IRIS FailSafe™ 2.0 INFORMIX® Administrator’s Guide

Document Number 007-3932-001

CONTRIBUTORS

Written by Susan Ellis, revised by Anita Manders
Illustrated by Dany Galgani
Edited by Rick Thompson
Production by Heather Hermstad and Amy Swenson
Engineering contributions by Gilberto Arnaiz, Rishi Avtar, and Bill Sparks
Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson,
Erik Lindholm, and Kay Maitz

IRIS FailSafe™ 2.0 INFORMIX® Administrator's Guide
Document Number 007-3932-001

Contents

List of Figures

List of Tables

List of Examples

About This Guide

This guide provides information about how to configure IRIS FailSafe 2.0 systems with the IRIS FailSafe 2.0 INFORMIX database option. This option enables INFORMIX database resources to be failed over from one node to another if a component fails. This guide is intended as a supplement to the information about configuring IRIS FailSafe that is described in the *IRIS FailSafe 2.0 Administrator's Guide* .

This guide was prepared in conjunction with Release 2.0 of the IRIS FailSafe INFORMIX option. It describes the IRIS FailSafe INFORMIX software for INFORMIX 7.3 ODS.

Audience

This guide is written for system administrators who are responsible for configuring and administering an IRIS FailSafe system with the optional IRIS FailSafe INFORMIX software. These system administrators must be familiar with INFORMIX configuration.

Structure of This Document

This guide contains the following chapters:

- Chapter 1, "IRIS FailSafe INFORMIX Overview," describes the IRIS FailSafe components that are included in the IRIS FailSafe INFORMIX option. It also provides an overview of the steps required to add INFORMIX databases as highly available services to an IRIS FailSafe cluster.
- Chapter 2, "Configuring IRIS FailSafe for INFORMIX," explains the procedures for adding INFORMIX databases as highly available services.

Related Documentation

For INFORMIX installation information, see the *Informix Installation Guide Version 7.1 for UNIX Products*.

Besides this guide, other documentation for the IRIS FailSafe system includes

- *IRIS FailSafe 2.0 Administrator's Guide*
- *IRIS FailSafe 2.0 Programmer's Guide*
- *IRIS FailSafe 2.0 Netscape Server Administrator's Guide*
- *IRIS FailSafe 2.0 NFS Administrator's Guide*
- *IRIS FailSafe 2.0 Oracle Administrator's Guide*

The IRIS FailSafe reference pages are as follows:

- failsafe(7M)
- ha_cilog(1M)
- ha_cmsd(1M)
- ha_exec2(1M)
- ha_filelock(1M)
- ha_fileunlock(1M)
- ha_fsadm(1M)
- ha_fsd(1M)
- ha_gcd(1M)
- ha_http_ping2(1M) (IRIS FailSafe Web)
- ha_ifdadmin(1M)
- ha_ifd(1M)
- ha_ifmx2(1M) (IRIS FailSafe INFORMIX option)
- ha_macconfig2(1M)
- ha_mom(1M)
- ha_srmd(1M)
- ha_statd2(1M)

Release notes are included with each IRIS FailSafe product. The names of the release notes are as follows:

Table i IRIS FailSafe Release Notes

Release Note	Product
failsafe2	IRIS FailSafe 2.0
failsafe2_nfs	IRIS FailSafe 2.0 NFS
failsafe2_web	IRIS FailSafe 2.0 Netscape
failsafe2_informix	IRIS FailSafe 2.0 INFORMIX
failsafe2_oracle	IRIS FailSafe 2.0 Oracle

Conventions Used in This Guide

These type conventions and symbols are used in this guide:

Italics New terms, manual/book titles, command names and command line arguments, file names, directory names, resource types, variable command line arguments, and variables to be supplied by the user in examples, code, and syntax statements

`Fixed-width type`
Examples of command output and code

`Bold fixed-width type`
Commands and text that you are to type literally in response to shell and command prompts

# IRIX shell prompt for the super user (*root*)

Chapter 1

IRIS FailSafe INFORMIX Overview

This chapter provides information about the INFORMIX database agent that is added to the base IRIS FailSafe 2.0 product by the IRIS FailSafe 2.0 INFORMIX option. It assumes that you are familiar with the basic components of IRIS FailSafe described in Chapter 1 of the *IRIS FailSafe 2.0 Administrator's Guide*. This chapter also lists some important points about setting up INFORMIX databases and lists the steps you must take to add INFORMIX databases to the highly available services that are failed over on an IRIS FailSafe cluster.

