WHITE PAPER: TECHNICAL

symantec.

Confidence in a connected world.

Symantec High Availability Solution for Informix Shared Disk Secondary

Mangilal Daravath
Sr. SQA Engineer

Sharad Srivastava
Software Engineer

Storage and Availability Management Group

Mangilal_D@Symantec.com
Sharad_Srivastava1@Symantec.com

Table of Contents

Table of Figures

Introduction

IBM Informix Dynamic Server (IDS) is the Informix flagship database server. IDS is the best-of-breed, general purpose, but mission critical, online transaction processing (OLTP) database for e-business. IDS delivers the performance, reliability, scalability, and high availability needed for today`s global e-business enterprises.

Currently, many organizations rely on the Shared Disk Secondary (SDS) environment of IDS for their infrastructure to run without any disruptions. Any disruption of this infrastructure translates directly into bottom-line business losses. As an organization's information systems become increasingly integrated and interdependent, the potential impact of failures and outages grows to enormous proportions.

The challenge for IT organizations is to maintain continuous IDS availability in a complex, interconnected, and heterogeneous application environment. The difficulties are significant:

- there are many potential points of failure or disruption
- the interdependencies between components complicates administration
- the infrastructure itself undergoes constant change

Symantec's High Availability solution for IDS is an integrated, out-of-the-box solution that helps to make IDS highly available. This solution combines Symantec's industry-leading, hardware-independent software for storage management and availability with a deep understanding of the Shared Disk Secondary (SDS) feature of the Informix Dynamic Server environment and its essential components. The result is a solution that you can quickly deploy to immediately protect critical Informix Dynamic Server from either planned or unplanned downtime.

Symantec's solution for ensuring Informix Dynamic Server high availability

Symantec offers an end-to-end, fully integrated solution for ensuring high availability of the Informix Dynamic Server environment. Symantec's solution reduces planned and unplanned downtime, simplifies the administration of the complex environment with a single interface, and supports global failover for disaster recovery purposes.
Symantec's High Availability solution for IDS utilizes the following products:

- Veritas Storage Foundation™
- Veritas Storage Foundation Cluster File System™
- Veritas Cluster Server™
- Veritas Cluster Server Agent for Informix Dynamic Server

This white paper explores how Veritas Storage Foundation Cluster File System and Veritas Cluster Server Agent for Informix Dynamic Server help to reduce the downtime of IDS's SDS server feature in failover scenarios.

About Informix Dynamic Sever (IDS)

IDS is a fast and powerful OLTP-oriented relational database management system. It is based on multithreaded Dynamic Scalable Architecture (DSA). This architecture requires fewer processes to take care of multiple database activities. That is the reason multiple users can be added to the database system with the provision to add the necessary resources the server needs to manage those user transactions. DSA is designed to provide efficient resource utilization so you need to have less hardware to support for growing business needs.

The key IDS features include:

- Data portioning
- High-Availability Data Replication (HDR)
- Enterprise Replication (ER)
- Shared Disk Secondary (SDS)
- Remote Standalone Secondary servers (RSS)
- Parallel queries
- High Performance Loader (HPL)
- Backups and restore

About Shared Disk Secondary (SDS) servers

SDS servers access the same physical disk as the primary server. They provide increased availability and scalability without the need for multiple copies of the database. SDS servers can be made available very quickly. Once configured, an SDS server joins an existing system and is ready for immediate use. A little performance degradation may be caused due to multiple SDS servers accessing a single primary server via fully duplex communication.

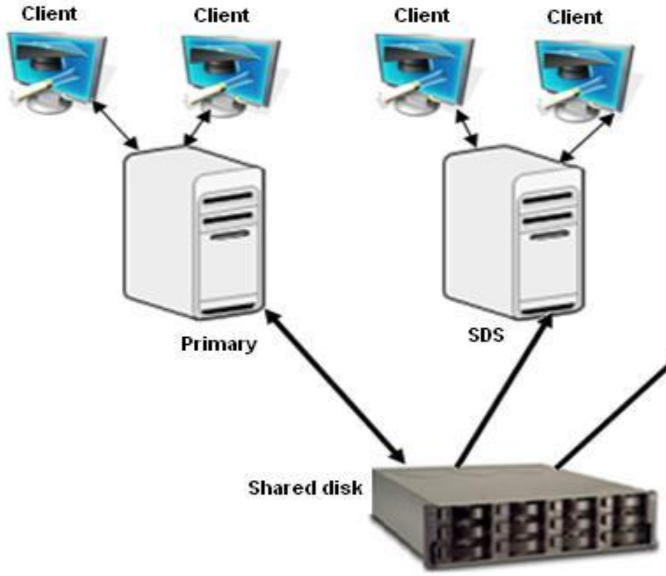


Figure 1: Primary server configured with two SD secondary servers

When the primary server becomes unavailable, any one of the SDS servers can easily be converted into a primary server. When a specified SDS server becomes the new primary server, all the other primary servers automatically recognize the new primary server. Multiple SDS servers also provide load balancing functionality from the primary server.

