

Novell Cluster Services

Troubleshooting Novell Cluster Services for NetWare 5 and NetWare 6

- [Server Abends](#)
- [Cluster Resource Problems](#)
- [Failure to Start Cluster Services](#)
- [Client Problems](#)
- [NSS Errors](#)
- [Trustee Problems](#)
- [Upgrade Problems](#)
- [Install Problems](#)
- [License Issues](#)
- [ConsoleOne Issues](#)
- [LAN and SAN Related Problems](#)

Server Abends

One of the most important problems that requires troubleshooting Novell® Cluster Services™ is that of intentional abends. Intentional abends is functionality that is included with Novell Cluster Services to cause node isolation. Before discussing how to troubleshoot intentional abends, a more detailed discussion will help you understand why it is included with Novell Cluster Services.

A split-brain condition is a condition in which a single node or a group of nodes becomes isolated from the other nodes in the cluster. Consider, for example, a case where three nodes in a cluster are connected to one switch and another six nodes in the same cluster are connected to a different switch. If the cross-connect between the two switches were to fail, a split-brain condition would then exist. The three nodes connected to one switch would believe that the other six nodes failed, and the six nodes would believe that the three-node group had failed.

If this condition were allowed to continue, the group of three nodes would start activating and mounting the resources that were currently running on the group of six nodes. Meanwhile, the group of six would start activating and mounting the resources that were currently running on the group of three.

Novell Cluster Services does include a distributed lock for the Split Brain Detection partition used by the clustering software. But it does not support a lock across cluster nodes for user data. So if two servers were allowed to write to the same volume at the same time, there would be no way to prevent file system corruption.

To prevent file system corruption, Novell Cluster Services uses the Cluster Services partition that is created on the shared storage system during the installation. Each node is assigned its own disk space on this partition, and each node performs a periodic write to its own area. In addition to writing to its own area, each node can also read the information on the disk space areas of all the other nodes. If a cluster node can no longer access the Cluster Services partition, it removes itself from the cluster.

In this three-node/six-node example, suppose both the three-node group and the six-node group still had access to the Cluster Services partition. In this case they could also see that while the other group is no longer communicating on the LAN, the group is still active in the cluster. This causes the split-brain algorithm to force a vote, in which case the group of six nodes wins and the group of three nodes loses. Each node on the losing side then "eats the poison pill," meaning it causes a self-inflicted abend to remove itself from the cluster and stop all processes on the server.

Why intentionally cause the servers to abend? Why not just have the servers leave the cluster or issue a DOWN command to bring them down? There are several reasons why it is preferable to cause the servers to abend. The two most important reasons are as follows:

- To allow client reconnections, the services must be migrated very quickly. If a cluster node has 10 services running on it, it might take several minutes to gracefully stop all these services. This would be far too slow for a clean client reconnection.
- If a node is being removed from the cluster, there must be some problem with the node. There would be no way to guarantee that the console hasn't already hung, or some processes are hung while others are not. If a partially functioning server were to fail to gracefully leave the cluster,

it might still have services writing to the shared storage, which could cause file system corruption.

So although an abend decreases the availability of a single node in the cluster, it actually improves overall availability of the actual services by quickly restarting these services on nodes that are not experiencing problems.

For more information on intentional abends and how to prevent them, see [LAN and SAN related problems](#).

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

Cluster Resource Problems

Resources in a Comatose State

The following screen is an example of what you will see in the ConsoleOne® Cluster State View when a cluster resource is in a comatose state.

The screenshot shows the Novell ConsoleOne interface. On the left is a tree view of the cluster hierarchy, including nodes like MOBE2_SYS, MOBE3, MOBE4, MOBE5, and the selected 'Mobe_Cluster'. The main pane displays the 'Cluster State View' for 'Mobe_Cluster' at 'Epoch 56'. It shows five server icons labeled MOBE1 through MOBE5. Below these are two progress bars: 'Nodes Available (%)' at 100% and 'Resources Available (%)' at approximately 35%. At the bottom is a table of cluster resources.

Cluster Resource	State	Locati...	Lives	Up Since
Idaptest	Offline		25	
Master_IP_Address_R...	Running	MOBE5	11	12/28/01 11:11:39 AM
TESTPOOL_SERVER	Comatose	MOBE1	26	1/3/02 2:57:52 PM

User: admin.MobeC1 Tree: MOBET1

A comatose state indicates that the resource is not running properly and requires administrator intervention.