This chapter provides information about configuring the IRIS FailSafe INFORMIX database option for use on an IRIS FailSafe system. The major sections in this chapter are as follows:

- "IRIS FailSafe INFORMIX Database Monitoring" on page 1
- "Setting Up INFORMIX Databases for IRIS FailSafe" on page 3
- "Overview of Configuring IRIS FailSafe for INFORMIX" on page 4

IRIS FailSafe INFORMIX Database Monitoring

The IRIS FailSafe INFORMIX option adds a database agent, shown in Figure 1-1, to the IRIS FailSafe base software. The database agent, *ha_ifmx2*, monitors INFORMIX databases regularly. Figure 1-1 shows an example of the message paths in IRIS FailSafe.

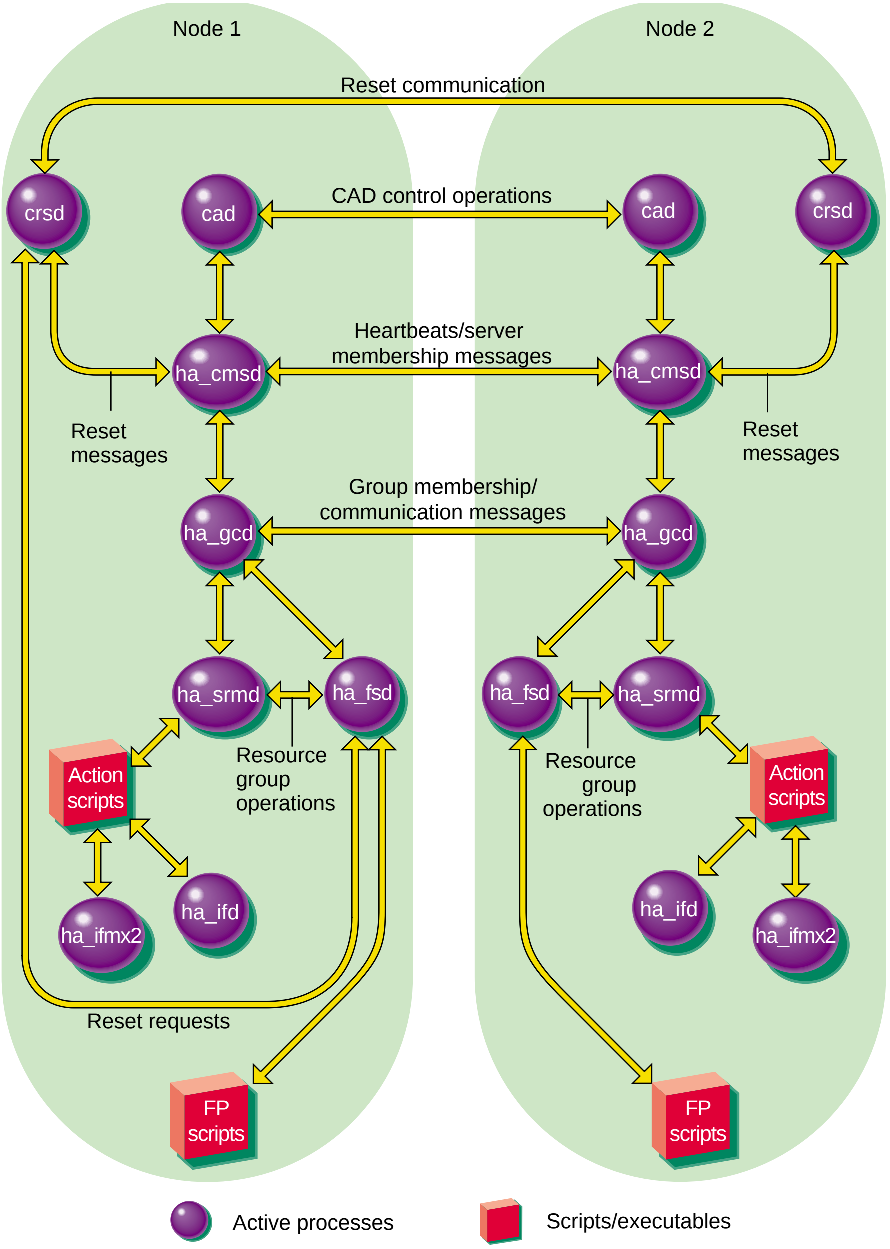

Figure 1-1 Message Paths in IRIS FailSafe

The IRIS FailSafe INFORMIX database agent can perform two types of monitoring:

- Performing an SQL query to detect the state of *Database*.

