

Tru64 UNIX Best Practice

Migrating from Automount to AutoFS in TruCluster Server

This Best Practice describes how to migrate from Automount to AutoFS in a TruCluster™ Server Version 5.1 or higher environment.

Contents

Migrating from Automount to AutoFS in a TruCluster Server Environment

Is This Best Practice Right for You?	1
Before You Begin	2
Applying the Best Practice	2
Migrating from Automount to AutoFS	2
Migrating Without a Reboot	2
Migrating When Rebooting a Cluster Member	6
Migrating When Rebooting the Cluster	9
Verifying Success	11
Troubleshooting	11
Comments and Questions	13
Legal Notice	14

Migrating from Automount to AutoFS in a TruCluster Server Environment

The `autofs` daemon automatically and transparently mounts and unmounts NFS file systems on an as-needed basis. Like the `automount` daemon, it provides another alternative to using the `/etc/fstab` file for mounting NFS file systems on client machines. However, AutoFS is more efficient than the `automount` daemon because it requires less communication between the kernel and the user space daemon. The `autofs` daemon also provides higher availability than the `automount` daemon.

Is This Best Practice Right for You?

Not all Best Practices apply to all configurations, so you must be sure that it is appropriate for your system and circumstances. To use this Best Practice, you must already be using Automount in a TruCluster Server and meet the requirements described in the following table:

Requirement	Description
Operating System	Tru64™ UNIX Version 5.1 or higher and TruCluster Server Version 5.1 or higher.
Map Files	This Best Practice assumes that you have working map files. AutoFS uses your existing Automount map files. By convention, Automount map files are located in the <code>/etc</code> directory with names that have the prefix <code>auto</code> . They indicate which remote file systems to mount, where to mount them, and which options to use. See the discussion of map files in <code>automount(8)</code> .

Requirement	Description
Impact on Availability	This Best Practice describes three methods of migrating from Automount to AutoFS: without rebooting any cluster member, rebooting only one member, and rebooting the entire cluster. System availability depends on the method you choose to use.
Skill Level Requirements	This Best Practice is intended for experienced system administrators. You must be familiar with Automount and AutoFS.

Before You Begin

Before you apply the Best Practice for migrating from Automount to AutoFS, use the CAA `caa_stat` command to verify whether the `autofs` CAA resource is registered. If it is not, register it.

```
# /usr/bin/caa_stat autofs
Could not find resource autofs.

# /usr/sbin/caa_register autofs
# /usr/bin/caa_stat autofs
NAME=autofs
TYPE=application
TARGET=OFFLINE
STATE=OFFLINE
```

Applying the Best Practice

Before you migrate from Automount to AutoFS, be sure to follow the recommendations in *Before You Begin*.

Migrating from Automount to AutoFS

This Best Practice describes how to migrate from Automount to AutoFS. Three possible migration scenarios are presented: migrating without rebooting any cluster member, migrating when you reboot the active Automount server node, and migrating when rebooting the entire cluster.

Migrating Without a Reboot

Migrating without rebooting any cluster member requires the largest number of procedural steps, but provides the highest availability. This procedure requires the most steps because you cannot rely on a reboot to clean up the Automount intercept points and to automatically start AutoFS.

Note

Most Automount environments have a single automount instance for all map files. This procedure describes this common case.

If you have a complex Automount environment with a separate automount instance for each map file, you might have a customized version of the `/etc/rc.config.common` file, or the `ps` command might return multiple process identifiers and you must kill them all, or one cluster member might not be the Automount server node for all NFS file systems, and so forth.

As you extrapolate the procedure to fit your Automount environment, kill the “standby” copies of the Automount process first to prevent the Automount service from failing over when you kill the active Automount server process.

Follow these steps to migrate from Automount to AutoFS without rebooting any cluster member:

1. Change the `rc.config.common` file.
 - a. Determine the arguments to pass to `autofsmount`. These arguments are typically a subset of those already specified by the `AUTOMOUNT_ARGS` environment variable. To view the value of that variable, use the `rcmgr -get` command, as shown in the following example:

```
# /usr/sbin/rcmgr -c get AUTOMOUNT_ARGS
-D MACH=alpha -D NET=f /- /etc/auto.direct
```

Environment variables set by using the `-D` option resolve placeholders in the definition of automount map file entries. For example, the associated `NET` entry might appear in the map file as follows:

```
vsx ${NET}system:/share/hunch/usr/projects2/vsx
```

and would resolve to

```
vsx fsystem:/share/hunch/usr/projects2/vsx
```

- b. Set the arguments to pass to `autofsmount`, as determined in the previous step. To do this, use the `rcmgr -set` command, as shown in the following example:

```
# /usr/sbin/rcmgr -c set AUTOFSMOUNT_ARGS -D MACH=alpha -D NET=f /- /etc/auto.direct
```

- c. Set the arguments to pass to the `autofs` daemon, as shown in the following example:

```
# /usr/sbin/rcmgr -c set AUTOFS_ARGS -D MACH=alpha -D NET=f
```

These arguments must match the environment variables, specified with the `-D` option, as set for `AUTOMOUNT_ARGS`.

