (without HDR) | 225 | Offline |
| ■ Primary database server | | |
| ■ Secondary database server | | |
| ■ Standard database server (without HDR) | Any other | Unknown |
| ■ Primary database server | | |
| ■ Secondary database server | | |

## High availability for Informix HDR

The Cluster Server agent for Informix monitors the Informix Database Server instances running as Primary and Secondary Database Servers, if the HDR is configured in manual mode (DR_AUTO is (0)OFF).

When an Informix Database Server fails, the cluster fails over this instance to the node configured as the failover node. It then brings up this failover node as Primary or Secondary node, based on its original configuration. During this failover the agent does not change the membership type of the Database server.

# How the agent makes SDS highly available

When the Informix Database server is configured as Shared Disk (SD), the agent uses SDSInstances attribute to access the next primary shared disk. Thus, the secondary shared disk now becomes the Primary shared disk. The secondary instances now identify this disk as the primary disk.

Whenever the Primary SD Server fails, a preonline trigger gets called on the new target system. This trigger finds out the first ONLINE Secondary SD and executes the action entry point which promotes it to Primary SD, using the command:

```
# onmode -d set SDS primary SDS Instance Name
```

After the secondary shared disk is successfully converted into the primary shared disk, the preonline trigger failovers the faulted node to a new target and this instance is started as Secondary shared disk. This disk automatically gets connected to the previously promoted Primary shared disk as its updater node alias. You can confirm the same using the following command.

```
# onstat -g sds verbose
```

# Informix Dynamic Server agent functions

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

## Online

The online operation performs the following tasks:

- Verifies that the required attributes are set correctly.

- Verifies whether the Informix Dynamic Server instance is not already online. If the instance is online, the online operation exits immediately.

- Executes the following command, in context of the user 'informix', to start the Informix Dynamic Server instance:

  ```
  $ InformixDir/bin/oninit -yw 1> /dev/null 2>&1
  ```

- Starts the Informix Dynamic Server in system.slice, if the UseSystemD attribute is set on the supported Linux platforms.

- Verifies whether the Informix Dynamic Server is completely functional.

- Returns the control to HAD.

## Offline

The offline operation performs the following tasks:

- Verifies that the required attributes are set correctly.

- Verifies that the Informix Dynamic Server instance is not offline. If the instance is already offline, the operation exits immediately.

- Executes the following command, in context of the user 'informix' to stop the Informix Dynamic Server instance.

  ```
  $ InformixDir/bin/onshutdown.sh 60
  ```

- Executes the Onclean utility is to remove any remnants of shared memory and to stop database virtual processes, if any.

- Performs the following actions, if the execution of the offline script fails:

  - Kills any existing processes that belong to this instance of Informix Dynamic Server.

  - Removes any shared memory resources associated with the existing Informix Dynamic Server instance.

- Returns the control to HAD.

- If the instance is configured as SharedDisk, then after successful completion of the Offline function the agent checks if the offline instance was SD Primary. If yes, it then promotes one of the SD Secondary to SD Primary so that other SDS nodes do not fail.

## Monitor

The monitor operation monitors the states of the Informix Dynamic Servers on all nodes within the cluster.

The operation performs the following tasks:

- Conducts a first level check to determine that the Informix Dynamic Server processes are running on the system in the cluster. If the first level check does not find these processes running on the node, the check exits immediately, and reports the instance as offline.

- Conducts a second level check if the SecondLevelMonitor attribute is set to a value greater than 0.
  During this check agent executes the following command in context of the user 'informix' to verify that the Informix Dynamic Server is completely functional.

  $ *InformixDir*/bin/onstat -

- Depending upon the MonitorProgram attribute, the monitor operation performs a customized check using a user-supplied monitoring utility.
  See "Executing a customized monitoring program" on page 45.

## Clean

The clean operation performs the following tasks in the event of a failure or an unsuccessful attempt to bring an Informix Dynamic Server instance online or take it offline:

- Attempts to gracefully shut down the Informix Dynamic Server instance, using the following command in context of the user 'informix'.

  $ *InformixDir*/bin/onmode -ky

- Executes the Onclean utility is to remove any remnants of shared memory and to stop database virtual processes, if any.

- Performs the following actions, if the instance does not shut down normally:

  - Kills the remaining processes pertaining to this instance of the Informix Dynamic Server.

  - Removes any shared memory resources associated with the existing Informix Dynamic Server instance.

■ Returns the control to HAD.

## Action

The Action operation enables you to perform the following predefined actions on a resource.

| Action | Description |
| --- | --- |
| switchprim | Promotes the Informix SD Secondary Server to SD Primary Server. This action expects the name of the current/failed SD Primary Informix Instance as actionargs. |
| | This action is executed by preonline trigger in case of SD Primary failure. |
| | For example, |
| | `# hares -action infx_inst2_res switchprim -actionargs infx_inst1 -sys infx_sys1` |
| isprimary | Retrieves the Shared disk configuration of current informix instance i.e. it is configured as SD Primary or SD Secondary. |
| | For example, |
| | `# hares -action infx_inst2_res isprimary -sys infx_sys1` |

## Preonline

The preonline operation performs the following tasks:

■ Facilitates proper Informix server failover behaviour if it is configured as a SharedDisk.

■ During a MANUAL switch-over, the agent checks if the current instances are SD Primary. If yes, the agent promotes one of the online SD Secondary to SD Primary before switching the instances to a new node. The switched-over node now becomes the new SD Secondary and automatically connects to SD Primary.

■ If preonline is called due to a FAULT, the agent checks if the faulted node was SD Primary. If yes, the agent promotes one of the online SD Secondary to SD Primary before switching the instances to a new node. The switched-over node now becomes the new SD Secondary and automatically connects to SD Primary.

# Detecting HDR Failure

After a database server detects an HDR failure, it writes a message to its message log (for example, DR: receive error) and switches off the data replication. If an HDR failure occurs, the HDR connection between the two database servers is dropped and the secondary database takes appropriate action based upon the DR_AUTO configuration settings.

# Setting up IBM Informix Dynamic Server in a VCS cluster

Perform the following tasks to set up IBM Informix Dynamic Server in a cluster:

1.  Set up a VCS cluster.

2.  Install and configure IBM Informix Dynamic Server for high availability.

3.  Install the Cluster Server agent for IBM Informix Dynamic Server.

    See "Installing the agent in a VCS environment" on page 30.

4.  Configure the service groups for IBM Informix Dynamic Server.

    See "About configuring service groups for IBM Informix Dynamic Server" on page 54.

# Installing and configuring IBM Informix Dynamic Server for high availability

This chapter includes the following topics:

## About Informix Dynamic Server

Informix Dynamic Server is a multithreaded relational database server that exploits symmetric multiprocessor (SMP) and uniprocessor architectures. The Dynamic Server is a database server that processes requests for data from client applications. The client is an application program that you run to request information from a database.

The database server accesses the requested information from its databases and sends back the results to the client applications. Accessing the database includes activities such as coordinating concurrent requests from multiple clients, performing read and write operations to the databases, and enforcing physical and logical consistency on the data.

Dynamic Server provides the following features:

- Client/Server architecture

- Scalability

- High performance

- Fault tolerance and high availability

- Dynamic system administration

- Distributed data queries

- Database server security

# About Shared Disk Secondary (SDS) Servers

SDS servers access the same physical disk as the primary server. Some of the key highlights of SDS servers are:

- Provide increased availability and scalability without the need to maintain multiple copies of the database.

- Since the SDS servers use fully duplexed communications with the primary servers, having multiple SDS servers have little effect on the performance of the primary server.

- SDS servers are completely compatible with both, hardware and software disk mirroring.

Since both the SD Secondary servers read from the same disk subsystem, they can equally resume the primary server role. Thus if the primary server is unavailable, either of the secondary server can resume as the primary server.

Figure 2-1 illustrates a situation in which the primary server is offline.