A resource goes into the comatose state when it encounters an error during the load or unload script execution or when the scripts do not complete within the load or unload script time-out period.

Comatose resources are almost always caused by an error or typo in the load or unload script or by interference in the cluster as a result of an administrator's manual intervention.

The best way to determine the cause is to watch the server console screen while you online or offline the resource. Watch for any error messages or warnings on the screen. These will most likely reveal the cause of the error.

Since an error or typo in a load or unload is a common cause of comatose resources, you should first

check the scripts to ensure everything is correct, especially volume names.

Common Load Script Errors

The following screen is an example of what you will see on the server console screen when a volume name is incorrectly spelled in a load script.

```
TCPIP-6.4-186: Mon Jan 7 15:22:49 2002
Deleted secondary IP address 151.155.129.138.
Deactivating pool "TESTPOOL"...
Deactivating volume "SHAREVOL"...
Dismounting Volume SHAREVOL
    Pool TESTPOOL set to the DEACTIVE state.
Activating pool "TESTPOOL"...
    ** Pool layout v40.07
    ** Previous clean shutdown detected (consistency check OK)
    ** Loading system objects
    ** Processing volume purge log
    ** .
    ** Processing pool purge log
    ** .
Loading volume "SHAREVOL"
    Volume SHAREVOL set to the DEACTIVE state.
    Pool TESTPOOL set to the ACTIVE state.
CLUSTER- WARNING - 10264 : CRM: Error executing load script for TESTPOOL_SERVER
    failed on command line 2 "mount SHAREVOL2 VOLID=252"

TCPIP-6.4-211: Mon Jan 7 15:23:51 2002
Secondary IP address 151.155.129.138 is not associated with any local binding.
CLUSTER- INFO - 211 : WSASetService() failed, error = -1, 10109
Deactivating pool "TESTPOOL"...
    Pool TESTPOOL set to the DEACTIVE state.
```

The following screen is an example of what you will see on the server console screen when an IP address specified in the load script is already in use somewhere else.

```
TCPIP-6.4-185: Thu Jan 3 15:24:35 2002
Added secondary IP address 151.155.129.48.
Activating pool "SHAREPOOL"...
    ** Pool layout v40.07
    ** Processing journal
        ** 0 uncommitted transaction(s)
        ** 0 Redo(s), 0 Undo(s), 0 Logical Undo(s)
    ** System verification completed
    ** Loading system objects
    ** Processing volume purge log
    ** .
    ** Processing pool purge log
    ** .
    Pool SHAREPOOL set to the ACTIVE state.

TCPIP-6.4-178: Thu Jan 3 15:24:43 2002
Cannot add the secondary IP Address 151.155.129.48. IPAddress is in use.
CLUSTER- WARNING - 10264 : CRM: Error executing load script for SHAREPOOL_SERVER
```

```
failed on command line 4 "add secondary ipaddress 151.155.129.48"

TCPIP-6.4-186: Thu Jan  3 15:24:49 2002
Deleted secondary IP address 151.155.129.48.
Deactivating pool "SHAREPOOL"...
Pool SHAREPOOL set to the DEACTIVE state.
```

The following screen provides an example of what you will see on the server console screen when an IP address specified in the load script is not associated with a local binding. This means the secondary IP address can not be added because a primary IP address with the same mask does not exist.

```
** Pool layout v40.07
** Previous clean shutdown detected (consistency check OK)
** Loading system objects
** Processing volume purge log
** .
** Processing pool purge log
** .
Loading volume "TESTVOL"
Volume TESTVOL set to the DEACTIVE state.
Pool TESTPOOL set to the ACTIVE state.
Activating volume "TESTVOL"...
** Volume layout v35.00
** Volume creation layout v35.00
** Processing volume purge log
** .
Volume TESTVOL set to the ACTIVE state.
CLUSTER- WARNING - 10264 : CRM: Error executing load script for TESTPOOL_SERVER
failed on command line 5 "addsf secondary ipaddress 151.155.129.48"

TCPIP-6.4-211: Thu Jan  3 14:57:54 2002
Secondary IP address 151.155.129.48 is not associated with any local binding.
Deactivating pool "TESTPOOL"...
Deactivating volume "TESTVOL"...
Dismounting Volume TESTVOL
Pool TESTPOOL set to the DEACTIVE state.
```