  The query checks the *sh_mode* from the *sysmaster@sysshmvals* table and reports success if INFORMIX is in one of these modes: initialization, quiescent, fast recovery, archive backup, or online. It reports failure if the server is in shutdown mode or the system is aborting.

- Checking the output of the *onstat* command.

  To check the output of the *onstat* command, the IRIS FailSafe INFORMIX database agent parses the output to look for strings that are specified in the configuration database. The search string is specified as part of the resource definition *search-string*. If it finds any of these strings, it assumes that the INFORMIX database is running.

Setting Up INFORMIX Databases for IRIS FailSafe

When setting up INFORMIX database servers, follow these guidelines:

- IRIS FailSafe requires that all INFORMIX database servers be set up using the *root* account.
- Because IRIS FailSafe INFORMIX software runs as *root*, each INFORMIX database server must be set up so that `root` can access it to monitor it.
- The interactive logging sequence for INFORMIX users is not recommended.

The configuration files involved in setting up INFORMIX such as *$ONCONFIG*, *$INFORMIXDIR/etc/sqlhosts*, */etc/services*, and */etc/hosts* do not require any additional modification for IRIS FailSafe beyond the changes described in Chapter 3 of the *IRIS FailSafe 2.0 Administrator's Guide*.

Overview of Configuring IRIS FailSafe for INFORMIX

To configure an IRIS FailSafe cluster for failover of INFORMIX databases, follow these steps:

1. Install, configure, and test the base IRIS FailSafe software as described in the *IRIS FailSafe 2.0 Administrator's Guide*.
2. Install additional software if required. See the section "Required Software" in Chapter 2.
3. Choose how to configure INFORMIX software and databases on the disks. See the section "Planning INFORMIX Configuration" in Chapter 2.
4. Add INFORMIX information to the configuration database. See the section "Adding INFORMIX Information to the Configuration Database (CDB)" in Chapter 2.
5. Test INFORMIX failover. See the section "Testing the INFORMIX Resource" in Chapter 2.

Chapter 2

Configuring IRIS FailSafe for INFORMIX

This chapter provides information about configuring the IRIS FailSafe 2.0 INFORMIX database option for use on an IRIS FailSafe 2.0 system. The major sections in this chapter are as follows:

- "Required Software" on page 5
- "Planning INFORMIX Configuration" on page 6
- "Adding INFORMIX Information to the Configuration Database (CDB)" on page 9
- "Testing the INFORMIX Resource" on page 16

Required Software

The required software for INFORMIX failover is as follows:

- INFORMIX database software

  See the *Informix Installation Guide Version 7.3 for UNIX Products* for installation instructions.
- Base IRIS FailSafe software

  See the section "Installing the IRIS FailSafe Software" in Chapter 2 of the *IRIS FailSafe 2.0 Administrator's Guide* for a complete list of required base software.
- IRIS FailSafe INFORMIX software
  - The *IRIS FailSafe 2.0 INFORMIX Administrator's Guide* subsystem is *failsafe2_informix.books.book_AG*.
  - The man pages subsystem is *failsafe2_informix.man.man*.
  - The release notes subsystem is *failsafe2_informix.man.relnotes*.
  - The software subsystem is *failsafe2_informix.sw.base*.

Planning INFORMIX Configuration

The following subsections discuss choosing the location of key INFORMIX files and INFORMIX databases. They also describe an example configuration of INFORMIX files and the IRIS FailSafe configuration parameters for this configuration.

Choosing an INFORMIX Configuration

In configuring INFORMIX on an IRIS FailSafe cluster, consider the locations of these components:

- INFORMIX software (installed in *$INFORMIXDIR*, for example */usr/informix*)
- INFORMIX configuration file (specified by *$ONCONFIG*)
- INFORMIX databases

You have the flexibility to use different copies of each component, for example INFORMIX software, for each database server (the INFORMIX software must be identical on both nodes). In addition, the INFORMIX configuration file for one database server can be different on each node. For example, if the two nodes have different amounts of memory, you may want different parameter values in the INFORMIX configuration file on the two nodes. However, the two configuration files must have identical pathnames.

To plan an IRIS FailSafe configuration for INFORMIX, you must answer these questions:

- Do all of the database servers use the same INFORMIX software?
- Is the INFORMIX configuration file identical on each node for each database server?