**Figure 2-1**        Primary server configured with two SD secondary server
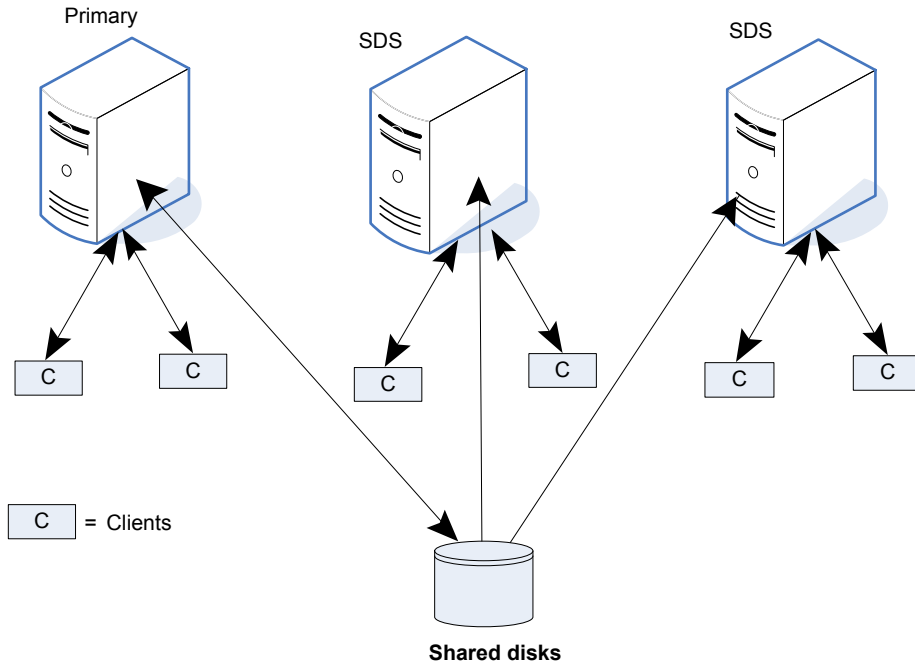


Multiple SDS servers also provide the opportunity to offload reporting and other functionality from the primary server. For example, a system with four SDS servers can have two servers allocated for analytics and two for read-only Web site data. During the holiday season, all four SDS servers could be allocated to Web site data to support the extra traffic.

Figure 2-2 illustrates a situation in which the secondary server is taking over as a primary server

**Figure 2-2**        Secondary server taking over as primary server
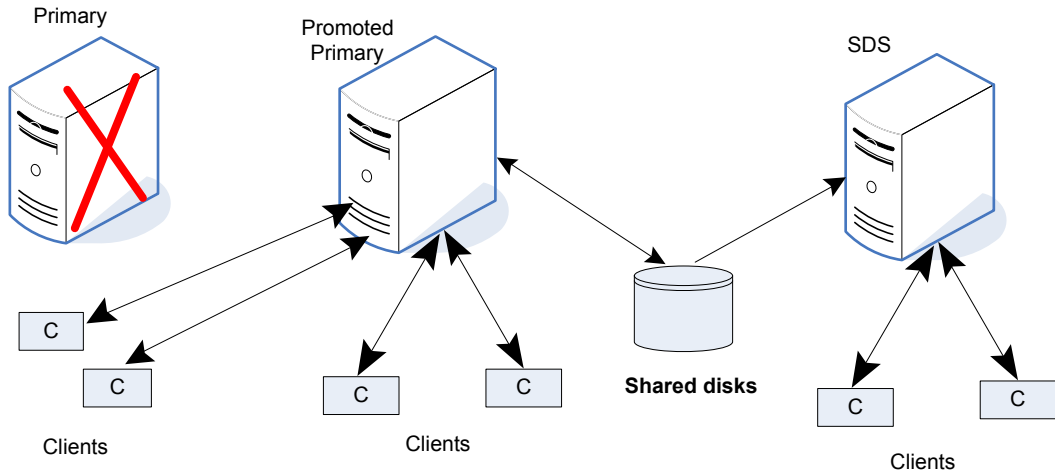


# About Informix high availability and disaster recovery

Informix HDR provides a way to maintain a backup copy of the entire database server at more than one distinct geographical location, so that the applications can access the data quickly in the event of a catastrophic failure.

In an HDR server pair, one server works as a Primary Database Server and other as a Secondary Database Server. During normal operation, clients can connect to the primary database server and use it as they would use an ordinary database server. Clients can use the secondary database server during normal operation, but only to read the data. The secondary database server does not permit updates from client applications.

Figure 2-3 represents the primary and secondary servers in a HDR pair

**Figure 2-3**            Primary and Secondary servers in a HDR
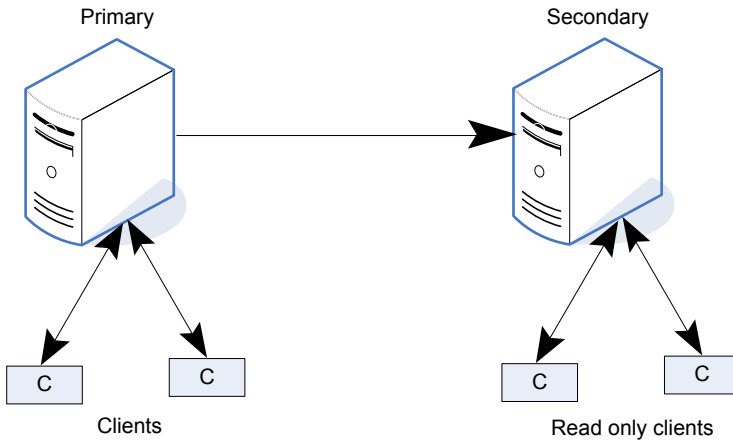


If one of the database servers in an HDR pair fails, as shown in the figure below, you can redirect the clients that use that database server to the other database server in the pair. You can also change the secondary database server to a standard database server so that it can accept updates.

Figure 2-4 represents the failover of a primary HDR server

**Figure 2-4**            Primary HDR server failover
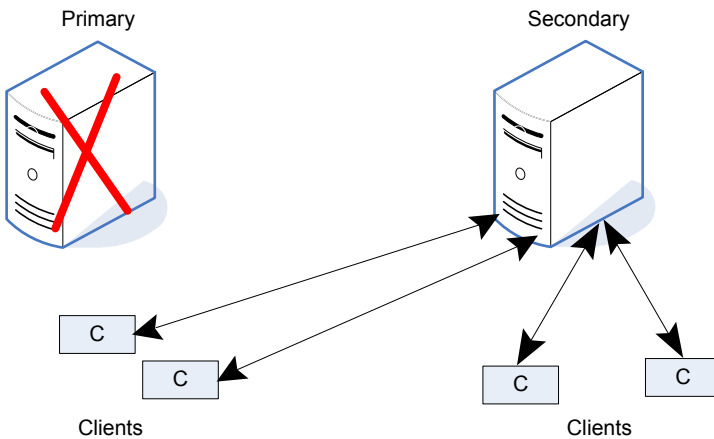
The HDR Connection depends upon DR_AUTO ONCONFIG configuration variable. If its value is 0/OFF, the secondary database server periodically attempts to re-establish communication with the primary database server.

If its value is 1 (RETAIN_TYPE), the type of that secondary database server changes automatically to standard. If its value is 2 (REVERSE_TYPE), the secondary database server becomes a primary database server as soon as the connection ends when the old primary server fails, rather than when the old primary server is restarted.

# Basic resources to cluster an IBM Informix Dynamic Server

A service group containing all resources that can support an IBM Informix Dynamic Server instance in a clustered environment forms a basic setup to cluster the server.

The required resources are as follows:

| | |
|---|---|
| Disk Group | A disk group contains a volume and a file system, which is a mount resource containing the IBM Informix Dynamic Server installation files. |
| | Use the DiskGroup resource type to create this resource. Also, create the resource on a 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 IBM Informix Dynamic Server installation files. |
| | Use the Mount resource type to create this resource. |
| Network Interface | The Network Interface resource monitors the Network Interface Card (NIC) through which the IBM Informix Dynamic Server instances communicates with the 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 IBM Informix Dynamic 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. |
| Informix | The Informix resource starts, stops, and monitors the IBM Informix Dyanmic Server instance. |
| | Use the Informix resource type to create this resource. |

# Virtualizing IBM Informix Dynamic Server

To ensure that your IBM Informix Dynamic Server can function properly on any node of the cluster, you need to virtualize all the parameters that could be dependent on a particular node.

Review the following basic notes for virtualization:

Host names

When installing and configuring the IBM Informix Dynamic Server, ensure that you enter the virtual host name associated with the IP address used to configure the IP resource. This ensures that if the application needs to be migrated, you are not tied down by the physical IP address given to the IBM Informix Dynamic Server.

The following is an excerpt from an INFORMIXSQLHOSTS file.

```
demo_on ontlitcp  infxsrvr.vxindia.veritas.com  sqlexec
```

'infxsrvr.vxindia.veritas.com' refers to the hostname associated with this instance (demo_on) of the Dynamic Server. Ensure that you use a virtual hostname for this value. The IP address associated with this hostname would then be configured within VCS using an IP resource type that can be failed over.

If the last column of this entry (sqlexec) describes a service registered via the `/etc/services` file, ensure this entry is present in the `/etc/services` file of each failover node configured for this instance. This ensures the availability of the service on the failover node.