The following screen is an example of what you will see on the server console screen when load script commands are in the wrong order. In this case, a dependent module or volume was not available and script commands could not be completed.

```
Volume TESTVOL set to the DEACTIVE state.
Pool TESTPOOL set to the ACTIVE state.
Activating volume "TESTVOL"...
** Volume layout v35.00
** Volume creation layout v35.00
** Processing volume purge log
** .
Volume TESTVOL set to the ACTIVE state.

TCPIP-6.4-185: Thu Jan  3 15:29:58 2002
Added secondary IP address 151.155.129.48.

TCPIP-6.4-186: Thu Jan  3 15:30:10 2002
```

```

Deleted secondary IP address 151.155.129.48.
Deactivating pool "TESTPOOL"...
Deactivating volume "TESTVOL"...
Dismounting Volume TESTVOL
    Pool TESTPOOL set to the DEACTIVE state.
CLUSTER- WARNING - 10264 : CRM: Error executing load script for TESTPOOL_SERVER
    failed on command line 1 "mount TESTVOL VOLID=253"

TCPIP-6.4-211: Thu Jan  3 15:30:47 2002
Secondary IP address 151.155.129.48 is not associated with any local binding.
CLUSTER- INFO - 211 : WSASetService() failed, error = -1, 10109
    Pool TESTPOOL is already in the DEACTIVE state.

```

Comatose resources caused by administrator intervention are most commonly due to an administrator manually executing portions of the resource load or unload script. You should use ConsoleOne, NetWare® Remote Manager, or the cluster command line interface (available with Novell Cluster Services™ 1.6 and later) to start or stop cluster resources. Treat the resource as a whole. Do not manually execute parts of resource load or unload scripts.

Resources in an NDS Sync State

The following screen is an example of what you will see in the ConsoleOne Cluster State View when a cluster resource is in an NDS® Sync state.

The screenshot shows the Novell ConsoleOne interface. The left pane displays a tree view of the cluster hierarchy, including MOBE2_SYS, MOBE3, MOBE4, MOBE5, and the Mobe_Cluster. The right pane shows the Cluster State View for Mobe_Cluster, Epoch 56. It displays five nodes (MOBE1 to MOBE5) with their respective resource availability percentages. Below the node icons are two sliders: 'Nodes Available (%)' and 'Resources Available (%)'. At the bottom, a table lists the cluster resources and their states.

Cluster Resource	State	Locati...	Lives	Up Since
20-Resource	NDS Sync		0	11/28/03 10:58:59 PM
Idaptest	Offline		29	
Master_IP_Address...	Running	MOBE5	11	12/28/01 11:11:39 AM
TESTPOOL_SERVER	Running	MOBE1	36	1/3/02 3:33:04 PM

User: admin.MobeC1 Tree: MOBE1

The most common cause of a resource going into and remaining in an NDS Sync state is when a number is used as the first character in a resource name. This problem does not exist in Novell Cluster Services version 1.01 Support Pack 2 and later.

When a number is used as the first character in a resource name, one of two things will occur:

- The resource will never leave the NDS Sync resource state
- The resource will go comatose and never actually start

When the resource never leaves the NDS Sync state, you will see continuous error messages similar to those shown in the following server console screen.

[illegible]

As an interesting test, create a cluster resource with the same name as another cluster resource, but add a number to the beginning of the cluster resource name. For example, you can create a cluster resource named DNS, and then create a second cluster resource named 2DNS. Both resources will go into a comatose state when they start and you won't be able to bring the DNS resource online or offline until you delete the 2DNS Resource object.

There isn't much troubleshooting to do here; just remember not to use numbers as the first character in cluster resources names.

Failure to Start Cluster Services

There are a number of reasons that Novell® Cluster Services™ might fail to start. The most common reasons are described below.