- d. Use the `mount -e` command to identify a file system served by automount.

```
# mount -e | grep "(pid"
deli.zk3.dec.com:(pid524825) on /net type nfs (v2, ro, noatime, noac, timeo=350, retrans=5)
```

The automounted file system is indicated by `hostname: (pid)`.

- e. Determine which cluster member is the Automount server node for the NFS file system you identified in the previous step, as shown in the following example:

```
# cfsmgr -p /net
Domain or filesystem name = /net
Server Name = swiss
Server Status: OK
```

- f. Stop the Automount service on all cluster members other than the Automount server you identified in the previous step. To do this, use the `ps -ef` command to display process identifiers, search the output for instances of `automount`, and then use the `kill -TERM` command, where `TERM` is the default, to kill each process.

```
# ps -ef | grep automount
root 1049132 1048577 0.0 May 10 ?? 0:00.00 /usr/sbin/automount -D MACH=alpha -D NET=f /- /etc/auto.direct
```

```
# kill 1049132
```

Note that, starting with Tru64 UNIX Version 5.1A, the `kill` command is cluster aware; you can kill a process from any cluster member.

- g. Disable Automount and enable AutoFS in the `rc.config.common` file, as follows:

```
# /usr/sbin/rcmgr -c set AUTOMOUNT 0
# /usr/sbin/rcmgr -c set AUTOFS 1
```

2. Wait for all automounted file systems to become quiescent.
3. Stop the Automount service on the cluster member that is operating as the server. To do this, use the `ps -ef` command to display process identifiers, search the output for instances of `automount`, and then use the `kill -TERM` command, where `TERM` is the default, to kill each process. Sending the `SIGTERM` signal to the `automount` daemon causes it to unmount all file systems that it has mounted, and to exit.

```
# ps -ef | grep automount
root 524825 524289 0.0 May 10 ?? 0:00.01 /usr/sbin/automount -D MACH=alpha -D NET=f /- /etc/auto.direct
```

```
# kill 524825
```

4. Use the `mount -e` command and search the output for `tmp_mnt`, or the directory specified with the `automount -M` command, to verify that automounted file systems are no longer mounted. No file systems should be mounted on `tmp_mnt`.

```
# mount -e | grep tmp_mnt
```

If some mount points still exist, they will no longer be usable via the expected pathnames. However, they are still usable under the full `/tmp_mnt/...` pathnames. Because AutoFS does not use the `/tmp_mnt` mount point, there is no conflict and the full automount name space is available for AutoFS. If these `tmp_mnt` mount points later become idle, you can unmount them by using the `-f` option of the `umount` command, which unmounts remote file systems without notifying the server.

5. Start AutoFS. AutoFS provides automatic failover of the automounting service by means of CAA: one cluster member acts as the CFS server for automounted file systems, and runs the one active copy of the AutoFS daemon. If this cluster member fails, CAA starts the `autofs` resource on another member.

If you do not care which node serves AutoFS, use the `/usr/sbin/caa_start autofs` command without specifying a cluster member; otherwise, use the `/usr/sbin/caa_start autofs -c member-name` command to specify the cluster member that you want to serve AutoFS.

```
# /usr/sbin/caa_start autofs
```

The `-c` option starts the `autofs` resource on the specified member if the cluster member is allowed by the placement policy and resource dependencies. If the cluster member specified is not allowed by the placement policy and resource dependencies, the `caa_start` command fails. If the specified member is not available, the command fails.

See the discussion of the resource file options in `caa_profile(8)`.

6. Use the `caa_stat autofs` command to make sure that the `autofs` resource started as expected.

```
# /usr/bin/caa_stat autofs
NAME=autofs
TYPE=application
```

```
TARGET=ONLINE
STATE=ONLINE on swiss
```

Migrating When Rebooting a Cluster Member

Migrating when rebooting a cluster member requires fewer procedural steps than migrating without a reboot, at the expense of availability.