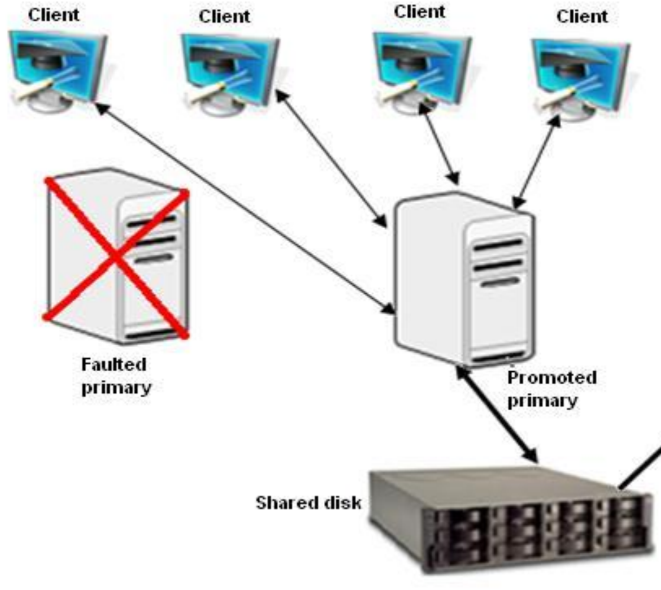


Figure 2: SDS server assuming the role of a primary server

Overview of Cluster File System

The basic functionality of Cluster File System (CFS) is to access shared files from all the cluster nodes that are running with the same mission-critical applications to achieve fast failover in a high availability environment.
The Veritas Storage Foundation CFS contains Veritas CFS, Cluster Volume Manager (CVM), and Cluster Server to support reliable, robust, and scalable shared file solutions.

CFS helps parallel applications to achieve linear scalability and is globally used as a fast failover technique to minimize application downtime in the event of an application server hardware or software failure and malfunction.

About CFS CVM

Cluster File System is an extension to Veritas File System (VxFS). CFS requires a distributed volume manager to allow the disk groups to be shared on all the nodes. CVM provides this functionality and is built on top of Veritas Volume Manager (VxVM). A VxFS file system can be created on volumes as part of this disk group. CFS allows this file system to be mounted on all those nodes in the cluster that have the disk group shared.

Failover is simple to describe, but in practice, is a complex, multi-stage operation. As part of failure detection, varied tasks are performed, including detection of failure, membership calculation, transfer of I/O control, file system verification, database restart, and client reconnection.

CVM facilitates sharing of the disk group among different servers that are part of the same cluster. The notion of a cluster is defined by VCS with group atomic broadcast (GAB) providing cluster membership. Both CFS and CVM depend on GAB to provide consistent cluster membership across all the nodes. When all the nodes of the cluster are running with CVM, the disk group is shared on all the nodes. The failover node already has some degree of control over the volumes and can take over I/O after membership calculation. Similarly, CFS-shared file systems are mounted on all cluster nodes, so file system structural verification and re-mounting on the failover node is unnecessary.

About Veritas Cluster Server

Veritas Cluster Server is the industry's leading cross-platform clustering solution for minimizing application downtime. Through central management tools, automated failover, features to test disaster recovery plans without disruption, and advanced failover management based on server capacity, Cluster Server allows IT managers to maximize resources by moving beyond reactive recovery to proactive management of application availability.

Veritas Cluster Server agent for Informix Dynamic Server

The fundamental functionality of the VCS agent for IDS is:

- to bring online, take offline, and monitor Informix database configured critical process
- to clean the environment for the Informix database application in case of any discrepancies or malfunction
- to switch over the Informix database server application group for planned downtime scenarios

The incredible feature of the VCS agent for IDS is that a single agent provides high availability support to Informix Standard, HDR, and SDS servers in a VCS environment.

Functions of the VCS agent for IDS

The VCS agent for IDS starts, stops, and monitors essential processes of the Informix database server. Additionally, If any critical process fails, the agent cleans the node to remove any system processes and shared resources pertaining to the particular application.

Online

The online operation performs the following tasks:

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

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

2. Verifies that the Informix Dynamic Server is completely functional.
3. Verifies that the Informix Dynamic Server instance is not already online.

Offline

The offline operation performs the following tasks:

1. Promotes one of the SDS servers as primary and takes the faulted primary server offline. The offline operation then initiates the interaction between the new SD primary server and the other SD servers, without a break in connection.

Monitor

The agent provides multiple levels of monitoring of the Informix application to confirm a proper health check:

1. The first level, which is the default level, confirms the essential processes of the Informix database application.
2. The second level, which is optional and is additional to the first level, checks the type of server that is currently running and its exit code.
3. The third level, which is also optional, invokes external program monitoring, thus allowing the user to provide custom scripts for application monitoring.