After answering all of the preceding questions, follow these guidelines for each database server to decide whether to use shared or non-shared disks for each of the INFORMIX components for that server:

- INFORMIX databases must be on one or more XFS file systems or XLV logical volumes that are on shared disks.
- If the INFORMIX configuration file is identical on each node, it can be on a shared disk or replicated on a non-shared disk on each node.
- If the INFORMIX configuration file is not identical on each node, it must be on non-shared disks on each node.
- INFORMIX software can be on a shared disk or replicated on non-shared disks (it must be identical on each node).

Example INFORMIX Configuration

For a simple INFORMIX configuration with one database server called *informix1* whose primary node is *xfs-ha2*, an example configuration is as follows:

INFORMIX software

On a shared disk with *$INFORMIXDIR* set to */usr/informix*.

INFORMIX configuration file

Identical on both nodes and stored in *$INFORMIXDR/etc/onconfig.1*, which is on the same shared disk as the INFORMIX software.

INFORMIX database

On file systems that are on shared disk. The *rootdbs* is mounted at */xfs-ha2fs* on *xfs-ha2*.

INFORMIX Configuration Parameters

Table 2-1 lists the label and configuration parameters for the INFORMIX database in the configuration described in the previous section, “Example INFORMIX Configuration.”

Table 2-1 INFORMIX Resource Type Configuration Parameters

Label or Parameter	**Example Value**	**Comments**
resource-name	*informix1*	Database server name, also known as *$INFORMIXSERVER*.
label	*informix1*	Database server name, also known as *$INFORMIXSERVER*.
release-dir	*/usr/informix*	The database installation directory, also known as *$INFORMIXDIR*.
db-master-fs	*/usr*	The file system on which INFORMIX is installed.
informix-user	*informix*	The UNIX user of INFORMIX.
config-file	*onconfig*	The INFORMIX startup configuration file, also known as *$ONCONFIG*.
monitor-level	*1*	The type of monitoring to perform; you can specify one of the following levels: 1 Uses the onstat command to determine if the database is accessible 2 Posts an SQL query to the database
search-string	*"On-line"*	The value must begin and end with a double quote (") and represents the search string to use on the onstat command. This string is used in an *egrep*(1M) command.
db-probe-time	*30*	The agent monitor interval, specified in seconds.
shutdown-options	*1*	Database shutdown types; you can specify one of the following types: 0 Normal shutdown. 1 Shutdown with nowait and no checkpoint. 2 Shutdown with nowait and checkpoint.
sqlhosts-file	*/usr/informix/sqlhosts.test*	The location of the sqlhosts file.

Table 2-2 shows the configuration parameters that control monitoring. They are specified per database instance, so you can choose different monitoring for each database.

Table 2-2 Configuration Parameters for Monitoring

Parameter	Possible Values	Comment
monitoring-level	*1* *2*	If the value is `1`, the `onstat` command is used for monitoring. If the value is `2`, an SQL query is used.
search-string	*"string*|*string"*	The value must begin and end with a double quote ("). It can contain one *string* or multiple *strings* separated by an "or" symbol (|). See the following examples of strings.

An example of the parameter *search-string* is as follows:

```
search-string = "On-Line|Recovery"
```

In this example, the IRIS FailSafe INFORMIX database agent determines that the database is running if the output contains the *string On-Line* or *Recovery*. For example:

```
# /disk2/server/bin/onstat | egrep "On-Line|Recovery"
INFORMIX Dynamic Server Version 7.30.UC2   -- On-Line -- Up 1 days
00:08:20 -- 9216 Kbytes
...
```

Adding INFORMIX Information to the Configuration Database (CDB)

This section describes procedures that show you how to create the *INFORMIX_DB* resource type, resource, resource group, and configuration database (CDB) that include INFORMIX configuration information. These procedures assume that a CDB has been created, installed, and tested as described in the *IRIS FailSafe 2.0 Administrator's Guide*, and that it does not contain any INFORMIX information. Using the following procedures, you can create the INFORMIX resource type, create the INFORMIX resource information, and test the configuration.

Creating an INFORMIX Resource Type

To create the INFORMIX resource type, this section assumes that you are already familiar with the concepts of resource types. The *INFORMIX_DB* resource type defines the following resource attributes. See Table 2-1 for the definitions of these resource attributes.

- *resource_name*
- *label*
- *release-dir*
- *db-master-fs*
- *informix-user*
- *config-file*
- *monitor-level*
- *search-string*
- *db-probe-time*
- *shutdown-options*
- *sqlhosts-file*

The *INFORMIX_DB* resource type is created at cluster creation time. If this automatic resource creation fails, the administrator must create the resource type before an INFORMIX resource is created. The *INFORMIX_DB* resource type must be installed if you want to add an INFORMIX resource to a cluster that was created before the INFORMIX software was installed.

