IRIS FailSafe™ Version 2 Samba Administrator's Guide

007–3988–003

CONTRIBUTORS

Written by Lori Johnson and Anita Manders

Illustrated by Dany Galgani

Edited by Susan Wilkening

Production by Glen Traefald

Engineering contributions by Jeremy Allison, Herb Lewis, Paddy Sreenivasan, Saadia Khan

Cover Design By Sarah Bolles, Sarah Bolles Design, and Dany Galgani, SGI Technical Publications.

New Features in This Guide

This guide supports the IRIS FailSafe 2.2 Samba release and the following new features:

- Samba 2.*x* up through 2.2.4
- Multiple Samba resources in a resource group
- The following Samba server configurations:
  - Standalone Samba server
  - Primary domain controller
  - Member of a Windows NT domain

Record of Revision

Version	Description
001	November 1999 Incorporates information for the IRIS FailSafe 2.0 release.
002	March 2001 Incorporates information for the IRIS FailSafe 2.1 release.
003	March 2002 Incorporates information for the IRIS FailSafe 2.2 Samba release.

Contents

Tables

About This Guide

This guide provides information about how to configure IRIS FailSafe 2.1.*x* systems with the IRIS FailSafe Samba 2.2 option. The Samba option provides backward and forward capability for all IRIS FailSafe 2.1.*x* releases.

This option enables Samba 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 Version 2 Administrator's Guide*.

Audience

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

This book also assumes that you are familiar with the basic components of IRIS FailSafe described in the *IRIS FailSafe Version 2 Administrator's Guide*.

Related Publications

For more information about Samba, see the following:

- *Samba for IRIX Installation and Administration Guide*
- Samba HOWTO collection (included with Samba as `/usr/samba/swat/help/Samba-HOWTO-Collection.html`)
- *Using Samba* by Robert Eckstein, David Collier-Brown, and Peter Kelly (included with Samba in HTML format in the `/usr/samba/swat/using_samba/` directory)

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

- *IRIS FailSafe Version 2 Administrator's Guide*
- *IRIS FailSafe Version 2 Programmer's Guide*

- *IRIS FailSafe 2.0 INFORMIX Administrator's Guide*
- *IRIS FailSafe 2.0 Netscape Server Administrator's Guide*
- *IRIS FailSafe Version 2 NFS Administrator's Guide*
- *IRIS FailSafe 2.0 Oracle Administrator's Guide*

The IRIS FailSafe man pages are as follows:

- `cdbBackup`(1M)
- `cdbRestore`(1M)
- `cmgr`(1M)
- `crsd`(1M)
- `failsafe`(7M)
- `fs2d`(1M)
- `ha_cilog`(1M)
- `ha_cmsd`(1M)
- `ha_exec2`(1M)
- `ha_fsd`(1M)
- `ha_gcd`(1M)
- `ha_http_ping2`(1M) (IRIS FailSafe Netscape Web option)
- `ha_ifd`(1M)
- `ha_ifdadmin`(1M)
- `ha_ifmx2`(1M)(IRIS FailSafe INFORMIX option)
- `ha_macconfig2`(1M)
- `ha_srmd`(1M)
- `ha_statd2`(1M)
- `haStatus`(1M)

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

Release Note	Product
`cluster_admin`	Cluster administration services
`cluster_control`	Cluster node control services
`cluster_services`	Cluster services
`failsafe2`	IRIS FailSafe 2.1 release
`failsafe2_informix`	FailSafe /INFORMIX
`failsafe2_nfs`	FailSafe /NFS
`failsafe2_oracle`	FailSafe /Oracle
`failsafe2_samba`	FailSafe /Samba
`failsafe2_web`	FailSafe /Netscape web
patch_number	FailSafe patch release

Obtaining Publications

To obtain SGI documentation, go to the SGI Technical Publications Library at:

`http://techpubs.sgi.com.`

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	Italic typeface denotes variable entries and words or concepts being defined.

`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a command or directive line.
GUI element	This bold font denotes the names of graphical user interface (GUI) elements, such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, and fields.