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:

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

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

2. Performs the following actions, if the instance does not shut down normally:
   a. Kills the remaining processes pertaining to this instance of the Informix Dynamic Server.
   b. Removes any shared memory resources associated with the existing Informix Dynamic Server instance.

Preonline

During detection of an SD primary server failure, a preonline trigger is called on the new target system. This trigger locates the first online SD secondary server and executes the action entry point. The entry point uses the following command to promote the SD secondary server to an SD primary server:

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

After a successful promotion, the preonline trigger fails over the faulted node to the new target and this instance is started as an SD secondary. This instance is then automatically connected to the previously promoted SD primary. Use the following Informix command to confirm the connection:

```
# onstat –g sds verbose
```

Key features of the Preonline trigger are:

- It deduces which type of server is running under its own service group
- If the Preonline trigger interprets any fault with the SDS primary sever, it immediately adapts any one of the existing SDS servers as the new primary server
- It translates previous Informix database SDS primary server as SDS servers after a failover
- It adapts any one of the SDS severs as a primary SDS, even if the primary server goes offline unexpectedly
- It adapts any one of the SDS servers as primary SDS, even if unplanned switchover occurs

The preonline script functions in the following manner:

- Executes the preonline trigger script, which adapts one of the SDS servers to primary in case of planned switchover or a planned offline occurrence. See Figure 3.

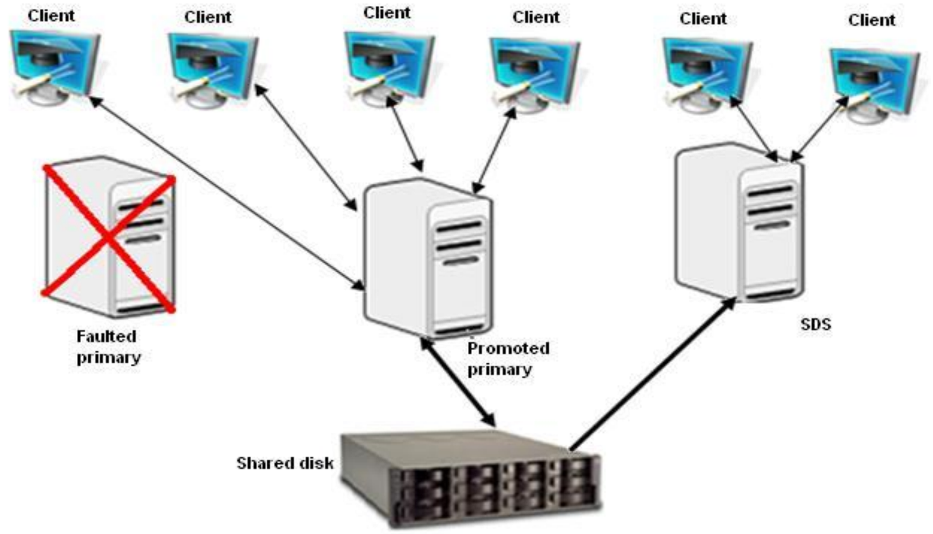


Figure 3: SD secondary server promoted to the role of a primary server

- Faulted SD primary sever can be automatically converted to an SDS server.

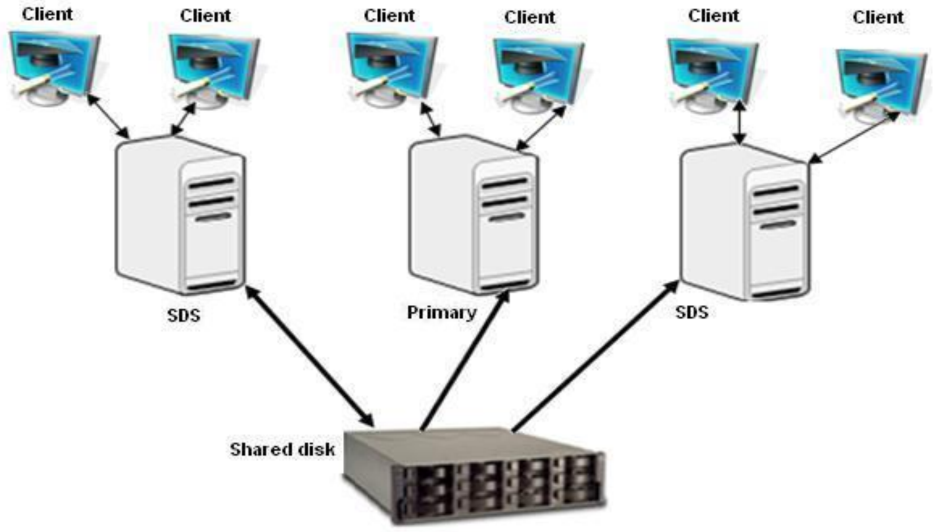


Figure 4: Previous SDS primary converted to SD server

Configuring Informix SDS with Storage Foundation CFS for high availability and fault tolerance

The Informix SD service groups are created on top of the CFS service group in the VCS configuration file. The service groups have an online, local, firm dependency on the CFS service group. The CFS service group is a parallel service group and is online on all the nodes simultaneously. The Informix service groups are defined as a failover service group among various nodes in the cluster.