Path names

Ensure that your application gets installed on a shared disk so that it is not constrained by anything that is local to the node. If this is not possible every time, make sure that the local data is available on each configured node.

If you are using raw devices on the shared disks for dbspaces, change the permissions (for the user "informix" to 660), or access mode, on the disk groups storing the Informix Dynamic Server data.

For example, if you are using VERITAS Volume Manager™, type:

```
# vxedit -g diskgroup_name \
set group=informix user=informix mode=660 \
volume_name
```

# Virtualizing Informix Dynamic Server libraries

For the server utilities to work, the following versions of Informix Dynamic Server (IDS) need their libraries to be linked symbolically from /usr/lib to the libraries under *InformixDir* directory.

- Informix Dynamic Server 7.31 on UNIX

- Informix Dynamic Server 9.x and later on HP-UX PA-RISC

These symbolic links are created during IDS installation, on the node on which IDS was installed. To ensure that an application successfully boots on another node after a failover, you must manually create these links on all configured failover nodes, within the Informix service group's SystemList.

## Informix Dynamic Server 7.31 on UNIX

For Informix Dynamic Server 7.31, libraries must be linked from /usr/lib to the libraries in the *InformixDir*/lib directory, on all machines within the Informix service group's SystemList. This is applicable for all Unices.

**To ensure all libraries have symbolic links, perform the following steps.**

**1**   Locate the libraries.

```
# ls -l InformixDir/lib
total 1947
-rw-r--r--  1 informix informix   36496 Oct 31 13:39 cdrapi.a
drwxr-xr-x  2 informix informix      96 Oct 31 13:39 csm
-rwxr-xr-x  1 informix informix    3126 Aug  5  2006 iosm07a.so
-rwxr-xr-x  1 informix informix   10971 Aug  5  2006 ipldd07a.so
-rwxr-xr-x  1 informix informix    1980 Aug  5  2006 ismdd07b.so
-r-xr-xr-x  1 informix informix 1198415 Aug  5  2006 libbsa.so
-rwxr-xr-x  1 informix informix  739573 Oct 31 13:39 sqlrm
```

The files with .so extension are the library files that should have symbolic links from /usr/lib.

On the node where IDS was installed, the links can be located as follows:

```
# cd /usr/lib
# ls -ltac | grep informix
lrwxrwxrwx  1 root root 29 Nov  4 18:03 ismdd07b.so.20081031
-> InformixDir/lib/ismdd07b.so
lrwxrwxrwx  1 root root 29 Nov  4 18:02 ismdd07b.so
-> InformixDir/lib/ismdd07b.so
lrwxrwxrwx  1 root root 29 Nov  4 18:02 ipldd07a.so
-> InformixDir/lib/ipldd07a.so
lrwxrwxrwx  1 root root 28 Nov  4 18:02 iosm07a.so.20081031
-> InformixDir/lib/iosm07a.so
lrwxrwxrwx  1 root root 28 Nov  4 18:01 iosm07a.so
-> InformixDir/lib/iosm07a.so
```

**2**   Create the symbolic links on each configured node, if fail to exist.

```
# cd /usr/lib
# ln -s InformixDir/lib/iosm07a.so iosm07a.so
# ln -s InformixDir/lib/ipldd07a.so ipldd07a.so
# ln -s InformixDir/lib/ismdd07b.so ismdd07b.so
# ln -s InformixDir/lib/libbsa.so libbsa.so
```

Where, *InformixDir* is INFORMIXDIR, the installation directory of IDS.

## Informix Dynamic Server 9.x and later on HP-UX PA-RISC

**To ensure all libraries have symbolic links, perform the following steps.**

**1**  Locate the JRE library.

```
/usr/lib/informix/IDS.version-JREjavarel
```

For example, for Informix Dynamic Server version 11.50.FC1 the following link exists on the installation node.

```
# ls -ltac /usr/lib/informix/IDS.11.50.FC1-JRE1.5
lrwxr-xr-x 1 root sys 33 Nov  8 18:51 /usr/lib/informix/
IDS.11.50.FC1-JRE1.5 -> InformixDir/extend/krakatoa/jre
```

**2**  Create the symbolic link, if fail to exists.

```
# mkdir /usr/lib/informix
# cd /usr/lib/informix
# ln -s InformixDir/extend/krakatoa/jre /usr/lib/informix/
IDS.version-JREjavarel
```

Where,

*InformixDir* is INFORMIXDIR, the installation directory of the IDS.

*version* is the version of the IDS.

*javarel* is the version of the associated JRE release.

# Installing, upgrading, and removing the agent for IBM Informix Dynamic Server

This chapter includes the following topics:

## Before you install the Cluster Server agent for IBM Informix Dynamic Server

You must install the Cluster Server agent for IBM Informix Dynamic Server on all the systems that will host IBM Informix Dynamic Server service groups.

Before you install the agent for IBM Informix Dynamic Server, 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.

- Remove the old Informix agent, if installed.
  To remove the agent,
  See "Uninstalling the agent in a VCS environment" on page 33.

- 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 "About the ACC library" on page 27.

- Ensure that the 'informix' login belonging to the group 'informix', with identical user and group ids exist on all systems that host the Informix Dynamic Server.

- Ensure that all the systems have sufficient shared memory to run the Informix Dynamic Server.

## Prerequisites for installing the agent to support Solaris zones

Ensure that you meet the following prerequisites to install the agent for IBM Informix Dynamic Server:

- Install Informix to support Solaris zones. For details refer to the Informix user documentation.

- Install and configure the VCS environment to support Solaris zones. Refer to the VCS user documentation for details.

- Install the required version of ACC Library.

# 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 Veritas Services and Operations Readiness Tools (SORT) site (https://sort.veritas.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 |
| 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` |
| Linux | `# rpm -i \`<br>`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

Install the ACC library IPS package on an Oracle Solaris 11 system.

**To install the ACC library IPS package on Oracle Solaris 11 systems**

**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 Veritas
```

**4** Install the package.

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

**5** Remove the publisher from the system.

```
# pkg unset-publisher Veritas
```

**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 services, `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 Veritas
```

**7** Install the package.

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

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

```
# pkg unset-publisher Veritas
```

**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 IBM Informix Dynamic Server on each node in the cluster.

**To install the agent in a VCS environment**

**1** Download the agent from the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.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/database/informix_agent/` |
| | *`vcs_version`*/*`version`*`_agent/pkgs` |
| Linux | `cd1/linux/generic/vcs/database/informix_agent/` |
| | *`vcs_version`*/*`version`*`_agent/rpms` |
| Solaris | `cd1/solaris/`*`dist_arch`*`/vcs/database/informix_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`<br>`VRTSinformix.rte.bff VRTSinformix.rte` |
| Linux | `# rpm -ihv \`<br>`VRTSinformix-`*`AgentVersion`*`-GA_GENERIC.noarch.rpm` |
| Solaris | `# pkgadd -d . VRTSinformix` |
| | **Note:** See "Installing the agent IPS package on Oracle Solaris 11 systems" on page 32. |

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

See "Importing the agent types files in a VCS environment" on page 38.

# 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 VRTSinformix.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/VRTSinformix.p5p Veritas
```

**4**   Install the package.

```
# pkg install --accept VRTSinformix
```

**5**   Remove the publisher from the system.

```
# pkg unset-publisher Veritas
```

**6**   Enable the publishers that were disabled earlier.

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

# Installing agent packages on Solaris brand non-global zones

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

**1**   Ensure that the SMF services, 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 VRTSinformix.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/VRTSinformix.p5p Veritas
```

**7** Install the package.

```
# pkg install --accept VRTSinformix
```

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

```
# pkg unset-publisher Veritas
```

**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 Solaris 10 brand zone

To install the IBM Informix Dynamic Server agent in a Solaris 10 brand zone:

- Ensure that the ACC library package, VRTSacclib, is installed in the non-global zone.
  To install VRTSacclib in the non-global zone, run the following command from the global zone:
  ```
  # pkgadd -R /zones/zone1/root -d VRTSacclib.pkg
  ```

- To install the agent package in the non-global zone, run the following command from the global zone:
  ```
  # pkgadd -R zone-root/root -d . VRTSinformix
  ```
  For example: `# pkgadd -R /zones/zone1/root -d . VRTSinformix`

# Uninstalling the agent in a VCS environment

You must uninstall the agent for IBM Informix Dynamic Server from a cluster while the cluster is active.

**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 IBM Informix Dynamic Server resources from the cluster. Run the following command to verify that all resources have been removed:

```
# hares -list Type=Informix
```

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

```
# hatype -delete Informix
```

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 IBM Informix Dynamic Server from each node in the cluster.