Example 2-1 Example INFORMIX Installation Process

```
Inst> l *
 View: distribution
 Status: N=new, U=upgrade, S=same, D=downgrade
 Selection: i=install, r=remove, k=keep

 Subsystem Types [bdrpoc]: b=reBoot needed, d=Default, r=Required,
 p=patch, o=overlay, c=Client only

i N failsafe2_informix.books.book_AG [d]  472+ FailSafe INFORMIX
Plugin 2.0 Administrator’s Guide
i U failsafe2_informix.man.man [d]  0 FailSafe INFORMIX Plugin 2.0
release notes
i U failsafe2_informix.man.relnotes [d]  0 FailSafe INFORMIX Plugin
2.0 release notes
i U failsafe2_informix.sw.base [d]  12+ FailSafe INFORMIX Plugin 2.0
base software

Disk space summary (Kbytes):          /       /apps

Current free space               233880     4696292
- Selections net change             488+          0
- Temporary inst overhead          1524+          0
= Minimum free during install    231868     4696292

Final projected free space       233392     4696292

Inst> g
Pre-installation check ..   8%
Checking space requirements ..  16%
Installing/removing files ..  16%
Installing new versions of selected failsafe2_informix.books
subsystems
Installing/removing files ..  49%
Upgrading selected failsafe2_informix.man subsystems
Installing/removing files ..  62%
Upgrading selected failsafe2_informix.sw subsystems
Installing/removing files ..  94%
Running exit-commands ..  99%
Checking dependencies .. 100% Done.
Installations and removals were successful.
You may continue with installations or quit now.

Inst> quit
```

Creating the *INFORMIX_DB* Resource Type

You can use one of the following methods to create the *INFORMIX_DB* resource type if it was not automatically installed:

- Run cluster manager (*cmgr*) and manually create the resource type. For more information, see the *IRIS FailSafe 2.0 Administrator's Guide*.
- Run cluster manager (*cmgr*) and install the resource type, as follows:

```
cmgr> show resource_types installed

NFS
template
Netscape_web
statd
Oracle_DB
MAC_address
IP_address
INFORMIX_DB
filesystem
volume

cmgr> install resource_type INFORMIX_DB in cluster eagan

cmgr>
```

- Use the template scripts supplied with IRIS FailSafe 2.0 located in */var/cluster/cmgr-template/cmgr-create-resource_type*.
- Execute */var/cluster/ha/resource_type/INFORMIX_DB/ create_resource_type* and include the path of the CDB argument and the cluster name.
- Use the **Load Resource Type** GUI task to load the resource type.

Creating an INFORMIX Resource

After you have defined the resource type, the administrator must define the INFORMIX resources based on the resource type. Each resource requires a unique resource name (for example, the *INFORMIX_DB* resource type is the INFORMIX instance name). Then, the administrator must supply the resource parameters. To create the resource, either use cluster manager (*cmgr*) or the FailSafe GUI. You can access the *cmgr* either interactively or by using the template script in */var/cluster/cmgr_templates/cmgr-create-resource-INFORMIX_DB*. If you are using the template script, you must first edit the script, add the resource type attributes, and then execute the script. The following command can be used to access the template script:

```
cmgr> /usr/cluster/bin/cluster_mgr -f
/var/cluster/cmgr_templates/cmgr-create-resource-INFORMIX_DB
```