Installing and configuring Informix SDS on shared disks

To install and configure Informix SDS on shared disks, perform the following tasks:

1. Install Symantec`s Veritas Storage foundation Cluster File system.
2. Create disk group and volume with CFS.
3. Initialize disks for CVM/CFS. You need to initialize the physical disks that will be employed in CVM disk groups. To initialize a disk for CVM:

a. Log on to the master node

b. Use the `vxdiskadm` program to initialize multiple disks, or use the `vxdisksetup` command to initialize one disk at a time. For example: `# /etc/vx/bin/vxdisksetup -i c4t4d0`

4. Create the disk group for the Informix SDS server. Use the `vxdg` command to create disk groups. Use the -s option to specify shared mode, as shown in the following examples:

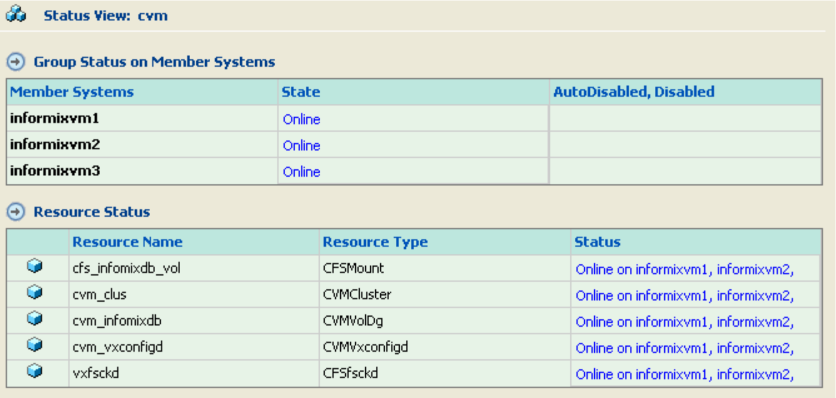
```
# vxdg -s init cfsdg1 c4t4d0

Create the Disk Group Multi-Node package. Use the following command to add the disk group
to the cluster:

# cfsdgadm add cfsdg1 all=sw

Activate the Disk Group

# cfsdgadm activate cfsdg1

Create volumes and add a Cluster File System

# vxassist -g cfsdg1 make vol1 10240m

# newfs -F vxfs /dev/vx/rdsk/cfsdg1/vol1
```

Note: Refer to the *Veritas Storage Foundation CFS User Guide* for more information.

Configuring the VCS agent for IDS

To configure the VCS agent for IDS, create the following service groups and resources:

1. Create a CFS CVM parallel service group that can come online on all CFS nodes simultaneously.



Status View: cvm

Group Status on Member Systems

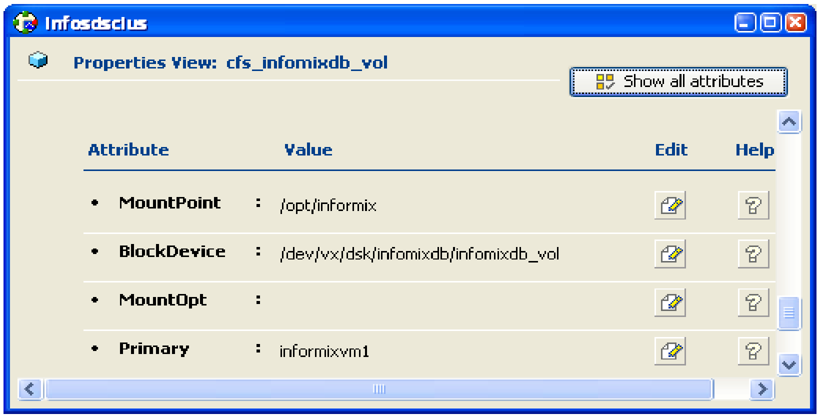
Member Systems	State	AutoDisabled, Disabled
informixvm1	Online	
informixvm2	Online	
informixvm3	Online	

Resource Status

	Resource Name	Resource Type	Status
	cfs_infomixdb_vol	CFSMount	Online on informixvm1, informixvm2,
	cvm_clus	CVMCluster	Online on informixvm1, informixvm2,
	cvm_infomixdb	CVMVolDg	Online on informixvm1, informixvm2,
	cvm_vxconfigd	CVMVxconfigd	Online on informixvm1, informixvm2,
	vxfsckd	CFSfsckd	Online on informixvm1, informixvm2,

Figure 5: CVM parallel disk group and resources to come online on all clustered nodes