This guide uses *FailSafe* as an abbreviation for *IRIS FailSafe*.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, please tell us. Be sure to include the title and document number of the manual with your comments. (Online, the document number is located in the front matter of the manual. In printed manuals, the document number is located at the bottom of each page.)

You can contact us in any of the following ways:

- Send e-mail to the following address:

  `techpubs@sgi.com`

- Use the Feedback option on the Technical Publications Library World Wide Web page:

  `http://techpubs.sgi.com`

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

  Technical Publications
  SGI
  1600 Amphitheatre Pkwy., M/S 535
  Mountain View, California 94043–1351

- Send a fax to the attention of "Technical Publications" at +1 650 932 0801.

We value your comments and will respond to them promptly.

Introduction

This chapter discusses the following:

What is Samba for IRIX?

Samba for IRIX is a suite of programs implementing a subset of the server message block (SMB) protocol, the file sharing protocol used by Microsoft Windows operating systems. Samba for IRIX allows a machine running the IRIX operating system to provide filesystem and printer services to Windows clients.

Samba for IRIX can be configured to allow the IRIX machine to appear as one of the following:

- A standalone workgroup server (sometimes referred to as a *peer-to-peer server*)
- A member server of a Windows NT domain
- A Windows NT primary domain controller

For more information, see Chapter 3, "Editing Configuration Files", page 11, the *Samba for IRIX Installation and Administration Guide*, and the other documentation that comes with Samba for IRIX.

What is FailSafe?

IRIS FailSafe enables applications and services to be highly available, at a fraction of the cost of specialized fault-tolerant systems. FailSafe automatically fails over applications from one system in the cluster to another, in case of failure. In

combination with a RAID or mirrored disk configuration, a FailSafe cluster provides resilience from any single point of failure.

What is FailSafe for Samba?

FailSafe for Samba makes IRIX filesystems highly available for Windows clients. IRIX filesystem can be XFS or CXFS.

FailSafe operates on the concept of resources. A *resource* can be a filesystem, an IP address, or any entity that can be moved from one node to another when a problem (or scheduled downtime) occurs. In the case of Samba, the resource that can be failed over is the ability of a Samba server to respond to service requests on a particular network basic input output system (NetBIOS) name.

In an IRIS FailSafe 2.1 cluster, one or more nodes can export Samba resources. A FailSafe resource group can contain multiple Samba resources (each with a unique NetBIOS name) and a single node in the FailSafe cluster may have multiple resource groups that contain Samba resources. If a node that exports Samba resources fails, another node provides backup service.

The FailSafe for Samba plug-in provides the `Samba` resource type and the following set of action scripts:

`start`	Starts the Samba resource
`stop`	Stops the Samba resource
`monitor`	Tests to see if the Samba resource is running
`exclusive`	Tests to see if the Samba resource is already running
`restart`	Restarts the Samba resource

These scripts are found in the `/var/cluster/ha/resource_types/Samba` directory.

Required Software

The required software for Samba failover is as follows:

- Samba software

  See the *Samba for IRIX Installation and Administration Guide* for more information about Samba.

- Base IRIS FailSafe software

  See the installation chapter of the *IRIS FailSafe Version 2 Administrator's Guide* for a complete list of required base software.

- IRIS FailSafe Samba software
  - The *IRIS FailSafe Version 2 Samba Administrator's Guide* subsystem is `failsafe2_samba.books.book_AG`
  - The man page subsystem is `failsafe2_samba.man.man`
  - The release notes subsystem is `failsafe2_samba.man.relnotes`
  - The base software subsystem is `failsafe2_samba.sw.base`

Overview of Configuring IRIS FailSafe for Samba

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

1. If not previously performed, install, configure, and test the base IRIS FailSafe software as described in the *IRIS FailSafe Version 2 Administrator's Guide*.
2. Install the Samba for IRIX software on the cluster nodes.
3. Install the IRIS FailSafe for Samba software. See "Required Software", page 3.
4. Edit the configuration files. See Chapter 3, "Editing Configuration Files", page 11.
5. Install the `Samba` resource type. The `Samba` resource type will be installed in the cluster database automatically if the FailSafe cluster was created after installing Samba for IRIX software. See "Installing the `Samba` Resource Type", page 17.
6. Create Samba FailSafe resources and add the resources and their dependencies to a resource group. See "Creating a Samba Resource", page 18, and "Creating a Samba Resource Group", page 21.