Note

Before you shut down a cluster member, you need to determine whether the cluster member you are shutting down is a critical voting member, and whether it is the only hosting member for one or more applications with a restricted placement policy. Both of these issues are described in the chapter about managing cluster members in the *Cluster Administration* manual.

Note

Most Automount environments have a single automount instance for all map files. This procedure describes this common case.

If you have a complex Automount environment with a separate automount instance for each map file, you might have a customized version of the `/etc/rc.config.common` file, or the `ps` command might return multiple process identifiers and you must kill them all, or one cluster member might not be the Automount server node for all NFS file systems, and so forth.

As you extrapolate the procedure to fit your Automount environment, kill the “standby” copies of the Automount process first to prevent the Automount service from failing over when you kill the active Automount server process.

Follow these steps to migrate from Automount to AutoFS when rebooting a cluster member:

1. Change the `rc.config.common` file.
 - a. Determine the arguments to pass to `autofs mount`. These arguments are typically a subset of those already specified by the `AUTOMOUNT_ARGS` environment variable. To view the value of that variable, use the `rcmgr -get` command, as shown in the following example:

```
# /usr/sbin/rcmgr -c get AUTOMOUNT_ARGS
-m -D MACH=alpha -D NET=f /- /etc/auto.direct
```

Environment variables set by using the `-D` option resolve placeholders in the definition of automount map file entries. For example, the associated `NET` entry might appear in the map file as follows:

```
vsx ${NET}system:/share/hunch/usr/projects2/vsx
```

and would resolve to

```
vsx fsystem:/share/hunch/usr/projects2/vsx
```

- b. Set the arguments to pass to `autofs` mount, as determined in the previous step. To do this, use the `rcmgr -set` command, as shown in the following example:

```
# /usr/sbin/rcmgr -c set AUTOFSMOUNT_ARGS -D MACH=alpha -D NET=f /- /etc/auto.direct
```

- c. Set the arguments to pass to the `autofs` daemon, as shown in the following example:

```
# /usr/sbin/rcmgr -c set AUTOFS_ARGS -D MACH=alpha -D NET=f
```

These arguments must match the environment variables, specified with the `-D` option, as set for `AUTOMOUNT_ARGS`.

- d. Use the `mount -e` command to identify a file system served by Automount:

```
# mount -e | grep "(pid"
deli.zk3.dec.com:(pid524825) on /net type nfs (v2, ro, noprpid, udp, hard, intr, noac, timeo=350, retrans=5)
```

The automounted file system is indicated by `hostname:(pid)`.

- e. Determine which cluster member is the Automount server node for the NFS file system you identified in the previous step.

```
# cfsmgr -p /net
Domain or filesystem name = /net
Server Name = swiss
Server Status: OK
```

- f. Stop the Automount service on all cluster members other than the Automount server you identified in the previous step. To do this, use the `ps -ef` command to display process identifiers, search the output for instances of `automount`, and then use the `kill -TERM` command, where `TERM` is the default, to kill each process.

```
# ps -ef | grep automount
root 1049132 1048577 0.0 May 10 ?? 0:00.00 /usr/sbin/automount -D MACH=alpha -D NET=f /- /etc/auto.direct
```

```
# kill 1049132
```

Note that, starting with Tru64 UNIX Version 5.1A, the `kill` command is cluster aware; you can kill a process from any cluster member.

- g. Disable Automount and enable AutoFS in the `rc.config.common` file, as follows:

```
# /usr/sbin/rcmgr -c set AUTOMOUNT 0
# /usr/sbin/rcmgr -c set AUTOFS 1
```

2. Optionally, specify the AutoFS server. AutoFS provides automatic failover of the automounting service by means of CAA: one cluster member acts as the CFS server for automounted file systems, and runs the one active copy of the AutoFS daemon. If this cluster member fails, CAA starts the `autofs` resource on another member.

You can use the `caa_profile autofs -print` command to view the CAA hosting and placement policy, if any. The hosting policy specifies an ordered list of members, separated by white space, that can host the application resource. The placement policy specifies the policy according to which CAA selects the member on which to start or restart the application resource. The autostart policy determines whether you want to start the resource automatically, regardless of whether it had been stopped or running before a reboot. Set `auto_start=1` if you want to start the resource regardless of whether it had been running before the reboot.

```
# /usr/sbin/caa_profile autofs -print
NAME=autofs
TYPE=application
ACTION_SCRIPT=autofs.scr
ACTIVE_PLACEMENT=0
AUTO_START=0
CHECK_INTERVAL=0
DESCRIPTION=Autofs Services
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
OPTIONAL_RESOURCES=
PLACEMENT=balanced
REQUIRED_RESOURCES=
RESTART_ATTEMPTS=3
SCRIPT_TIMEOUT=3600
```

The default, and recommended, behavior is to run on any cluster member, with a placement policy of `balanced`. If this is not suitable for your environment, use the `/usr/sbin/caa_profile -update` command to change the `autofs` resource profile.