2. Create similar mount points on all CFS nodes and mount parallel.

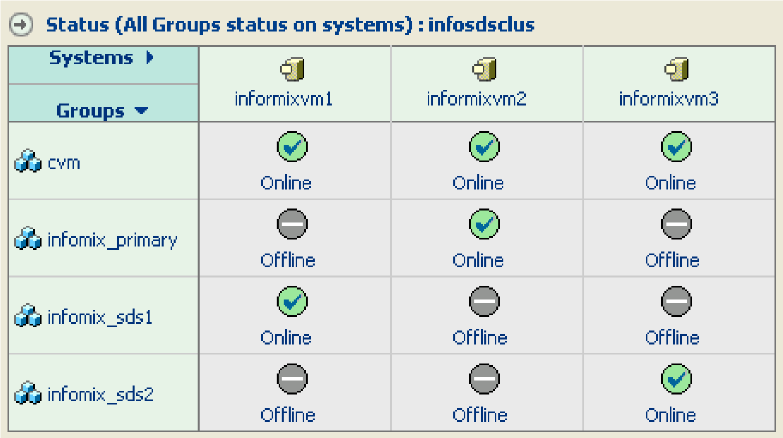


Properties View: cfs_infomixdb_vol

Attribute	Value	Edit	Help
• MountPoint	: /opt/informix		
• BlockDevice	: /dev/vx/dsk/infomixdb/infomixdb_vol		
• MountOpt	:		
• Primary	: informixvm1		

Figure 6: Mount agent properties

3. Create individual instances of Informix SD server service groups with Informix agent resource. Figure 7 shows four service groups, of which, three service groups are configured for the Informix SD server and the fourth is the CVM parallel service group.



Status (All Groups status on systems) : infosdsclus

Systems ▸ Groups ▾	informixvm1	informixvm2	informixvm3
cvm	Online	Online	Online
infomix_primary	Offline	Online	Offline
infomix_sds1	Online	Offline	Offline
infomix_sds2	Offline	Offline	Online

Figure 7: Service groups on all clustered nodes

4. Configure a group level dependency between the Informix SD server service group and the CVM service group.

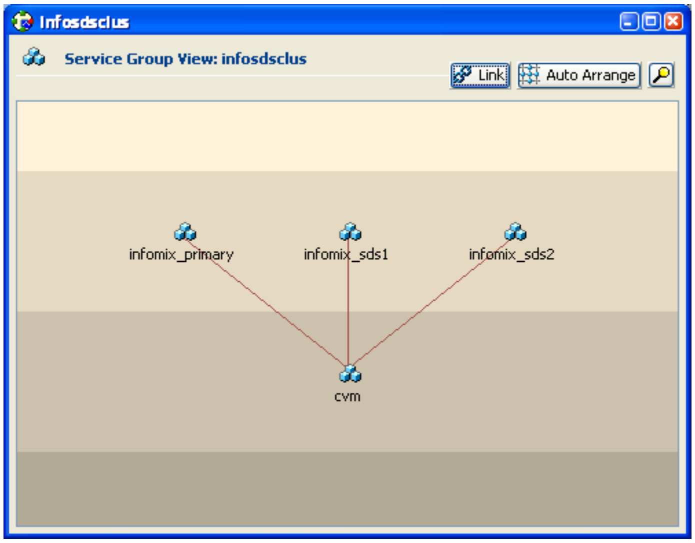
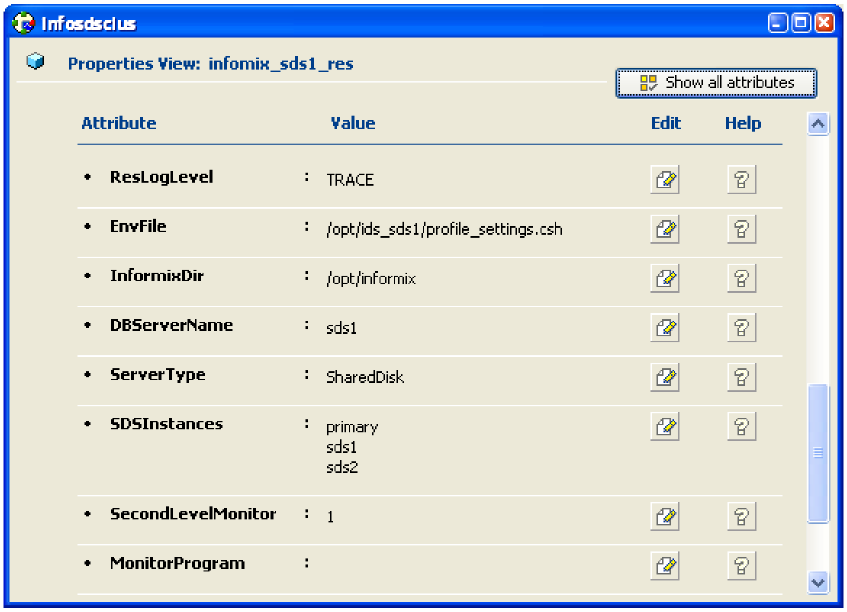


Figure 8: Individual Informix SDS service groups with group dependencies

Agent attribute values

Figure 9 shows the attribute values of the VCS agent for IDS.