Run the following command to uninstall the agent:

| | |
|---|---|
| AIX | # installp -u VRTSinformix.rte |
| Linux | # rpm -e VRTSinformix |
| Solaris | # pkgrm VRTSinformix |
| | **Note:** To uninstall the agent IPS package on a Solaris 11 system, run the following command: |
| | # pkg uninstall VRTSinformix |

# 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` |
| 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
```

# Upgrading the agent in a VCS environment

Perform the following steps to upgrade the agent with minimal disruption, in a VCS environment.

**To upgrade the agent in a VCS environment**

**1**    Persistently freeze the service groups that host the application.

```
# hagrp -freeze groupName -persistent
```

**2**    Stop the Informix agent on all the nodes.

```
# haagent -stop Informix -force -sys systemName
```

**3**    Ensure that the agent is stopped on all nodes:

```
# haagent -display Informix -sys systemName
```

**4**  Uninstall the agent package from all the nodes. Use the platform's native software management program to remove the agent for IBM Informix Dynamic Server from each node in the cluster.

Run the following command to uninstall the agent:

AIX                    `# installp -u VRTSinformix.rte`

Linux                  `# rpm -e VRTSinformix`

Solaris                For Solaris 10:

                       `# pkgrm VRTSinformix`

**5**  Install the new agent on all the nodes.

The `UpdateTypes.pl` script runs automatically on successful installation of the agent.

**6**  Check for the changes in the resource values required, if any, due to the new agent types file.

---

**Note:** To note the list of changed attributes, compare the new type definition file with the old type definition file.

---

**7**  Start the agent on all nodes, if not started.

`# haagent -start Informix -sys *systemName*`

**8**  Ensure that the agent is started on all the nodes.

`# haagent -display Informix -sys *systemName*`

**9**  Unfreeze the service groups after all the resources come to an online steady state.

`# hagrp -unfreeze *groupName* -persistent`

# Configuring the agent for IBM Informix Dynamic Server

This chapter includes the following topics:

- About configuring the Cluster Server agent for IBM Informix Dynamic Server
- Importing the agent types files in a VCS environment
- Informix Dynamic Server agent attributes
- Executing a customized monitoring program

## About configuring the Cluster Server agent for IBM Informix Dynamic Server

After installing the Cluster Server agent for IBM Informix Dynamic Server, you must import the agent type configuration file. After importing this file, you can create and configure an IBM Informix Dynamic Server resource. Before you configure a resource, review the attributes table that describes the resource type and its attributes.

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

See "About sample configurations for the agents for IBM Informix Dynamic Server" on page 68.

# Importing the agent types files in a VCS environment

To use the agent for IBM Informix Dynamic Server, 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 Java GUI**

**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<br>■ Linux<br>■ Solaris | `/etc/VRTSvcs/conf/sample_Informix/`<br>`InformixTypes.cf` |
| VCS 5.x or later | ■ AIX<br>■ Linux | `/etc/VRTSagents/ha/conf/Informix/`<br>`InformixTypes.cf` |
| VCS 5.0 | Solaris SPARC | `/etc/VRTSagents/ha/conf/Informix/`<br>`InformixTypes50.cf` |
| VCS 5.1 or later | Solaris SPARC | `/etc/VRTSagents/ha/conf/Informix/`<br>`InformixTypes51.cf` |

**4** Click **Import**.

**5** Save the VCS configuration.

The IBM Informix Dynamic Server agent type is now imported to the VCS engine.

You can now create IBM Informix Dynamic Server 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 CLI**

**1** Log on to any one of the systems in the cluster as the superuser.

**2** Create a temporary directory.

```
# mkdir ./temp

# cd ./temp
```

**3** Copy the sample file `Types.cf`.

| | | |
|---|---|---|
| VCS 4.x | AIX | /etc/VRTSvcs/conf/sample_Informix/ |
| | Linux | InformixTypes.cf |
| | Solaris | |
| VCS 5.x or later | AIX | /etc/VRTSagents/ha/conf/Informix/ |
| | Linux | InformixTypes.cf |
| VCS 5.0 | Solaris SPARC | /etc/VRTSagents/ha/conf/Informix/ |
| | | InformixTypes50.cf |

The following example assumes VCS 5.0 is installed on AIX:

```
# cp /etc/VRTSagents/ha/conf/Informix/InformixTypes.cf .
```

**4** Create a dummy `main.cf` file.

```
# echo 'include "InformixTypes.cf"' > main.cf
```

**5** Create the Informix Server resource type as follows:

```
# hacf -verify .
# haconf -makerw
# sh main.cmd
# haconf -dump
```

# Importing the agent types file using Cluster Manager (Java Console)

**To import the agent types file using Cluster Manager (Java Console)**

**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 | /etc/VRTSvcs/conf/sample_Informix/ |
| | ■ Linux | InformixTypes.cf |
| | ■ Solaris | |
| VCS 5.x or later | ■ AIX | /etc/VRTSagents/ha/conf/Informix/ |
| | ■ Linux | InformixTypes.cf |

| | | |
|---|---|---|
| VCS 5.0 | Solaris SPARC | `/etc/VRTSagents/ha/conf/Informix/` `InformixTypes50.cf` |
| VCS 5.1 or later | Solaris SPARC | `/etc/VRTSagents/ha/conf/Informix/` `InformixTypes51.cf` |

**4** Click **Import**.

**5** Save the VCS configuration.

The IBM Informix Dynamic Server agent type is now imported to the VCS engine.

You can now create IBM Informix Dynamic Server resources.

# Importing the agent types file using the CLI

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

**1** If VCS is running, run the `/etc/VRTSagents/ha/conf/Informix/InformixTypes.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/<AgentTypes_file>` directory to the `/etc/VRTSvcs/conf/config` directory.

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

| | | |
|---|---|---|
| VCS 4.x | ■ AIX<br>■ Linux<br>■ Solaris | `/etc/VRTSvcs/conf/sample_Informix/` `InformixTypes.cf` |
| VCS 5.x or later | ■ AIX<br>■ Linux | `/etc/VRTSagents/ha/conf/Informix/` `InformixTypes.cf` |
| VCS 5.0 | Solaris SPARC | `/etc/VRTSagents/ha/conf/Informix/` `InformixTypes50.cf` |
| VCS 5.1 or later | Solaris SPARC | `/etc/VRTSagents/ha/conf/Informix/` `InformixTypes51.cf` |

2. Include the agent types file in the main.cf file.

3. Start HAD.

# Informix Dynamic Server agent attributes

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

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

**Table 4-1**     Required attributes

| Required attributes | Description |
|---|---|
| ResLogLevel | Specifies the logging detail performed by the agent for the resource.<br><br>The valid values are as follows:<br><br>■ ERROR: Only logs error messages.<br>■ WARN: Logs above plus warning messages.<br>■ INFO: Logs above plus informational messages.<br>■ TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations.<br><br>Type and dimension: string-scalar<br><br>Default: INFO<br><br>Example: INFO<br><br>**Note:** You must use the LogDbg attribute instead of the ResLogLevel attribute to enable 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. The agent captures the first failure data of the unexpected events and automatically logs debug messages in their respective agent log files. |
| DBServerName | Specifies the unique name of this instance of the Database Server. This is identical to the environment variable INFORMIXSERVER needed to start the instance. It is also known by the DBSERVERNAME variable specified in the ONCONFIG configuration file.<br><br>Type and dimension: string-scalar<br><br>Default: ""<br><br>Example: oasis_mumbai |

**Table 4-1**        Required attributes *(continued)*

| Required attributes | Description |
|---|---|
| EnvFile | Full path for the file to be sourced before executing the Database Server management programs. Veritas recommends to save the file on a shared disk with InformixDir.<br><br>The shell environments supported are: ksh, sh, and csh.<br><br>Type and dimension: string-scalar<br><br>Default: ""<br><br>Example: /space/infxll.50/envfile_demo_on |
| InformixDir | Absolute path name of the directory in which the Database Server is installed.<br><br>Type and dimension: string-scalar<br><br>Default: ""<br><br>Example 1: /space/infxll.50<br><br>Example 2: /usr/infx9.40 |
| ServerType | Specifies the Informix IDS server configuration type.<br><br>The valid values are:<br><br>■ Standard<br>　If the Informix instance has standard configuration<br>■ HDR (High-availability and Disaster Recovery)<br>　If the instance is configured as HD Primary or HD Secondary<br>■ SharedDisk<br>　If the instance is configured as SD Primary or SD Secondary<br><br>**Note:** If the Instance is configured as both, HD Primary and SD Primary then ServerType should be SharedDisk.<br><br>Type and dimension: string-scalar<br><br>Default: Standard<br><br>Example: HDR |

**Table 4-1**        Required attributes *(continued)*

| Required attributes | Description |
|---|---|
| UseSystemD | SystemD is a system and a service manager for Linux operating systems. It helps manage applications across Linux distributions that support the SystemD feature. The UseSystemD attribute is applicable only on those systems that use the supported distributions of RHEL and SLES. |
| | When this attribute is set to 1, the Informix agent starts the Informix resource in system.slice. When this attribute is set to 0, a typical online function starts the resource in user.slice. |
| | Type and dimension: boolean-scalar |
| | Example: 1 |
| | Default: 0 |
| SDSInstances | Specifies the list of all the connected SDSInstances. |
| | Type and dimension: string-vector |
| | Default: "" |
| | Example: infx_sds1, infx_sds2 |

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

**Table 4-2**        Optional attributes

| Optional attribute | Description |
|---|---|
| LogDbg | LogDbg For ACCLib-based agents, you must use the LogDbg resource type attribute to enable the debug logs when the ACCLib version is 6.2.0.0 or later and the VCS version is 6.2 or later. |
| | Set the LogDbg attribute to DBG_5 to enable debug logs for ACCLIB based agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all Informix resources in the cluster. If debug logs must be enabled for a specific Informix resource, override the LogDbg attribute. |
| | Type and dimension: string-keylist |
| | Default: {} (none) |
| | For more information on how to use the LogDbg attribute, refer to the `Cluster Server Administrator's Guide`. |

**Table 4-2**        Optional attributes *(continued)*

| Optional attribute | Description |
|---|---|
| MonitorProgram | Absolute path name of an external, user-supplied monitor executable. If specified, the monitor function executes this file to perform an additional state check of the server. There are no restrictions for what actions the external monitor program performs to determine the state of the server.<br><br>For information about setting this attribute:<br><br>See "Executing a customized monitoring program" on page 45.<br><br>Type and dimension: string-scalar<br><br>Default: ""<br><br>Example 1: /space/infx11.50/bin/myMonitor.pl<br><br>Example 2: /space/infx11.50/bin/monitor_demo.pl *arg1 arg2* |
| SecondLevelMonitor | Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the Informix Database 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. This interpretation may be extended to other values.<br><br>**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.<br><br>Type and dimension: integer-scalar<br><br>Default: 0<br><br>Example: 5 |

**Table 4-2**         Optional attributes *(continued)*

| Optional attribute | Description |
|---|---|
| DBServerAliases | Used to specify an alias name, or a list of unique alias names for the database server. The list of alias names should be identical to the list of values specified in the INFORMIXSERVERALIASES environment variable and the DBSERVERALIASES variable specified in the onconfig configuration file. |
| | Type and dimension: string-vector |
| | Default: "" |
| | Example: oasis_mumbai_alias, sample_alias |
| | This example would be specified in the `$INFORMIXDIR/etc/onconfig.<DBServerName>` file. |
| | While specifying multiple aliases in the environment file, separate aliases with colon, ":". |
| | Example: INFORMIXSERVERALIAS=ol_informix1210_2_alias:sample_alias:sample_alias1 |
| | This example would be specified in the environmental file `$INFORMIXDIR/<DBServerName>.<ksh/csh>`. |

# Executing a customized monitoring program

You can configure the monitor function to execute a custom monitor utility to perform a user-defined Informix Server state check. The utility is executed in the context of the user 'informix'. The environment is set by sourcing the file specified in the EnvFile attribute.

The monitor function executes the utility specified in the MonitorProgram attribute if the following conditions are satisfied:

- The MonitorProgram attribute value is set to a valid executable utility.

- The first-level process check indicates that the IBM Informix Dynamic Server instance is online.

- The LevelTwoMonitorFreq attribute is set to 1 and the second-level check returns the server state as "online" or the LevelTwoMonitorFreq attribute is set to a value greater than 1, but the second-level check is deferred for this monitoring cycle.

The monitor function interprets the program exit code as follows:

| 110 or 0 | IBM Informix Dynamic Server instance is online |
| 100 or 1 | IBM Informix Dynamic Server instance is offline |
| Any other value | IBM Informix Dynamic Server instance is unknown |

To ensure that the custom monitor utility is always available to the agent, Veritas recommends storing the file in the directory where the IBM Informix Dynamic Server is installed.

# Enabling the agent for IBM Informix Dynamic Server to support IMF

This chapter includes the following topics:

## About Intelligent Monitoring Framework

With the IMF feature, VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring. You can enable or disable the intelligent resource monitoring functionality of the IBM Informix Dynamic Server agent.

VCS process and mount-based agents use the AMF kernel driver that provides asynchronous event notifications to the agents that are enabled for IMF.

You can enable the IBM Informix Dynamic Server agent for IMF, provided the following software versions are installed:

- Cluster Server (VCS) 5.1 SP1 or later

- Cluster Server agent for IBM Informix Dynamic Server version 5.1.0.0 or later

Refer to the *Cluster Server Administrator's Guide* for more information about IMF notification module functions and administering the AMF kernel driver.

## Benefits of IMF

IMF offers the following benefits:

- Performance
  Enhances performance by reducing the monitoring of each resource at a default of 60 seconds for online resources, and 300 seconds for offline resources. IMF enables the agent to monitor a large number of resources with a minimal effect on performance.

- Faster detection
  Asynchronous notifications would detect a change in the resource state as soon as it happens. Immediate notification enables the agent to take action at the time of the event.

# Agent functions for the IMF functionality

If the IBM Informix Dynamic Server agent is enabled for IMF support, the agent supports the following functions, in addition to the functions mentioned in the IBM Informix Dynamic Server agent functions topic.

## imf_init

This function initializes the IBM Informix Dynamic Server agent to interface with the AMF kernel driver, which is the IMF notification module for the agent for IBM Informix Dynamic Server. This function runs when the agent starts up.

## imf_getnotification

This function gets notifications about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

## imf_register

This function registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into a steady state—online or offline.

# Attributes that enable IMF

If the agent for IBM Informix Dynamic Server is enabled for IMF support, the agent uses type-level attributes in addition to the agent-specific attributes.

## IMF

This resource type-level attribute determines whether the IBM Informix Dynamic Server agent must perform intelligent resource monitoring. You can also override the value of this attribute at the resource level.

This attribute includes the following keys:

### Mode

Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows:

- 0—Does not perform intelligent resource monitoring

- 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources

- 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources

- 3—Performs intelligent resource monitoring for both online and for offline resources.

**Note:** The agent for IBM Informix Dynamic Server supports intelligent resource monitoring for online resources only. Hence, Mode should be set to either 0 or 2.

Type and dimension: integer-association

Default: 0 for VCS 5.1 SP1, 3 for VCS 6.0 and later.

### MonitorFreq

This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer.

Default: 1

You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring.

If the value is 0, the agent does not perform poll-based process check monitoring.

After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows:

- After every (MonitorFreq x MonitorInterval) number of seconds for online resources
- After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources

### RegisterRetryLimit

If you enable intelligent resource monitoring, the agent invokes the imf_register agent function to register the resource with the AMF kernel driver.

The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes.

Default: 3.

## IMFRegList

An ordered list of attributes whose values are registered with the IMF notification module.

Type and dimension: string-vector

Default: No default value

**Note:** The attribute values can be overriden at the resource level.

# Before you enable the agent to support IMF

Before you enable the IBM Informix Dynamic Server agent to support IMF, ensure that the AMF kernel module is loaded and AMF is configured. For details, refer to the 'Administering the AMF kernel driver' section of the *Cluster Server Administrator's Guide*. For details about the commands you can configure AMF using the `amfconfig -h` command.

# Enabling the agent to support IMF

In order to enable the IBM Informix Dynamic Server agent to support IMF, you must make the following configuration changes to the attributes of the agent:

- AgentFile: Set the AgentFile attribute to **Script51Agent**

- IMF Mode: Set the IMF Mode attribute to **2**

- IMFRegList: Update the IMFRegList attribute

The following sections provide more information about the commands you can use to make these configuration changes, depending on whether VCS is in a running state or not.

---

**Note:** If you have upgraded VCS from an earlier version to version 5.1 SP1 or later, and you already have IBM Informix Dynamic Server agent 5.1.00 or later installed, ensure that you run the following commands to create appropriate symbolic links:

```
# cd /opt/VRTSagents/ha/bin/Informix

# ln -s /opt/VRTSamf/imf/imf_getnotification imf_getnotification

# ln -s /opt/VRTSagents/ha/bin/Informix/monitor imf_register
```

---

## If VCS is in a running state

**To enable the IBM Informix Dynamic Server resource for IMF when VCS is in a running state:**

**1** Make the VCS configuration writable.