7. Bring the resource group containing Samba resources online. Enter the following:

   ```
   cmgr> admin online resource_group RG_name in cluster clustername
   ```

   The Windows client should be able to see all of the resources that you have defined under server names.

8. *(Optional)* If you are running with encrypted passwords from the `smbpasswd` file and you want all Samba resources to use the same `smbpasswd` file, create a symbolic link to the default `smbpasswd` file. For example, for a Samba resource named `server2` with a filesystem dependency of `/fs2`, you would execute the following commands:

   ```
   # cd /fs2/.samba/server2/private/
   # ln -s /usr/samba/private/smbpasswd smbpasswd
   ```

   If you make changes to the `smbpasswd` file, you must copy it to all nodes that will be able to serve this Samba resource.

9. Test the Samba failover. See Chapter 5, "Testing FailSafe for Samba", page 23.

Basic Samba Configuration Information

Samba configuration is controlled by the `smb.conf`(4) file. For complete information about the layout of the `smb.conf` file and a description of all available parameters, see the `smb.conf`(4) man page.

This chapter discusses the following:

Format of the `smb.conf` File

The `smb.conf` file contains sections and parameters that pertain to each section. Section names are contained within square brackets. Parameters consist of a parameter name and a value. The format of the file is as follows:

[*section_name*]
parameter_name1 = *value1*
parameter_name2 = *value2*

The `global` section contains parameters that control the overall behavior of Samba, such as the workgroup name and the NetBIOS name of the server. Other sections define *shares* (printers or directories in a filesystem) that will be made available to clients by Samba.

Parameters Set by FailSafe

The following is a brief description of the parameters in the `global` section that are set by FailSafe Samba.

NetBIOS Name

The following parameter determines Samba's primary *NetBIOS name* (the name by which clients can refer to the server):

```
netbios name = name
```

Legal names contain only letters, digits, hyphens, underscores, and periods. They must be 15 characters or fewer in length.

Interfaces

The `interfaces` parameter allows Samba to support multiple IP interfaces. It takes the following format, where *IP* is a dotted decimal IP address and *netmask* is a dotted decimal netmask such as 255.255.255.0:

```
interfaces = IP1/netmask1 IP2/netmask2
```

The Samba daemons will typically attempt to bind to all interfaces in the system. To force the Samba daemons to only bind to the interfaces specified in the `interfaces` parameter, you would add the following parameter:

```
bind interfaces only = yes
```

Password, Log, and Database Files

The following parameters specify non-default locations for the password file, log files, and database files:

```
smb passwd file = password_path
log file = log_path
lock file = lock_path
```

Including Information from Other Files

The `include` parameter allows Samba to include lines from another file as if they had been entered into the `sbm.conf` file at the location of the `include` parameter. (In other words, the included file replaces the `include` parameter line in the `smb.conf` file.) The value of this parameter can contain *macros* (characters that are expanded by the Samba daemons based on how the client contacts the daemon).

The macro used by FailSafe Samba is `%L`, which expands to the NetBIOS name of the server that the client is contacting:

```
include = /usr/samba/lib/smb.conf.%L
```

Using this parameters allows FailSafe Samba to split the `smb.conf` file into two separate files:

- The first file contains the global parameters used by all resources.
- The second file contains any global parameters that may need to be different depending on the NetBIOS name of the server as well as the sections that define the shares to be made available to clients.

How Does Samba Failover Work?

In order to fail over the Samba resources, separate Samba daemons that are bound to different interfaces are started for each Samba resource. Each of these instances of Samba daemons use a different `smb.conf` configuration file. The `chkconfig` variable `samba` should be off to prevent the system from trying to start Samba on reboot.