Attribute		Value	Edit	Help
• ResLogLevel	:	TRACE		
• EnvFile	:	/opt/ids_sds1/profile_settings.csh		
• InformixDir	:	/opt/informix		
• DBServerName	:	sds1		
• ServerType	:	SharedDisk		
• SDSInstances	:	primary sds1 sds2		
• SecondLevelMonitor	:	1		
• MonitorProgram	:			

Figure 9: VCS agent for IDS configuration for Informix SDS server

Informix SD server agent resource dependencies

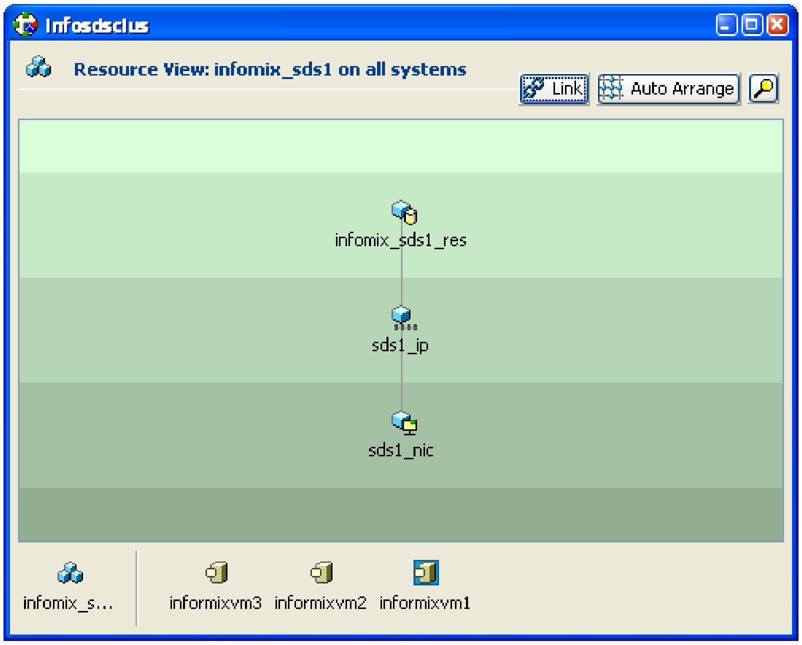


Figure 10: VCS agent for IDS configuration for Informix SDS server

Appendix A: VCS configuration script entries

The complete VCS configuration file for a three-node Informix SD server service groups and resources is as follows.

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

cluster infosdsclus (
        UserNames = { admin = fopHojOlpKppNxpJom }
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        )

system informixvm1 (
        )

system informixvm2 (
        )

system informixvm3 (
        )

group cvm (
        SystemList = { informixvm1 = 0, informixvm2 = 1, informixvm3 = 2 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { informixvm1, informixvm2, informixvm3 }
        )

        CFSMount cfs_informixinformixdb_vol (
                Critical = 0
                MountPoint = "/opt/informix"
                BlockDevice = "/dev/vx/dsk/informixinformixdb/informixdb_vol"
                )

        CFSfsckd vxfsckd (
                )

        CVMCluster cvm_clus (
                CVMClustName = infosdsclus
                CVMNodeId = { informixvm1 = 0, informixvm2 = 1,
                         informixvm3 = 2 }
                CVMTransport = gab
                CVMTimeout = 200
                )

        CVMVolDg cvm_informixdb (
                Critical = 0
                CVMDiskGroup = informixdb
                CVMVolume = { informixdb_vol }
                CVMActivation = sw
                )

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

        cfs_informixdb_vol requires cvm_informixdb
        cvm_clus requires cvm_vxconfigd
        cvm_informixdb requires vxfsckd
        vxfsckd requires cvm_clus