```
# haconf -makerw
```

**2** Run the following command to update the AgentFile attribute.

```
# hatype -modify Informix AgentFile\
/opt/VRTSvcs/bin/Script51Agent
```

**3** For VCS version 6.0 or later, run the following commands to add the IMF attributes:

```
# haattr -add -static  Informix IMF -integer -assoc Mode 0 \
MonitorFreq 1 RegisterRetryLimit 3

# haattr -add -static Informix IMFRegList -string -vector
```

---

**Note:** Run these commands only once after you first enable IMF support for the agent.

---

**4** Run the following command to update the IMF attribute.

```
# hatype -modify Informix IMF Mode num MonitorFreq num
RegisterRetryLimit num
```

For example, to enable intelligent monitoring of online resources, with the MonitorFreq key set to 5, and the RegisterRetryLimit key is set to 3, run the following command:

```
# hatype -modify Informix IMF Mode 2 MonitorFreq 5 \
RegisterRetryLimit 3
```

---

**Note:** The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

---

**5** Run the following command to update the IMFRegList attribute:

```
# hatype -modify Informix IMFRegList DBServerName DBServerAliases
EnvFile InformixDir
```

**6** Save the VCS configuration.

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

**7** If the IBM Informix Dynamic Server agent is running, restart the agent.

For information on the commands you can use to restart the agent, see Restarting the agent.

## Restarting the agent

**To restart the agent:**

**1** Run the following command to stop the agent forcefully:

```
# haagent -stop Informix -force -sys <system>
```

---

**Note:** Stopping the agent forcefully eliminates the need to take the resource offline.

---

**2** Run the following command to start the agent:

```
# haagent -start Informix -sys <system>.
```

## If VCS is not in a running state

**To change the Informix type definition file when VCS is not in a running state:**

**1** Update the AgentFile attribute.

```
static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
```

**2** Update the IMF attribute.

The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

```
static int IMF{} = { Mode=num, MonitorFreq=num,
RegisterRetryLimit=num }
```

For example, to update the IMF attribute such that the Mode key is set to 2, the MonitorFreq key is set to 5, and the RegisterRetryLimit key is set to 3:

```
static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3
}
```

**3** Update the IMFRegList attribute.

```
static str IMFRegList[] = { DBServerName DBServerAliases EnvFile
InformixDir }
```

# Disabling intelligent resource monitoring

**To disable intelligent resource monitoring**

**1** Make the VCS configuration writable.

```
# haconf -makerw
```

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

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

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

```
# hares -override resource_name IMF
```

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

**4** Save the VCS configuration.

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

# Configuring the service groups for IBM Informix Dynamic Server using the CLI

This chapter includes the following topics:

## About configuring service groups for IBM Informix Dynamic Server

Configuring the IBM Informix Dynamic Server service group involves creating the IBM Informix Dynamic Server 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

# Before configuring the service groups for IBM Informix Dynamic Server

Before you configure the IBM Informix Dynamic Server service group, you must:

- Verify that the Cluster Server components are installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing the components, refer to the *InfoScale Availability Installation Guide*.
- Verify that the Cluster Server agent for IBM Informix Dynamic Server is installed on all nodes in the cluster.
  See "Installing the agent in a VCS environment" on page 30.

# Configuring IBM Informix Dynamic Server resources for Solaris zones support

To enable the agent for IBM Informix Dynamic Server to support Solaris zones, ensure that you perform the following configuration steps:

- Install IBM Informix Dynamic Server on dedicated Solaris zones.
- Import the `InformixTypes50.cf` (for VCS 5.0) or `InformixTypes51.cf` (for VCS 5.1 or later) file for Solaris zone support.
- Preferably, follow the Veritas recommendation of installing zones on a shared disk for convenient configuration, failover, and maintenance.
- Make sure that the name of the Solaris zone is the same as the virtual host name that you use to install and configure the IBM Informix Dynamic Server.
- In a VCS environment, ensure that you have set the value of ContainerName attribute to the name of the Solaris zone.
  By default, the agent function executes in the Global zone.

# Configuring Action script

The Action script comprises the following two scripts:

- switchprim: This script facilitates the promotion of SD Secondary Server to SD Primary Server, in case SD Primary server fails.

- isprimary: This script retrieves the Shared Disk (SD) configuration of current Informix instance. It is thus configured as SD Primary or SD Secondary Server.

**To configure the VCS action scripts in the VCS 4.x and 5.0 environments**

1   Verify if the symlinks are created between the action scripts and the monitor script. If the links do not exists, create them using the following commands:

    For VCS 4.x

    ```
    cd /opt/VRTSvcs/bin/Informix
    mkdir actions
    cd actions
    ln -s /opt/VRTSvcs/bin/Informix/monitor switchprim
    ln -s /opt/VRTSvcs/bin/Informix/monitor isprimary
    ```

    For VCS 5.x or later

    ```
    cd /opt/VRTSagents/ha/bin/Informix
    mkdir actions
    cd actions
    ln -s /opt/VRTSagents/ha/bin/Informix/monitor switchprim
    ln -s /opt/VRTSagents/ha/bin/Informix/monitor isprimary
    ```

2   Modify the informix agent type attribute "SupportedActions" and set it to "switchprim, isprimary".

    Refer to *Cluster Server Agent Developer's Guide* for more information.

# Configuring preonline script

In a clustered environment, if the Informix SDS Primary instance fails, it is required to promote one of the connected SD Secondary to SD primary and then failover the faulted SD Primary to other node as SD Secondary.

The existing VCS preonline script calls the Informix preonline script to facilitate smooth SDS server failover behavior.

The Informix preonline script performs the following tasks:

- Executes the "switchprim" action script which promotes one of the SD Secondary to SD Primary.

- Fails over the faulted SD Primary to next node in cluster and brings the SD Secondary online.

- Ensures that the online operation does not execute the VCS preonline script again.

**To configure the VCS preonline script in the VCS 4.x and 5.0 environments**

1   Verify if the symlink has been created between the preonline script and the monitor script. If not, create them using the following commands:

For VCS 4.x

```
cd /opt/VRTSvcs/bin/Informix
ln -s /opt/VRTSvcs/bin/Informix/monitor preonline
```

For VCS 5.x or later

```
cd /opt/VRTSagents/ha/bin/Informix
ln -s /opt/VRTSagents/ha/bin/Informix/monitor preonline
```

2   Navigate to $VCS_Home/bin/triggers directory.

**3** In the preonline file, add the following lines to integrate the call to the Informix preonline trigger, in the main trigger script.

---

**Note:** If the preonline file does not exists, proceed to step 4.

---

```
eval 'exec /opt/VRTSperl/bin/perl -Sw $0 ${1+"$@"}'
   if 0;

use strict;
use vars;

my $vcs_home = $ENV{"VCS_HOME"};
if (!defined ($vcs_home)) {
    $vcs_home="/opt/VRTSvcs";
}

use ag_i18n_inc;
VCSAG_SET_ENVS();
if (!defined $ARGV[0]) {
   VCSAG_LOG_MSG ("W", "Failed to continue;
undefined system name", 15028);
   exit;
} elsif (!defined $ARGV[1]) {
   VCSAG_LOG_MSG ("W", "Failed to continue;
undefined group name", 15031);
   exit;
}

#Add the Informix Trigger Call here..
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/Informix/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs);
my $iExitCode = undef;
#----------------------------------------
# Pass control to preonline, if it exists..
#----------------------------------------
if ( -x $sCmd ) {
   VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd] Args [$sArgs]", 15031);
```

```
   system ( $sCmd, @lsCmdArgs );
   #-------------------
   # Exit if successful..
#----------------------------
   exit $iExitCode unless ( $iExitCode = $?>> 8 );
}
# Give control back to HAD.
if (defined $ARGV[3]) {
   system("$vcs_home/bin/hagrp -online -nopre $ARGV[1]
-sys $ARGV[0] -checkpartial $ARGV[3]");
   exit;
}
system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] -sys $ARGV[0]");
exit;
```

4    If the preonline script does not exists, perform the following:

   ■   Navigate to the following directory.

   | VCS 4.x | /etc/VRTSvcs/conf/sample_Informix |
   | VCS 5.x or later | /etc/VRTSagents/ha/conf/Informix |

   ■   Copy the sample preonline script to the $VCS_HOME/bin/triggers directory.

   ■   Ensure that the file is executable and accessible to the "root" user.

5    For the service group, set the preonline flag to True.

The preonline script is now configured to facilitate Informix SDS behavior. To configure the logging level used in the preonline script, you can set the ResLogLevel attribute in the preonline wrapper. You can then view the logs in the VCS engine log, /var/VRTSvcs/log/engine_A.log

---

**Note:** Once the preonline trigger is configured, you may see unexpected behavior while manually switching or performing online operations on the Informix service group. This behavior is a result of the control logic within the preonline trigger that protects the Informix SDS Primary Secondary relationship. For system maintenance, if you prefer to perform manual operations on the service groups, you can do so by disabling the preonline trigger as follows:

```
# hagrp -modify service_group PreOnline 0
```

---

# Configuring the agent for Informix Shared Disk Configuration

**To configure the agent for Shared Disk configuration:**

1   Change the ServerType attribute and set this to "Shared Disk".

2   Set the SDSInstances attribute to the list of all connected Shared Disk instances.

3   Configure the Action Scripts.

   See "Configuring Action script" on page 56.

4   Configure the preonline script.

   See "Configuring preonline script" on page 56.

# Creating service groups for IBM Informix Dynamic Server under Solaris non-global zones

**To configure zones on each cluster node**

1   Set up the non-global zone configuration.

   ```
   hazonesetup servicegroup_name zoneres_name zone_name password
   systems
   ```

   For example:

   ```
   hazonesetup -g servicegroup_name -r zoneres_name -z zone_name
    -p password -s systems
   ```

2   Verify the non-global zone configuration.

   ```
   hazoneverify servicegroup_name
   ```

3   Whenever you make a change that affects the zone configuration, run the hazonesetup command to reconfigure the zones in VCS.

4   Make sure that the zone configuration files are consistent on all nodes at all times. The file is located at /etc/zones/zone_name.xml.

5   Make sure that the application is identical on all nodes. If you update the application configuration on one node, apply the same updates to all nodes.

6   Configure the service groups for IBM Informix Dynamic Server.

# Troubleshooting the agent for IBM Informix Dynamic Server

This chapter includes the following topics:

- Using the correct software and operating system versions
- Meeting prerequisites
- Configuring IBM Informix Dynamic Server resources
- Verifying virtualization
- Starting the IBM Informix Dynamic Server instance outside a cluster
- Reviewing error log files
- Handling the agent failure in SDS configuration

## 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 IBM Informix Dynamic Server supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

# Meeting prerequisites

Before installing the agent for IBM Informix Dynamic Server, ensure that the following prerequisites are met.

For example, you must install the ACC library on VCS before installing the agent for IBM Informix Dynamic Server.

See "Before you install the Cluster Server agent for IBM Informix Dynamic Server" on page 26.

# Configuring IBM Informix Dynamic Server resources

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

# Verifying virtualization

Verify that your application does not use anything that ties it down to a particular node of the cluster.

See "Virtualizing IBM Informix Dynamic Server " on page 22.

# Starting the IBM Informix Dynamic Server 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 IBM Informix Dynamic Server instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the IBM Informix Dynamic Server 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.

---

Execute the following commands to start, stop or monitor the IBM Informix Dynamic Server outside the cluster framework.

You must execute these commands as user 'informix', after sourcing the environment file specified by the EnvFile agent attribute.

To start      `$ InformixDir/bin/oninit -y`

To stop      `$ InformixDir/bin/onmode -ky`

To monitor      First verify whether the Informix 'oninit' parent process is running as the user 'informix'. This process forks off other helper 'child' processes.

For example,

```
# ps -ef | grep oninit | grep  -v grep
informix 3657     1  1 16:33:36 ?   0:00 oninit -y
    root 3750  3660  0 16:33:39 ?   0:00 oninit -y
    root 3756  3660  0 16:33:42 ?   0:00 oninit -y
    root 3710  3660  0 16:33:38 ?   0:00 oninit -y
    root 3753  3660  0 16:33:41 ?   0:00 oninit -y
    root 3752  3660  0 16:33:41 ?   0:00 oninit -y
    root 3679  3660  0 16:33:37 ?   0:00 oninit -y
    root 3751  3660  0 16:33:40 ?   0:00 oninit -y
    root 3660  3657  0 16:33:36 ?   0:00 oninit -y
    root 3661  3660  0 16:33:36 ?   0:00 oninit -y
    root 3755  3660  0 16:33:42 ?   0:00 oninit -y
    root 3754  3660  0 16:33:41 ?   0:00 oninit -y
```

In the above example the process with PID 3657 is the parent 'oninit' process. The agent essentially monitors this process.

Now, execute the following command and verify the exit code.

```
$ InformixDir/bin/onstat -
$ echo $?
5
```

An exit code of 5 implies that the Database Server is online and healthy.

An exit code of 255 implies that the Database Server is offline.

For all other exit codes the agent reports the application status as unknown.

# Reviewing error log files

If you face problems while using IBM Informix Dynamic Server or the agent for IBM Informix Dynamic Server, use the log files described in this section to investigate the problems.

The common reasons for issues are as follows:

| Insufficient privileges | Files that need to be created, written to, would be created as user 'informix'. Check if necessary privileges have been set. |
|---|---|
| Incorrect port, environment or parameter settings | Verify that ports have been properly configured and declared. Typically, ports from 1 through 1024 are reserved for the superuser. |
| | Ensure that parameters to the agent are correctly defined. |
| Expired licenses | Check the application log files for any error messages related to expired licenses. |
| | Ensure that the license keys/files have been placed at the appropriate location, as needed by the application. |
| Broken symlinks, missing files, and libraries | Verify your installation. |
| | Make sure that nothing is broken, and all dependencies for the executables are met. |
| Insufficient disk space or system parameters | Ensure that the file-system has sufficient space for creation of temporary files that the application might need. |
| | Verify that the kernel has been tuned for sufficient IPC resources, file descriptors and meets the hardware requirement. Consult your product documentation for these details. |

# Using IBM Informix Dynamic Server log files

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

The MSGPATH variable in the ONCONFIG file describes the location where the log file for an instance of the Database Server gets created.

For example, following is an excerpt from an ONCONFIG file

```
MSGPATH /space/infx11.50/demo2/server/online.log
```

Consult your application expert if needed.

# Reviewing cluster log files

In case of problems while using the agent for IBM Informix Dynamic Server, 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`.

# Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each IBM Informix Dynamic 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 are produced. Veritas recommends that you localize the ResLogLevel attribute for a particular resource.

---

**Note:** Starting with version 5.1.1.0 of the ACC library, the TRACE level logs for any ACCLib based agent are generated locally at the location `/var/VRTSvcs/log/`*Agent*`_A.log`.

---

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   Test the identified resource. The function 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   Save the configuration changes.

```
# haconf -dump
```

7   Review the contents of the log file.

You can also contact Veritas support for more help.

**To enable debug logs for all resources of type Informix**

◆ Enable the debug log.

```
# hatype -modify Informix 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 Informix LogDbg
# hares -modify Informix LogDbg DBG_5
```

# Using trace level logging for preonline trigger

While executing the preonline trigger, you can set the ResLogLevel attribute to
TRACE, to enable detailed logging.

**To set the ResLogLevel attribute for preonline trigger**

**1** Navigate to the `$VCS_HOME/bin/trigger`s directory.

**2** Open the preonline file, and go to this section:

```
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/Informix/preonline';
my $sResLogLevel = 'INFO'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

**3** Edit the value of the ResLogLevel attribute:

```
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/Informix/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

**4** Save and close the preonline file.

You can view the logs in the VCS engine log at
`/var/VRTSvcs/log/engine_A.log` and the agent log at
`/var/VRTSvcs/log/Informix_A.log`.

# Handling the agent failure in SDS configuration

If the agent is configured for Informix Shared Disk Configuration you may experience agent failure during the online function in the following cases:

- The SD Secondary is brought online before the SD Primary.
  In this case the SD Secondary is unable to initialize the SDS Network with the SD Primary and thus fails to come online.
  **Workaround**:

  - Login as a superuser

  - Navigate to $INFORMIXHOME/.VRTSInfxSDPrimDB

  - Find the name of the existing SD Primary instance

  - Bring the SD Primary instance online

  - Bring the SD Secondary instances online

- The $INFORMIXHOME/.VRTSInfxSDPrimDB file is corrupt.
  Login as a superuser and verify if the file is corrupt. If yes, delete the file and then bring the instance online.

# Sample Configurations

This appendix includes the following topics:

## About sample configurations for the agents for IBM Informix Dynamic Server

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 IBM Informix Dynamic Server. 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 IBM Informix Dynamic Server agent on different versions of VCS.

For VCS 4.x

```
type Informix (
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        InformixDir, DBServerName, ServerType, SDSInstances,
        SecondLevelMonitor, MonitorProgram }
        static keylist SupportedActions = { switchprim, isprimary }
        str ResLogLevel = INFO
        str EnvFile
        str InformixDir
        str DBServerName
        str ServerType = "Standard"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