Note: If there is to be a normal instance of Samba running that is not controlled by FailSafe, the `smb.conf` file must contain an `interfaces` parameter that does not specify any of the interfaces used by the FailSafe Samba resources and must contain the following parameter:

```
bind interfaces only= yes
```

For versions of Samba for IRIX prior to version 2.2.4, the `/etc/init.d/samba` script should not be used to start or stop this instance of Samba because it would also kill the daemons started by FailSafe. For version 2.2.4 or later, you may set the `chkconfig` variable `samba` to `on` and allow the instance of Samba that is not controlled by FailSafe to be automatically started by the `/etc/init.d/samba` script.

When FailSafe starts up, the `smb.conf` file for each Samba resource is constructed dynamically from the following template file:

```
/usr/samba/lib/smb.conf.template
```

This template file must contain the standard Samba `global` setup for the workgroup, but it **must not** contain any of the following `smb.conf` parameters:

- `smb passwd file`
- `log file`
- `lock dir`
- `netbios name`
- `bind interfaces only`
- `interfaces`

These lines in the `smb.conf` file are automatically added as FailSafe starts the Samba resources and corresponding lines in the `smb.conf` template file will be ignored. You should not define any nonglobal (share) names in this template file.

A Samba NetBIOS name must be defined for each Samba resource. For example, consider a machine named `hans1` with the following characteristics:

- A Samba resource name of `server1`
- A `filesystem` dependency of `/fs1`
- An `IP_address` resource dependency with an HA-IP of 123.45.6.78 and a netmask of 255.255.255.0

For this system, when FailSafe first brings the resource online it creates the following directories (if they do not already exist):

```
/fs1/.samba/server1
/fs1/.samba/server1/locks
/fs1/.samba/server1/log
/fs1/.samba/server1/private
```

Next, FailSafe generates the `/fs1/.samba/server1/smb.conf` configuration file by first copying the contents of the `/usr/samba/lib/smb.conf.template` file and then appending the following lines to the generated `smb.conf` file:

```
smb passwd file = /fs1/.samba/server1/private/smbpasswd
log file = /fs1/.samba/server1/log/log.%m
lock dir = /fs1/.samba/server1/locks
bind interfaces only = yes
netbios name = server1
interfaces = 123.45.6.78/255.255.255.0
include=/usr/samba/lib/smb.conf.%L
```

The `include=/usr/samba/lib/smb.conf.%L` line refers to a file that describes the shares that Samba serves out for a specific NetBIOS name. At run time, the `%L` is expanded into the NetBIOS name that the client is using to refer to the server. As the FailSafe administrator, you must create one of these files for each NetBIOS name and place them in the `/usr/samba/lib` directory. For this example, you would create a file named `/usr/samba/lib/smb.conf.server1`.

The `smb.conf` file above tells Samba to respond to connection requests on the given IP address under the name `server1`, and to register that name via broadcast on the given interface.

FailSafe then starts the Samba daemons, specifying that they use the configuration file just created.

Next, consider a second FailSafe node named `hans2` with the following characteristics:

- a Samba resource name of `server2`
- A `filesystem` dependency of `/fs2`
- An `IP_address` resource dependency with an HA-IP of 123.45.6.90 and a netmask of 255.255.255.0

For this system, when FailSafe first brings the resource online it creates the following directories (if they do not already exist):

```
/fs2/.samba/server2
/fs2/.samba/server2/locks
/fs2/.samba/server2/log
/fs2/.samba/server2/private
```

Next, FailSafe generates the `/fs2/.samba/server2/smb.conf` configuration file by first copying the contents of the `/usr/samba/lib/smb.conf.template` file and then appending the following lines to the generated `smb.conf` file:

```
smb passwd file = /fs2/.samba/server2/private/smbpasswd
log file = /fs2/.samba/server2/log/log.%m
lock dir = /fs2/.samba/server2/locks
bind interfaces only = yes
netbios name = server2
interfaces = 123.45.6.90/255.255.255.0
include=/usr/samba/lib/smb.conf.%L
```

FailSafe then starts the Samba daemons, specifying that they use the configuration file just created.

In this example, Windows clients requesting a connection to `server1` would access the resources on `hans1`, and those clients requesting a connection to `server2` would access the resources on `hans2`.