        // resource dependency tree
        //
        //      group cvm
        //      {
        //      CFSMount cfs_informixdb_vol
        //          {
```

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


group informix_primary (
        SystemList = { informixvm1 = 0, informixvm2 = 1, informixvm3 = 2 }
        PreOnline @informixvm1 = 1
        PreOnline @informixvm2 = 1
        PreOnline @informixvm3 = 1
        )

        IP informixprima_ip (
                Critical = 0
                Device = eth4
                Address = "10.209.68.244"
                NetMask = "255.255.252.0"
                )

        Informix informix_prima_res (
                ResLogLevel = TRACE
                EnvFile = "/opt/informix/demo/server/profile_settings.csh"
                InformixDir = "/opt/informix"
                DBServerName = primary
                ServerType = SharedDisk
                SDSInstances = { primary, sds1, sds2 }
                SecondLevelMonitor = 1
                )

        NIC informixprima_nic (
                Critical = 0
                Device @informixvm1 = eth4
                Device @informixvm2 = eth4
                Device @informixvm3 = eth4
                NetworkHosts = { "10.209.68.1" }
                )

        requires group cvm online local firm
        informix_prima_res requires informixprima_ip
        informixprima_ip requires informixprima_nic


        // resource dependency tree
        //
        //      group informix_primary
        //      {
        //      Informix informix_prima_res
        //          {
        //          IP informixprima_ip
        //              {
        //              NIC informixprima_nic
        //              }
        //          }
        //      }


group informix_sds1 (
        SystemList = { informixvm2 = 1, informixvm1 = 0, informixvm3 = 2 }
        PreOnline @informixvm2 = 1
        PreOnline @informixvm1 = 1
        PreOnline @informixvm3 = 1
        )

        IP sds1_ip (
                Critical = 0
                Device = eth4
                Address = "10.209.68.245"
                NetMask = "255.255.252.0"
                )

        Informix informix_sds1_res (
                ResLogLevel = TRACE
                EnvFile = "/opt/ids_sds1/profile_settings.csh"
                InformixDir = "/opt/informix"
                DBServerName = sds1
                ServerType = SharedDisk
                SDSInstances = { primary, sds1, sds2 }
                SecondLevelMonitor = 1
                )

        NIC sds1_nic (
                Critical = 0
                Device @informixvm2 = eth4
                Device @informixvm1 = eth4
                Device @informixvm3 = eth4
                NetworkHosts = { "10.209.68.1" }
                )

        requires group cvm online local firm
        informix_sds1_res requires sds1_ip
        sds1_ip requires sds1_nic


        // resource dependency tree
        //
        //      group informix_sds1
        //      {
        //      Informix informix_sds1_res
        //          {
        //          IP sds1_ip
        //              {
        //              NIC sds1_nic
        //              }
        //          }
        //      }


group informix_sds2 (
        SystemList = { informixvm1 = 0, informixvm3 = 2, informixvm2 = 1 }
        PreOnline @informixvm1 = 1
        PreOnline @informixvm3 = 1
        PreOnline @informixvm2 = 1
        )

        IP sds2_ip (
                Critical = 0
                Device = eth4
                Address = "10.209.68.246"
                NetMask = "255.255.252.0"
                )

        Informix informix_sds2_res (
                ResLogLevel = TRACE
                EnvFile = "/opt/ids_sds2/profile_settings.csh"
                InformixDir = "/opt/informix"
                DBServerName = sds2
                ServerType = SharedDisk
                SDSInstances = { primary, sds1, sds2 }
                SecondLevelMonitor = 1
                )

        NIC sds2_nic (
                Device @informixvm1 = eth4
                Device @informixvm3 = eth4
                Device @informixvm2 = eth4
                NetworkHosts = { "10.209.68.1" }
                )
        requires group cvm online local firm
        informix_sds2_res requires sds2_ip
        sds2_ip requires sds2_nic


        // resource dependency tree
        //
        //      group informix_sds2
        //      {
        //      Informix informix_sds2_res
        //          {
        //          IP sds2_ip
        //              {
        //              NIC sds2_nic
        //              }
        //          }
        //      }
```

Appendix B: Informix type definition script output

```
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 = "Standard"
        str SDSInstances[] = {}
        int SecondLevelMonitor = 0
        str MonitorProgram
```

Appendix C: Informix SD server configuration on Linux/Unix OS in a VCS environment

1. Install Informix database server on shared disk which is configured and mounted with CFS CVM.

   Note: Refer to the Informix technical guides for information on installation of the Informix database on Unix/Linux.

2. Create the Informix user and group, and assign the Informix user to the Informix group.
3. Set the following environment variables before installing the database.

   `INFORMIXDIR`

   `INFORMIXSERVER`

   For example:

   `Export INFORMIXDIR=/opt/informix`

   `Export INFORMIXSERVER=primary`

4. Mount the shared disk with INFORMIXDIR.

To run the UNIX Installer

1. Log in as user root.
2. If you are installing from a CD-ROM, insert the CD into the appropriate drive of your computer and use the mount command to mount the CD-ROM. For more information, see the Mount MAN page.
3. Change directories to the location of the installation media. For example, if you are installing from the CD-ROM and the mount point is **/cdrom**, enter: **`cd /cdrom`**
4. Enter the following installation command:

   `./ids_installer`

5. The Informix UNIX Bundle Installer displays a list of products to install.

   Important: If you select the option to configure a Demo IDS Server, you must also select the option to install Informix Dynamic Server. The Informix UNIX Bundle Installer automatically configures a basic database server that you can use as a template to customize to suit your requirements.

6. At the prompt, enter the directory in which to install products.

   (**$INFORMIXDIR**)

7. Select the product to install.
8. When asked for a Demonstration database server, select **Yes**.
9. When prompted, enter the database server name.
10. The installer configures ($INFORMIXDIR)/etc/onconfig.dbservername
11. The installation script creates additional configuration and log files to support the demo_on database server in $INFORMIXDIR/demo/server.