Example 2-2 Creating the *INFORMIX_DB* Resource

The following example uses `cmgr` to create the *INFORMIX_DB* resource named *informix1*.

```
# cluster_mgr
cmgr> create resource informix1 of resource_type INFORMIX_DB in cluster
eagan
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: shutdown-options
Type Specific Attributes - 2: search-string
Type Specific Attributes - 3: informix-user
Type Specific Attributes - 4: config-file
Type Specific Attributes - 5: monitor-level
Type Specific Attributes - 6: db-probe-time
Type Specific Attributes - 7: db-master-fs
Type Specific Attributes - 8: sqlhosts-file
Type Specific Attributes - 9: release-dir
Type Specific Attributes - 10: label

Resource type dependencies to add:

Resource Dependency Type - 1: IP_address
Resource Dependency Type - 2: filesystem

resource informix1 ? set label to informix1
resource informix1 ? set release-dir to /disk2/server
resource informix1 ? set db-master-fs to /disk2
resource informix1 ? set informix-user to informix
resource informix1 ? set config-file to onconfig.test
resource informix1 ? set monitor-level to 2
resource informix1 ? set search-string to "On-Line|Recovery"
resource informix1 ? set db-probe-time to 30
resource informix1 ? set shutdown-options to 1
resource informix1 ? set sqlhosts-file to /usr/informix/sqlhosts.test
resource informix1 ? done
Successfully created resource informix1

cmgr> modify resource informix1 of resource_type INFORMIX_DB in cluster
eagan
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to modify with set command:

Type Specific Attribute - 1: shutdown-options
Type Specific Attribute - 2: search-string
Type Specific Attribute - 3: informix-user
Type Specific Attribute - 4: config-file
Type Specific Attribute - 5: monitor-level
Type Specific Attribute - 6: db-probe-time
Type Specific Attribute - 7: db-master-fs
Type Specific Attribute - 8: sqlhosts-file
Type Specific Attribute - 9: release-dir
Type Specific Attribute - 10: label

No resource type dependencies to add

resource informix1 ? add dependency /disk2 of type filesystem
resource informix1 ? add dependency 128.162.101.21 of type IP_address
resource informix1 ? done
Successfully modified resource informix1

cmgr> quit
```

Creating an INFORMIX Resource Group

To create a resource group, you must first become familiar with the terms and concepts of FailSafe. A resource group can be created either by the GUI or by the cluster manager (*cmgr*).

To define an effective resource group, you must include all of the resources that the *INFORMIX_DB* resource is dependent on, such as file systems, volumes, and IP addresses. The following example shows how to create a typical resource group:

Example 2-3 Example INFORMIX Resource Group Creation Process

```
# cluster_mgr
Welcome to IRIS FailSafe Cluster Manager Command-Line Interface

cmgr> create resource_group informix in cluster eagan
Enter commands, when finished enter either "done" or "cancel"

resource_group informix1 ? set failover_policy to ordered-in-order
resource_group informix1 ? add resource informix1 of resource_type
INFORMIX_DB
resource_group informix1 ? add resource 128.162.101.21 of resource_type
IP_address
resource_group informix1 ? add resource /disk2 of resource_type
filesystem
resource_group informix1 ? add resource alternate of resource_type
volume
resource_group informix1 ? done
Successfully created resource group informix

cmgr> show resource_groups in cluster eagan

Resource Groups:
        informix

cmgr>
```

Testing the INFORMIX Resource

To ensure that the INFORMIX resource has been correctly configured, you can test individual actions by executing the scripts. Each script, located at */var/cluster/ha/resource_types/INFORMIX_DB*, requires two arguments, an input file and an output file. The contents of these files are the resource names. The scripts will display either a `0` if they are successfully executed or a positive integer that indicates the error type. For more information on error codes, see the *IRIS FailSafe 2.0 Programmer's Guide*.

In the following example, you can test the `start` script by starting the INFORMIX resource with the resource name *informix1*.

```
$ cd /var/cluster/ha/resource_types/INFORMIX_DB
$ echo "informix1" > /tmp/ipfile
$ ./start /tmp/ipfile /tmp/opfile
```

This should start the *informix1* instance, named by the INFORMIX resource *informix1*.

To view the individual script actions, you must edit the script, and add "*set -x*" to the *action* function.

In the following example, to see what the *start-informix*`()` function does, edit the *start* script and add "*set -x*" to the *start* function.

```
#
# Start the resource on the local machine.
# Return HA_SUCCESS if the resource has been successfully started on
# the local machine and HA_CMD_FAILED otherwise. The resource name is
# the informix instance id.
#
start_informix()
{
set -x
 # for all informix resource configured
...
```