Now, consider what happens when the machine `hans2` fails and `hans1` must take over the resource. Once `hans2` has failed and the `/fs2` filesystem and the HA-IP address 123.45.6.90 have been failed over to `hans1`, FailSafe will recreate the `/fs2/.samba/server2/smb.conf` file and another `smdb` and `nmdb` daemon will be started. These new daemons will point to the configuration file that was created on `/fs2`. The Windows clients that now connect to the NetBIOS name `server2` will access the machine `hans1` instead of `hans2`, although the change in physical servers will be transparent to them.

Note: The change in physical servers will not be transparent to clients that had active connections to server message block (SMB) resources on `server2`. They must drop these resource handles and reconnect. The success of this procedure depends on what operation the Windows clients were performing.

Editing Configuration Files

This chapter contains a summary of the basic procedure to edit the configuration for a Samba server. It also provides examples that show the additional steps needed for the following configurations:

- "Example: Member Server of a Windows NT Domain", page 13
- "Example: Windows NT Primary Domain Controller", page 15

(There are no additional steps for standalone Samba servers.)

Summary of the Editing Procedure

This section provides a summary of the editing procedure.

To edit configuration files, do the following:

1. Set up the contents of the `/usr/samba/lib/smb.conf.template` file for each server in the cluster. (An example is provided with the plugin.) This file should be edited to contain the correct workgroup for the Samba server and any site-specific `global` section changes that are needed. Copy this file to all nodes of the cluster that will be serving the Samba resource.

   Note: The `smb.conf.template` file should contain only the `global` section header and parameters in the `global` section. It must not contain any other sections that define shares. The following parameters should not be included because they will be added by FailSafe:

   - `smb passwd file`
   - `log file`
   - `lock dir`
   - `netbios name`
   - `bind interfaces only`
   - `interfaces`

   ---

2. Create the `/usr/samba/lib/smb.conf.`*servername* files, where *servername* is the NetBIOS name of the resources that are exported by the FailSafe cluster. (The NetBIOS name is the Samba resource name.)

Each of these files should contain sections for share definitions that each NetBIOS name will offer to clients. The pathnames in each section should exist under the highly available filesystem resource on which the Samba resource depends.

In other words, the area of the filesystem being shared by Samba should be able to be failed over from one node to another node. Any `global` parameters that are to differ from the template file can be added to the start of this file before any share sections.

3. Copy these `/usr/samba/lib/smb.conf.`*servername* files to all the nodes of the cluster that will be serving the Samba resources, so that all nodes are able to share out all the resources.

   For example, if there is a two-node cluster with the following:

   - A Samba resource named `server1` in a workgroup named `workgroup1` with the HA filesystem dependency `/fs1`
   - A Samba resource named `server2` in a workgroup named `workgroup2` with the HA filesystem dependency `/fs2`

   The two files might look like the following:

   Contents of /usr/samba/lib/smb.conf.server1:

```
workgroup = workgroup1
[fs1]
        comment = failsafe filesystem1
        path = /fs1
        read only = no
        guest ok = yes
```

   Contents of /usr/samba/lib/smb.conf.server2:

```
workgroup = workgroup2
[fs2]
        comment = failsafe filesystem2
        path = /fs2
        read only = no
        guest ok = yes
```

Note: For more information about the meaning of the various parameters, see the `smb.conf`(4) man page.

The `workgroup` parameter is normally set in the `smb.conf.template` file. However, because each of these resources is in a different workgroup, the `workgroup` parameter is entered at the beginning of each `smb.conf.servername` file; this action allows the value in the `smb.conf.template` file to be overridden by each resource. If all resources are in the same workgroup, you do not need to add the workgroup parameter to these files.