For VCS 5.0

```
type Informix (
        static boolean AEPTimeout = 1
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/Informix"
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        InformixDir, DBServerName, ServerType, SDSInstances,
        SecondLevelMonitor, MonitorProgram }
        static keylist SupportedActions = { switchprim, isprimary }
        str ResLogLevel = INFO
        str EnvFile
        str InformixDir
        str DBServerName
        str DBServerAliases[]
        str ServerType = "Standard"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

# Sample agent type definition with Solaris zone support

This section lists the sample agent type definition files with Solaris zone support, for IBM Informix Dynamic Server agent.

For VCS 5.0

```
type Informix (
        static boolean AEPTimeout = 1
        static str ContainerType = Zone
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/Informix"
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        InformixDir, DBServerName, ServerType, SDSInstances,
        SecondLevelMonitor, MonitorProgram }
        static keylist SupportedActions = { switchprim, isprimary }
        str ResLogLevel = INFO
        str EnvFile
        str InformixDir
        str DBServerName
        str ServerType = "Standard"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
        str ContainerName
)
```

### For VCS 5.1 or later

```
type Informix (
        static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
        static boolean AEPTimeout = 1
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/Informix"
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        InformixDir, DBServerName, ServerType, SDSInstances,
        SecondLevelMonitor, MonitorProgram }
        static keylist SupportedActions = { switchprim, isprimary }
        str ResLogLevel = INFO
        str EnvFile
        str InformixDir
        str DBServerName
        str ServerType = "Standard"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

# Sample agent type definition with Shared Disk (SD) Configuration

This section lists the sample agent type definition with SD configuration for IBM Informix Server agent.

For VCS 4.x

```
type Informix (
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        InformixDir, DBServerName, ServerType, SDSInstances,
        SecondLevelMonitor, MonitorProgram }
        static keylist SupportedActions = { switchprim, isprimary }
        str ResLogLevel = INFO
        str EnvFile
        str InformixDir
        str DBServerName
        str ServerType = "SharedDisk"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

For VCS 5.0

```
type Informix (
        static boolean AEPTimeout = 1
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/Informix"
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        InformixDir, DBServerName, ServerType, SDSInstances,
        SecondLevelMonitor, MonitorProgram }
        static keylist SupportedActions = { switchprim, isprimary }
        str ResLogLevel = INFO
        str EnvFile
        str InformixDir
        str DBServerName
        str ServerType = "SharedDisk"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

# Sample configuration

This section contains sample configurations for IBM Informix Dynamic Server and for Informix HDR.

## Sample configuration for IBM Informix Dynamic Server

This section provides sample configuration for IBM Informix Dynamic Server.

```
include "types.cf"
include "InformixTypes.cf"

cluster infx11sol (
 UserNames = { admin = bIJbIDiFJeJJhRJdIG }
 Administrators = { admin }
        CredRenewFrequency = 0
        CounterInterval = 5
 )

system nodeA (
 )

system nodeB (
 )

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

 DiskGroup infx11_dg (
  DiskGroup = infx11dg
  )

 IP infx11_ip (
  Device = bge0
  Address = "110.120.62.18"
  NetMask = "255.255.255.0"
  )

 Informix infx11_srvr (
  EnvFile = "/space/infx11.50/envfile_demo_on2"
  InformixDir = "/space/infx11.50"
  DBServerName = demo_on2
```

```
   DBServerAliases = { sample_alias, ol_informix1210_2_alias,
                       sample_alias1 }
   SecondLevelMonitor = 1
   )

Mount infx11_mnt (
 MountPoint = "/space/infx11.50"
 BlockDevice = "/dev/vx/dsk/infx11dg/infx11vol"
 FSType = vxfs
 MountOpt = largefiles
 FsckOpt = "-y"
 SecondLevelMonitor = 1
 )

NIC infx11_nic (
 Device = bge0
 )

infx11_ip requires infx11_nic
infx11_mnt requires infx11_dg
infx11_srvr requires infx11_ip
infx11_srvr requires infx11_mnt


// resource dependency tree
//
// group InformixDemo
// {
// Informix infx11_srvr
//      {
//      Mount infx11_mnt
//          {
//          DiskGroup infx11_dg
//          }
//      IP infx11_ip
//          {
//          NIC infx11_nic
//          }
//      }
// }
```

## Sample configuration for Informix HDR

This section provides sample configuration for Informix HDR.

```
group infx (
        SystemList = { nodeA = 0, nodeB = 1 }
        Parallel = 1
        )

Informix demo_on (
        Critical = 0
        EnvFile @nodeA = "/ibm/informix/envfile1"
        EnvFile @nodeA = "/ibm/informix/envfile2"
        InformixDir = "/ibm/informix"
        DBServerName @nodeA = demo_on
        DBServerName @nodeB = demo_on2
        ServerType = HDR
        )

// resource dependency tree
//
// group infx
// {
// Informix demo_on
// }
```

# Sample configuration with Solaris zone support

This section provides a sample configuration with Solaris zone support for IBM Informix Dynamic Server agent.

```
include "types.cf"
include "InformixTypes50.cf"

cluster infx115 (
 UserNames = { admin = bIJbIDiFJeJJhRJdIG }
 Administrators = { admin }
 HacliUserLevel = COMMANDROOT
 )

system nodeA (
 )
```

```
system nodeB (
 )

group InformixZone (
 SystemList = { nodeB = 0, nodeA = 1 }
 Administrators = { z_infx115_zone_nodeB, z_infx115_zone_nodeA }
 )

 DiskGroup infx115_dg (
  DiskGroup = infx115dg
  )

 Informix infx115_srvr (
  EnvFile = "/space/infx11.50/envfile_demo_on2"
  InformixDir = "/space/infx11.50"
  DBServerName = demo_on2
  SecondLevelMonitor = 1
  ContainerName = infx115
  )

 Mount infx115_mnt (
  MountPoint = "/space/zones/infx115"
  BlockDevice = "/dev/vx/dsk/infx115dg/infx115vol"
  FSType = vxfs
  MountOpt = largefiles
  FsckOpt = "-y"
  SecondLevelMonitor = 1
  )

 NIC infx115_nic (
  Device = bge0
  )

 Zone infx115_zone (
  ZoneName = infx115
  )

 infx115_mnt requires infx115_dg
 infx115_srvr requires infx115_zone
 infx115_zone requires infx115_mnt
 infx115_zone requires infx115_nic
```

```
// resource dependency tree
//
// group InformixZone
// {
// Informix infx115_srvr
//     {
//     Zone infx115_zone
//         {
//         Mount infx115_mnt
//             {
//             DiskGroup infx115_dg
//             }
//         NIC infx115_nic
//         }
//     }
// }
```
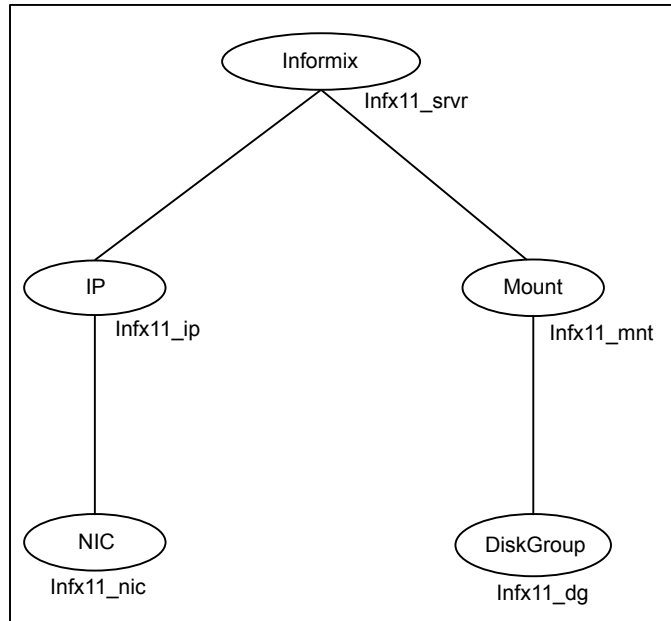
# Sample service group configuration

This section includes sample service groups configurations in a VCS environment.

Figure A-1 shows a service group with an Informix Dynamic Server instance running in a VCS environment.

The service group includes a DiskGroup resource, a NIC resource, an IP resource and Mount resource, along with an Informix Dynamic Server resource.

**Figure A-1**     Sample service group for an Informix Dynamic Server instance



# Sample service group configuration with Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure A-2 shows a service group with an Informix Dynamic Server instance running in a local zone, if the zone binaries are present on a shared disk.

The Dynamic Server itself is installed locally within the Solaris zone.

**Figure A-2**     Sample service group with an Informix Dynamic Server instance