    For information on the configuration settings for demo_on, see the $INFORMIXDIR/demo/server/profile_settings file.

Testing the installation

```
$ dbaccessdemo
$ dbaccess <dbservername>
```

Configurations for high availability

1. Create the Informix user and groups on all the cluster nodes.
2. Create the Informix home directory on all the nodes.
3. Change the environment variable according to the demoserver name.
4. These parameters should be edited in the $INFORMIXDIR/demo/server/profile_settings file as well.

   For example:

```
INFORMIXSERVER=sds0

DBSERVERNAME=sds0

DBSERVERALIASES=sds0

INFORMIXDIR="/opt/informix"

ONCONFIG=onconfig.sds0

INFORMIXSQLHOSTS="/opt/informix/etc/sqlhosts.sds0"

PATH=${INFORMIXDIR}/bin:${PATH}

export INFORMIXSERVER DBSERVERNAME DBSERVERALIASES INFORMIXDIR ONCONFIG INFORMIXSQLHOSTS PATH
```

5. Change the following parameter in `($INFORMIXDIR)/etc/sqlhosts and sqlhosts.<dbservername>`

```
<dbservername>   ontlitcp   <dbserverhostname>   9088
```

   Example of sqlhosts file entries:

   informixdb ontlitcp informixdb 9088

6. Add the virtual hostname and IP address in all the cluster nodes of the /etc/hosts file.
7. Install the agent with the latest ACClib package.

Informix DB SDS configuration

1. Validate that the parameters in the ‘onconfig’ file are similar to the parameters in the example configuration given below. All the parameters in the ‘onconfig’ file must be configured properly for each instance of the SD Primary and SD server.

   Example configuration for the SD primary server:

```
SDS_ENABLE   1
SDS_TIMEOUT 60
SDS_TEMPDBS  tempspace,/opt/informix/ids_pri/tempspace,2,0,100000
SDS_PAGING                    /opt/informix/ids_pri/pagefile,/opt/informix/ids_pri/pagefile1
```

Example configuration for the SD server:

```
SDS_ENABLE   1
SDS_TIMEOUT 60
SDS_TEMPDBS  tempspace,/opt/informix/ids_sds1/tempspace,2,0,100000
SDS_PAGING /opt/informix/ids_sds1/pagefile,/opt/informix/ids_sds1/pagefile1
```

Similarly, create configurations for each additional SD server.

2. Edit $INFORMIXDIR/etc/sqlhosts.<dbservername>

   ```
   # vi /opt/informix/etc/sqlhosts.sds0
   ```

   On SD primary:

   ```
   informixdb      onsoctcp      sds0  9099
   informixdb1     onsoctcp      sds1  9089
   ```

   On SD secondary:

   ```
   informixdb      onsoctcp      sds0  9099
   informixdb1     onsoctcp      sds1  9089
   ```

3. Make sure that each server on a different port is configured on all of the cluster nodes.

   Example:

   ```
   # vi /etc/services
   sds0      9099/tcp         # Primary server port
   sds1       9089/tcp       # secondary server ports
   ```

4. Edit the /etc/hosts file and add the virtual hostname of the Informix primary and secondary servers.
5. Add the server node IP addresses on all of the cluster nodes.
6. Create the full level 0 backup to faketape2 on the primary server. Execute the following commands.

   i. Start the primary server using the `$ onmode  -ky` command and convert it to an SD primary by using the `$onmode -d set SDS primary <alias name>` command.

   ii. Use the following commands to check the primary and secondary servers.

      Primary:

      ```
      informix@informixvm1:~> onstat  -
      IBM Informix Dynamic Server Version 11.50.FC4TL -- On-Line -- Up 5 days 04:05:05 -
      - 47928 Kbytes
      ```

      Secondary:

      ```
      informix@vmlx15:~> onstat  -
      IBM Informix Dynamic Server Version 11.50.FC4TL -- Read-Only (Sec) -- Up 5 days
      03:57:15 -- 39736 Kbytes
      ```

   iii. Once the servers start syncing, the following messages appear in the message logs:

      Primary log messages:

         DR: Primary server connected

         DR: Primary server operational

      Secondary log messages:

         DR: Trying to connect to primary server

```
DR: Secondary server connected
DR: Secondary server operational
```

About Symantec

Symantec is a global leader in infrastructure software, enabling businesses and consumers to have confidence in a connected world. The company helps customers to protect their infrastructure informatino, and interactions by delivering software and services that address risks to security, availability, compliance, and performance. Headquartered in Cupertino, Calif., Symantec has operations in 40 countries. More information is available at www.symantec.com.

For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 (800) 745 6054.

Symantec Corporation
World Headquarters
20330 Stevens Creek Boulevard
Cupertino, CA 95014 USA
+1 (408) 517 8000
1 (800) 721 3934
www.symantec.com

05/09