If resource `server2` is failed over to `hans1`, then as filesystem `/fs2` becomes mounted on `hans1`, the Samba daemons will be started that will start serving out the SMB share `fs2` under the NetBIOS name `server2`. You would now see two sets of Samba daemons running on the machine `hans1` (inline whitespace deleted for readability):

```
# ps -ef | grep mbd
root   6193606  1  0   Feb 08 ?  0:14 /usr/samba/bin/nmbd -D -s /fs1/.samba/server1/smb.conf
root   6199848  1  0   Feb 08 ?  0:00 /usr/samba/bin/smbd -D -s /fs1/.samba/server1/smb.conf
root   7689487  1  0   Feb 08 ?  0:00 /usr/samba/bin/nmbd -D -s /fs2/.samba/server2/smb.conf
root   7690286  1  0   Feb 08 ?  0:00 /usr/samba/bin/smbd -D -s /fs2/.samba/server2/smb.conf
```

Example: Member Server of a Windows NT Domain

This section contains a summary of the steps required to configure a node as a member server of a Windows NT domain. See the section entitled "security = domain in Samba 2.x" in the Samba HOWTO collection for detailed instructions. The HOWTO collection is available at `/usr/samba/swat/help/Samba-HOWTO-Collection.html`.

The example used in the summary assumes the following:

- A Samba resource named `server1`.
- A filesystem dependency of `/fs1`.
- A domain named `DOM`.
- A primary domain controller named `DOMPDC`.
- An account with administrative rights exists on the PDC with the name `administrator` and password of `password`,
- Not all Samba resources in the cluster are operating as a member of a Windows NT domain. (If all Samba resources were to be the running in the same server

configuration mode, the changes specified in step 1 could be made to the `/usr/samba/lib/smb.conf.template` file.)

To configure a node as a member server with the above assumptions, you would do the following:

1. Add the following lines to the beginning of the `/usr/samba/lib/smb.conf.server1` file:

```
workgroup = dom
security = domain
encrypt passwords = yes
password server = dompdc
```

Note: The `password server` line is not required if a WINS server has been specified in the `smb.conf.template` file and the domain controller specifies the same WINS server.

---

2. Start the Samba resource by adding it to a resource group as specified in "Creating a Samba Resource Group", page 21. This will create the default directories and files required.

3. Join the domain by executing the following command:

```
root# /usr/samba/bin/smbpasswd -c /fs1/.samba/server1/smb.conf -j DOM -r DOMPDC -Uadministrator%password
```

The `smbpasswd` command for versions of Samba prior to 2.2.4 will report joining the wrong domain name if the `workgroup = dom` line is not in the main `smb.conf` file.

Note: The `smbpasswd` command in versions of Samba prior to 2.2.4 does not allow the `-c` option to specify the location of the `smb.conf` file. If you are using an older version, you must copy the generated `smb.conf` file to `/usr/samba/lib/smb.conf` before running the command. Additionally, the `-U` option does not work in versions of Samba prior to 2.2.4. You must first create the machine account on the Windows NT domain controller using the **Server Manager for Domains** program from the Windows NT **Administrative Tools** menu and then execute the above `smbpasswd` command without the `-U` option.

Example: Windows NT Primary Domain Controller

The following is a summary of the steps required to configure a node as a Windows NT primary domain controller (PDC). For more details, see the section entitled "How to Configure Samba 2.2 as a Primary Domain Controller" in the Samba HOWTO collection. This section is also available as an individual document at `/usr/samba/swat/help/Samba-PDC-HOWTO.html`.

The example in the summary assumes the following:

- A Samba resource named `server1`.
- A filesystem dependency of `/fs1`.
- A domain named `DOM`.
- There is more than one Samba resource in the cluster. (If there was only one Samba resource, the changes specified in step 1 could be made to the `/usr/samba/lib/smb.conf.template` file.)

To configure a node as a primary domain controller with the above assumptions, you would do the following:

1. Add the following lines to the beginning of the `/usr/samba/lib/smb.conf.server1` file:

```
workgroup = dom
security = user
encrypt passwords = yes
os level = 34
local master = yes
preferred master = yes
domain master = yes
domain logons = yes

[netlogon]
comment = The domain logon service
path = /fs1/samba/logon
guest ok = no
read only = yes
browsable = no
```

You could also add other parameters such as `logon path`, `logon drive`, `logon home`, and `logon script` before the `netlogon` section. If roving profiles were desired, you could also add a `profile` section.