See the discussion of the resource file options in `caa_profile(8)`

If you make a change, use the CAA `/usr/sbin/caa_register -u autofS` command to have the update take effect.

3. Reboot the cluster member. Before you shut down the cluster member, make sure that it is not a critical voting member or the only hosting member for one or more applications with a restricted placement policy. Both of these issues are described in the chapter about managing cluster members in the *Cluster Administration* manual.

When it reboots, Automount will no longer be running in the cluster, and AutoFS will start.

```
# /sbin/shutdown -r now
```

Migrating When Rebooting the Cluster

Migrating when rebooting the entire cluster requires fewer procedural steps than migrating without a reboot or migrating when rebooting a single member, at the expense of cluster availability.

Rebooting the cluster is a drastic measure; this is not the preferred migration method.

Follow these steps to migrate from Automount to AutoFS when rebooting the cluster:

1. Change the `rc.config.common` file.
 - a. Determine the arguments to pass to `autofsmount`. These arguments are typically a subset of those already specified by the `AUTOMOUNT_ARGS` environment variable. To view the value of that variable, use the `rcmgr -get` command, as shown in the following example:

```
# /usr/sbin/rcmgr -c get AUTOMOUNT_ARGS  
-D MACH=alpha -D NET=f /- /etc/auto.direct
```

Environment variables set by using the `-D` option resolve placeholders in the definition of automount map file entries. For example, the associated `NET` entry might appear in the map file as follows:

```
vsx ${NET}system:/share/hunch/usr/projects2/vsx
```

and would resolve to

```
vsx fsystem:/share/hunch/usr/projects2/vsx
```

- b. Set the arguments to pass to `autofs`, as determined in the previous step. To do this, use the `rcmgr -set` command, as shown in the following example:

```
# /usr/sbin/rcmgr -c set AUTOFSMOUNT_ARGS -D MACH=alpha -D NET=f /- /etc/auto.direct
```

- c. Set the arguments to pass to the `autofs` daemon, as shown in the following example:

```
# /usr/sbin/rcmgr -c set AUTOFSD_ARGS -D MACH=alpha -D NET=f
```

These arguments must match the environment variables, specified with the `-D` option, as set for `AUTOMOUNT_ARGS`.

- d. Disable Automount and enable AutoFS in the `rc.config.common` file, as follows:

```
# /usr/sbin/rcmgr -c set AUTOMOUNT 0
# /usr/sbin/rcmgr -c set AUTOFS 1
```

2. Optionally, specify the AutoFS server. AutoFS provides automatic failover of the automounting service by means of CAA: one cluster member acts as the CFS server for automounted file systems, and runs the one active copy of the AutoFS daemon. If this cluster member fails, CAA starts the `autofs` resource on another member.

You can use the `/usr/bin/caa_profile autofs -print` command to view the CAA hosting and placement policy, if any. The hosting policy specifies an ordered list of members, separated by white space, that can host the application resource. The placement policy specifies the policy according to which CAA selects the member on which to start or restart the application resource. The autostart policy determines whether you want to start the resource automatically, regardless of whether it had been stopped or running before a reboot. Set `auto_start=1` if you want to start the resource regardless of whether it had been running before the reboot.

```
# /usr/bin/caa_profile autofs -print
NAME=autofs
TYPE=application
ACTION_SCRIPT=autofs.scr
ACTIVE_PLACEMENT=0
AUTO_START=0
CHECK_INTERVAL=0
DESCRIPTION=Autofs Services
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
OPTIONAL_RESOURCES=
PLACEMENT=balanced
REQUIRED_RESOURCES=
RESTART_ATTEMPTS=3
```

```
SCRIPT_TIMEOUT=3600
```

The default, and recommended, behavior is to run on any cluster member, with a placement policy of `balanced`. If this is not suitable for your environment, use the `/usr/bin/caa_profile -update` command to change the `autofs` resource profile.

See the discussion of the resource file options in `caa_profile(8)`

If you make a change, use the CAA `/usr/sbin/caa_register -u autofs` command to have the update take effect.

3. Reboot the cluster. When it reboots, Automount will no longer be running in the cluster, and AutoFS will start.

```
# /sbin/shutdown -c now
```

Verifying Success

After you apply the Best Practice for migrating from Automount to AutoFS, you can verify whether it was successful by making sure that the `autofs` resource is online:

```
# /usr/bin/caa_stat autofs
NAME=autofs
TYPE=application
TARGET=ONLINE
STATE=ONLINE on swiss
```

If the Best Practice was not successful, see *Troubleshooting* for information about identifying and solving problems.