You are now ready to run the script. In this example, */dev/null* is the output file specification.

```
heb1# echo "informix1" > /tmp/ipfile
# ./start /tmp/ipfile /dev/null
+ get_informix_info informix1
+ [ 0 -ne 0 ]
+ ha_get_field shutdown-options 1
search-string On-Line|Recovery
informix-user informix
config-file onconfig.test
monitor-level 2
db-probe-time 30
db-master-fs /disk2
sqlhosts-file /disk2/informix/sqlhosts.test
release-dir /disk2/server
label informix1 release-dir
+ [ 0 -ne 0 ]
+ release_dir=/disk2/server
+ informix_probe /disk2/server
+ [ 1 -ne 0 ]
+ ha_get_field shutdown-options 1
search-string On-Line|Recovery
informix-user informix
config-file onconfig.test
monitor-level 2
db-probe-time 30
db-master-fs /disk2
sqlhosts-file /disk2/informix/sqlhosts.test
release-dir /disk2/server
label informix1 informix-user
+ [ 0 -ne 0 ]
+ user=informix
+ ha_get_field shutdown-options 1
search-string On-Line|Recovery
informix-user informix
config-file onconfig.test
monitor-level 2
db-probe-time 30
db-master-fs /disk2
sqlhosts-file /disk2/informix/sqlhosts.test
release-dir /disk2/server
label informix1 label
+ [ 0 -ne 0 ]
+ server=informix1
+ ha_get_field shutdown-options 1
search-string On-Line|Recovery
informix-user informix
config-file onconfig.test
monitor-level 2
db-probe-time 30
db-master-fs /disk2
sqlhosts-file /disk2/informix/sqlhosts.test
release-dir /disk2/server
label informix1 config-file
+ [ 0 -ne 0 ]
+ config=onconfig.test
+ ha_get_field shutdown-options 1
search-string On-Line|Recovery
informix-user informix
config-file onconfig.test
monitor-level 2
db-probe-time 30
```

```
db-master-fs /disk2
sqlhosts-file /disk2/informix/sqlhosts.test
release-dir /disk2/server
label informix1 sqlhosts-file
+ [ 0 -ne 0 ]
+ sqlhosts_file=/disk2/informix/sqlhosts.test
+ [ -x /disk2/server/bin/oninit ]
+ export ONCONFIG=onconfig.test
+ export INFORMIXSERVER=informix1
+ export INFORMIXDIR=/disk2/server
+ export
PATH=/usr/sbin:/usr/bsd:/sbin:/usr/bin:/etc:/usr/etc:/usr/bin/X11:/disk
2/server
+ export INFORMIXSQLHOSTS=/disk2/informix/sqlhosts.test
+ HA_CMD=su informix -c “/disk2/server/bin/oninit”
+ ha_execute_cmd Informix startup on informix1
+ [ 0 -eq 0 ]
+ ha_write_status_for_resource informix1 0
```

Testing the *start* Script

You can use the following procedure to test the *start* script.

1. To test the *start* script, you must shut down the database on the node (if it is already running), as in the following example:

   ```
   # export INFORMIXDIR=/disk2/server
   # export INFORMIXSERVER=informix1
   # export ONCONFIG=onconfig.test
   # export INFORMIXSQLHOSTS=/disk2/informix/sqlhosts.test
   # su informix -c “/disk2/server/bin/onmode -ky”
   ```

   The *INFORMIXDIR*, *INFORMIXSERVER*, *ONCONFIG*, and *INFORMIXSQLHOSTS* configuration parameters are from the *INFORMIX_DB* resource definition. This should stop the INFORMIX instance.

2. To start the INFORMIX instance, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   ```

   ```
   # /var/cluster/ha/resource_types/INFORMIX_DB/start /tmp/ipfile
   /dev/null
   ```

3. You can now check to see if the database is running and if you can access the database. The user can check for the INFORMIX database processes by executing the following command:

   ```
   # ps -ef | grep oninit
   ```

Testing the *stop* Script

You can use the following procedure to test the *stop* script.

1. To test the *stop* script, you must start the INFORMIX database on the node (if it is not already running), as in the following example:

   ```
   # export ONCONFIG=onconfig.test

   # export INFORMIXSERVER=informix1

   # export INFORMIXDIR=/disk2/server

   # export
   PATH=/usr/sbin:/usr/bsd:/sbin:/usr/bin:/etc:/usr/etc:/usr/bin/X11:/d
   isk2/server

   # export INFORMIXSQLHOSTS=/disk2/informix/sqlhosts.test

   # su informix -c "/disk2/server/bin/oninit"
   ```

   The *INFORMIXDIR*, *INFORMIXSERVER*, *ONCONFIG*, and *INFORMIXSQLHOSTS* configuration parameters are from the *INFORMIX_DB* resource definition. This should start the INFORMIX instance.

2. To stop the INFORMIX instance, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   ```

   ```
   # /var/cluster/ha/resource_types/INFORMIX_DB/stop /tmp/ipfile
   /dev/null
   ```

3. You can now check to see if the database is running and if you can access the database. The user can check for the INFORMIX database processes by executing the following command:

   ```
   # ps -ef | grep oninit
   ```

   Check to ensure that the database has shut down.

Testing the *monitor* Script

You can use the following procedure to test the *monitor* script.