2. Ensure the `/fs1/samba/logon` directory exists.
3. Start the Samba resource by adding it to a resource group, as specified in "Creating a Samba Resource Group", page 21. This will create the default directories and files required.
4. If you have Windows NT clients (or Samba clients) that will be joining the domain, you must create machine trust accounts. These accounts may be created automatically by Samba if the `add user script` parameter is set in the `smb.conf` file. There is a sample command included in the `/usr/samba/lib/smb.conf.template` file that is installed by the FailSafe Samba software, but it is commented out by default; you can remove the semicolon (`;`) at the beginning of the line to enable it.

   Alternately, you may wish to create these accounts manually. For example, if you want the machine with a NetBIOS name of `server2` to join the domain, do the following:

   a. Add the `server2` machine name to the `/etc/passwd` file. (The machine name is the NetBIOS name followed by a `$` character).

   b. Use the `smbpasswd` command to add the machine trust account to the `smbpasswd` file by executing the following command:

```
root# /usr/samba/bin/smbpasswd -c /fs1/.samba/server1/smb.conf -a -m server2
```

Adding Samba to the Cluster Database

This chapter discusses the following:

Note: This chapter assumes that you are already familiar with the concepts of resource types. If not, see *IRIS FailSafe Version 2 Administrator's Guide*.

Installing the Samba Resource Type

If you have the FailSafe Samba software installed before you create a FailSafe cluster, the Samba resource type will be automatically installed at cluster creation time. However, if you already have a cluster before installing the FailSafe Samba software, you must manually install the resource type.

To determine if the resource type has been installed, enter the following cmgr(1M) command:

```
show resource_types in cluster clustername
```

For example, the following output shows that the Samba resource type is not present:

```
cmgr> show resource_types in cluster eagan
NFS
template
Netscape_web
statd
statd_unlimited
Oracle_DB
MAC_address
IP_address
INFORMIX_DB
filesystem
volume
```

You can also display this information with the FailSafe GUI.

To install the `Samba` resource type, use the following command:

`install resource_type Samba in cluster` *clustername*

For example:

```
cmgr> install resource_type Samba in cluster eagan
```

Creating a Samba Resource

After you have installed the resource type, you must define the individual Samba resources based on the resource type. Each Samba resource requires a unique resource name (the NetBIOS name). Then you must supply the resource parameters. To create the resource, either use the `cmgr`(1M) command, the `cmgr-create-resource-Samba` scripts, or the FailSafe GUI.

Resource Requirements

Table 4-1 lists the configuration parameters that must be defined within the FailSafe Samba resource definition.

Table 4-1 Samba Configuration Parameters

Category	Parameter	Description
Resource name	*resource-name*	Specifies the name of the Samba resource; this must be the NetBIOS name used to access the server.
Resource attribute	`monitor-level`	Defines the monitoring action performed by the `monitor` action script: • Level 1 checks for the existence of the `smbd` and `nmbd` processes. • Level 2 checks for the correct responses from `smbd` and `nmbd`. • Level 3 checks for the existence of `nmbd` and for the correct responses from `smbd`.
Resource dependency	`filesystem`	Specifies the name of the highly available filesystem that this particular Samba resource will use. The value specified for `filesystem` is the location where FailSafe Samba will create directories to contain the Samba `smb.conf` configuration file, log files, and various database files. Any directories that Samba will be making available as highly available shares should also be contained on this filesystem. No checks are made to ensure that the paths exported in the `smb.conf` file actually exist on this filesystem.
Resource dependency	`IP_address`	Specifies the highly available IP (HA-IP) address to which the NetBIOS name belongs.