Troubleshooting

If you determine that the Best Practice was not successful, as described in *Verifying Success*, ensure that the automounted file system is being served as expected. To do this, perform the following steps:

- If AutoFS seems completely unavailable:
 1. Verify the contents of the `/etc/rc.config.common` file to ensure that `AUTOFS_ARGS` and `AUTOFSMOUNT_ARGS` are set as described in this Best Practice and that `AUTOFS=1`.
 2. Find out the status of the `autofs` CAA resource with the `caa_stat` command. If CAA reports that it is online, proceed to the next step. If CAA reports that the `autofs` resource is offline, start the resource with the `caa_start autofs` command. If the status of the `autofs` does not change to online, add the verbose

(-v) argument to the set of arguments you pass to autofs mount and the autofs daemon. For example,

```
# /usr/sbin/rcmgr -c set AUTOFSMOUNT_ARGS -v -D MACH=alpha -D NET=f /- /etc/auto.direct
# /usr/sbin/rcmgr -c set AUTOFS_ARGS -v -D MACH=alpha -D NET=f
```

The -v argument must be in addition to any other arguments you pass. After you have done this, stop and restart the autofs CAA resource:

```
# caa_stop -f autofs
# caa_start autofs
```

Note any errors reported by CAA and examine the contents of the /var/adm/syslog.dated/[date]/daemon.log and /var/adm/syslog.dated/current/user.log log files for the cluster member on which CAA attempted to start the autofs resource.

3. Use the ps command on the member on which CAA indicates the autofs resource is online to verify that the autofs daemon is in fact running:

```
# ps agx | grep autofs
```

If autofs is running, proceed to the next step. Otherwise, if you have not already done so, add the verbose (-v) argument to the set of arguments you pass to the autofs daemon. For example,

```
# /usr/sbin/rcmgr -c set AUTOFS_ARGS -v -D MACH=alpha -D NET=f
```

The -v argument must be in addition to any other arguments you pass. After you have done this, stop and restart the autofs CAA resource:

```
# caa_stop -f autofs
# caa_start autofs
```

Note any errors reported by CAA and verify the contents of the /var/adm/syslog.dated/current/daemon.log log file for the cluster member on which autofs is running.

4. Verify that the intercept points exist as expected. To do this, use the mount -e command to identify AutoFS intercept points. Search the output for autofs references similar to the following:

```
# mount -e | grep autofs
/etc/auto.direct on /mnt/mytmp type autofs (rw, nogripid, direct)
```

If the intercept points are missing, run autofs mount from the shell to install the intercept points, specifying the same AUTOFSMOUNT_ARGS you set with rcmgr, plus the verbose -v argument. Look in /var/adm/syslog.dated/current/user.log for any messages.

5. Use the `cfsmgr -e` command to show the servers for all mount points. Search the output for map file entries similar to the following:

```
# cfsmgr -e
Domain or filesystem name = /etc/auto.direct
Mounted On = /mnt/mytmp
Server Name = swiss
Server Status: OK
```

The server should match that returned by the `caa_stat autofs` command:

```
# /usr/bin/caa_stat autofs
NAME=autofs
TYPE=application
TARGET=ONLINE
STATE=ONLINE on swiss
```

If the server does not match, stop and restart the `autofs CAA` resource:

```
# caa_stop -f autofs
# caa_start autofs
```

- If AutoFS seems to work, but there is a problem with a specific file system:
 1. After a failed mount attempt, examine the `/var/adm/syslog.dated/current/daemon.log` log file for the cluster member on which `autofs` is running.
 2. Verify the map entry associated with the file system, and that any `-D` parameters required are set in both `AUTOFSMOUNT_ARGS` and `AUTOFSMOUNT_ARGS`.

```
# /usr/sbin/rcmgr -c get AUTOFSMOUNT_ARGS
# /usr/sbin/rcmgr -c get AUTOFS_ARGS
```

Comments and Questions

We value your comments and questions on the information in this document. Please mail your comments to us at this address:

best_practices@zk3.dec.com

Legal Notice

Compaq, the Compaq logo, StorageWorks, and TruCluster Registered in U.S. Patent and Trademark Office. Tru64 is a trademark of Compaq Information Technologies Group, L.P. in the United States and other countries. Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States and other countries. UNIX is a trademark of The Open Group in the United States and other countries. All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.