1. To test the *monitor* script, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   ```

   ```
   # /var/cluster/ha/resource_types/INFORMIX_DB/start /tmp/ipfile
   /dev/null
   ```

2. As with all scripts, the output is logged to the */var/cluster/ha/log/script_hostname* files. To check that this script has worked, use the *tail*(1) command and execute the script as in the following example:

   ```
   tail /var/cluster/ha/log/script_heb1
   ```

   ```
   Mon Aug 17 11:36:07.161 <anonymous script 23295:0 Unknown:0>
   ./start:

   Mon Aug 17 11:36:07.360 <anonymous script 19943:0 Unknown:0>
   ./start: /usr/cluster/bin/ha_exec2 -s 0 -t /disk2/server/bin/oninit
   exited with status 101

   Mon Aug 17 11:36:07.567 <anonymous script 23081:0 Unknown:0>
   ./start: Failed to find process /disk2/server/bin/oninit

   Mon Aug 17 11:36:07.789 <anonymous script 23119:0 Unknown:0>
   ./start: Informix startup on informix1

   Mon Aug 17 11:36:15.521 <anonymous script 22694:0 Unknown:0>
   ./start: su informix -c "/disk2/server/bin/oninit" exited with
   status 0

   Mon Aug 17 11:38:35.178 <anonymous script 23284:0 Unknown:0>
   monitor: monitor called with /tmp/ipfile and /dev/null

   Mon Aug 17 11:38:37.978 <anonymous script 23182:0 Unknown:0>
   monitor: Checking for informix process (/disk2/server/bin/oninit)

   Mon Aug 17 11:38:38.373 <anonymous script 23027:0 Unknown:0>
   monitor: /usr/cluster/bin/ha_exec2 -s 0 -t /disk2/server/bin/oninit
   exited with status 0

   Mon Aug 17 11:38:38.704 <anonymous script 23168:0 Unknown:0>
   monitor: checking /disk2/server/bin/onstat for On-Line|Recovery

   Mon Aug 17 11:38:40.194 <anonymous script 23086:0 Unknown:0>
   monitor: /disk2/server/bin/onstat | egrep "On-Line|Recovery">
   /var/cluster/ha/tmp/onstat.23271 2>&1 exited with status 0
   ```

Testing the *restart* Script

You can use the following procedure to test the *restart* script.

1. To test the `restart` script, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   ```

   ```
   # /var/cluster/ha/resource_types/INFORMIX_DB/restart /tmp/ipfile
   /dev/null
   ```

Testing the *exclusive* and *probe* Scripts

You can use the following procedure to test the *exclusive* and *probe* scripts.

1. To test the *exclusive* and the *probe* scripts, enter one of the following sets of commands. To test the *exclusive* script, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   ```

   ```
   # /var/cluster/ha/resource_types/INFORMIX_DB/exclusive /tmp/ipfile
   /dev/null
   ```

   To test the *probe* script, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   ```

   ```
   # /var/cluster/ha/resource_types/INFORMIX_DB/probe /tmp/ipfile
   /dev/null
   ```

2. As with all scripts, the output is logged to the */var/cluster/ha/log/script_hostname* files. To check that the script has worked, use the *tail*(1) command and execute the script as in the following example:

   ```
   tail /var/cluster/ha/log/script_heb1

   Mon Aug 17 11:39:35.376 <anonymous script 23191:0 Unknown:0>
   exclusive: exclusive called with /tmp/ipfile and /dev/null

   Mon Aug 17 11:39:37.615 <anonymous script 23240:0 Unknown:0>
   exclusive: Checking for informix process /disk2/server/bin/oninit

   Mon Aug 17 11:39:37.933 <anonymous script 23264:0 Unknown:0>
   exclusive: /usr/cluster/bin/ha_exec2 -s 0 -t
   /disk2/server/bin/oninit exited with status 0

   Mon Aug 17 11:39:38.207 <anonymous script 23218:0 Unknown:0>
   exclusive: resource informix1 exclusive status: RUNNING

   # /var/cluster/ha/resource_types/INFORMIX_DB/probe /tmp/ipfile
   /dev/null
   ```

Testing the Resource Group Failovers

You can test the failover policy by using either *cmgr* or the GUI to move the resource group to another node in the cluster. To ensure that the resource group correctly failed over, use *cmgr* or the GUI to display the resource group states. The following example uses *cmgr* to test the failover policy:

```
cmgr> admin offline resource_group informix in cluster eagan
Resource group is changing state -- please wait


cmgr> admin move resource_group informix in cluster eagan to node cm2
Resource group is changing state -- please wait


cmgr> admin online resource_group informix in cluster eagan
Resource group is changing state -- please wait
```