Interactive `cmgr` Example

The following shows an interactive example of using `cmgr` to define a Samba resource named `server1`:

```
root@hans1:~ # /usr/cluster/bin/cmgr
Welcome to IRIS FailSafe Cluster Manager Command-Line Interface

cmgr> define resource server1 of resource_type Samba in cluster interop-ha
Enter commands, when finished enter either "done" or "cancel"
```

```
Type specific attributes to create with set command:

Type Specific Attributes - 1: monitor-level

Resource type dependencies to add:

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

resource /server1 ? set monitor-level to 2
resource /server1 ? add dependency /fs1 of type filesystem
resource /server1 ? add dependency 163.154.48.34 of type IP_address
resource /server1 ? done
Successfully modified resource server1

cmgr> show resource server1 of resource_type Samba

monitor-level: 2

Resource dependencies
IP_address 163.154.48.34
filesystem /fs1

cmgr> exit
```

Script to Create a Resource

Create the FailSafe Samba resource. The following `cmgr` script creates a Samba resource:

```
define resource NetBIOS_name of resource_type Samba in cluster cluster_name
     set monitor-level to 1|2|3
     add dependency IP_address_resource_name of type IP_address
     add dependency filesystem_resource_name of type filesystem
done
```

Creating a Samba Resource Group

To create an effective resource group, you must include all of the resources that the Samba resource is dependent on, such as filesystems, volumes, and IP addresses. The following example shows the creation of a typical resource group:

```
root@hans1:~ # /usr/cluster/bin/cmgr
Welcome to IRIS FailSafe Cluster Manager Command-Line Interface

cmgr> define resource_group hans1 in cluster interop-ha
Enter commands, when finished enter either "done" or "cancel"

resource_group hans1 ? set failover_policy to hans1-primary
resource_group hans1 ? add resource 163.154.48.34 of resource_type IP_address
resource_group hans1 ? add resource /fs1 of resource_type filesystem
resource_group hans1 ? add resource vol1 of resource_type volume
resource_group hans1 ? add resource server1 of resource_type Samba
resource_group hans1 ? done
Successfully defined resource group hans1

cmgr> show resource_group hans1 in cluster interop-ha

Resource Group: hans1
        Cluster: interop-ha
        Failover Policy: hans1-primary

Resources:
        server1 (type: Samba)
        163.154.48.34 (type: IP_address)
        /fs1 (type: filesystem)
        vol1 (type: volume)

cmgr> exit
```

Testing FailSafe for Samba

This chapter discusses the following:

Testing the Samba Resource

To ensure that the `Samba` resource has been correctly configured, you can test individual actions by executing the scripts. This assumes that all of the dependent resources have already been started. The easiest way to test the new `Samba` resources is to create a resource group that contains all of the dependent resources but does not contain the new `Samba` resource. You can then start this resource group and test the individual `Samba` action scripts.

Each script, located at `/var/cluster/ha/resource_types/Samba`, requires two arguments, an input file and an output file. The content of the input file is the resource name. The scripts will display 0 if they are successfully executed, or display a positive number that indicates the error type. For more information on error codes, see the *IRIS FailSafe Version 2 Programmer's Guide*.

In the following example, you can test the `Samba` resource named `samba8`. Test each script by starting with the following commands:

```
$ cd /var/cluster/ha/resource_types/Samba
$ echo samba8 > /tmp/ipfile
```

You can then execute each action script with the following command:

```
$ ./actionscript /tmp/ipfile /tmp/opfile
```

where *actionscript* is one of the following action script names:

- `start`
- `stop`
- `monitor`

- `exclusive`
- `restart`

(For more information about the action scripts, see "What is FailSafe for Samba?", page 2.)

For example:

```
$ ./monitor /tmp/ipfile /tmp/opfile
```

After executing each script, verify that it worked correctly by verifying the output it generates in the `/var/cluster/ha/log/script_`*hostname* file.

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

Testing Resource Group Failovers

After the scripts have been successfully tested, you can add the Samba resource to the resource group that contains all the dependent resources. You can test the failover policy by using either `cmgr` or the FailSafe GUI to move the resource group to another node in the cluster. To ensure that the resource group correctly failed over, use the `cmgr` or GUI to display the resource group states. The following example uses `cmgr` to test the failover policy:

```
cmgr> admin online resource_group Samba in cluster eagan
cmgr> admin move resource_group Samba in cluster eagan to node cm2
cmgr> admin offline resource_group Samba in cluster eagan
```

Index