NDS and Time Synchronization Problems

The following screen is an example of what you see on the server console when there are NDS® problems or time synchronization is not functioning properly.

```
IBM_N1:
IBM_N1:version
Novell NetWare 6, SP1.C19
Support Pack Revision 01
(C) Copyright 1983-2001 Novell Inc. All Rights Reserved. Patent Pending.
Server Version 5.60.01 December 18, 2001
Novell eDirectory Version 8.6.2 SMP
NDS Version 10310.02 December 14, 2001
IBM_N1:
IBM_N1:ldnccs
Auto Restart After Abend is ALREADY set to 0
Loading Module CLSTRLIB.NLM [ ]

Clustering failed to load its configuration.
Please wait while NDS synchronizes.

Type '1' to quit waiting.
```

Wait for a few minutes to see if the problem is resolved. If, after the wait, CLSTRLIB.NLM still does not load, check NDS health and time synchronization.

Follow the instructions in TID 10060600 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/10060600.htm>) to perform an NDS health check, and verify that time is synchronized.

To force time synchronization, complete the following steps:

1. Load MONITOR on the server console.
2. Go to Server Parameters>Time>TIMESYNC Time Source and then enter the IP address of first server in your NDS tree. Usually the first server is configured as a Single time server and it has the Master NDS replica.
3. Go to Server Parameters>Time>TIMESYNC Reset> and set the value to ON.
4. Go to Server Parameters>Time>TIMESYNC Restart Flag>, and then set the value to ON.

LAN Communication Problems

The following screen is an example of what you see on the server console when there are LAN communications problems.

```
CMA.NLM will run on processor 0 only
Uni-Processor NLM
Module CMA.NLM load status OK
Loading module CMON.NLM
Cluster Membership Monitor Build Number = 12122001
Version 1.60a   December 12, 2001
Copyright (C) 1999-2001 Novell, Inc. All Rights Reserved.
CMON.NLM will run on processor 0 only
Uni-Processor NLM
Module CMON.NLM load status OK
Loading module NCSSDK.NLM
Cluster API Build Number = Internal Build
Version 1.60a   December 13, 2001
Copyright (C) 1999-2001 Novell, Inc. All Rights Reserved.
NCSSDK.NLM will run on processor 0 only
Uni-Processor NLM
Module NCSSDK.NLM load status OK
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
CLUSTER- INFO - 2090 : Join retry, some other node acquired the cluster lock
```

You might see messages similar to those above when the entire cluster comes up. This is normal. The nodes are competing to see which will become the master node.

If you see a lot of these messages when the cluster has already formed and new nodes are joining the cluster, there could be a LAN communication problem. In this case check all LAN hardware and drivers. See [LAN/SAN Problems](#) for more information.

Failure to Find SBD Partition

The following screen is an example of what you see on the server console when a node which is trying to join the cluster cannot find the SBD Partiton.

```
MOBE1:ldnec
Auto Restart After Abend is ALREADY set to 0
Loading Module CLSTRLIB.NLM           [      OK      ]
Loading Module SBDLIB.NLM             [      OK      ]
Loading Module VLL.NLM                [      OK      ]
Loading Module GIPC.NLM               [      OK      ]
Loading Module SBD.NLM                [      OK      ]
CLUSTER- INFO - 6135 : Searching for SBD partition ...
CLUSTER- INFO - 6135 : Searching for SBD partition ...
CLUSTER- INFO - 6135 : Searching for SBD partition ...
```

```

CLUSTER- INFO - 6135 : Searching for SBD partition ...
CLUSTER- FATAL - 6022 : There is NO SBD Partition required by the cluster !!!
Please run " SBD install " to create SBD Partition.
Loading Module VIPX.NLM [ OK ]
Loading Module CSS.NLM [ OK ]
Loading Module CVB.NLM [ OK ]
Loading Module CRM.NLM [ OK ]
Loading Module RWAIT.NLM [ OK ]
Loading Module PCLUSTER.NLM [ OK ]
Loading Module CMA.NLM [ OK ]
Loading Module CMON.NLM [ OK ]
Loading Module NCSSDK.NLM [ OK ]
CLUSTER- INFO - 12033 : Joining...
CLUSTER- FATAL - 12038 : SBD has reported a fatal error
MOBE1:

```

If there are other nodes in the cluster already running, it means this particular node is disconnected from the SAN or the part of the SAN contains the SBD partition.

Check all LAN hardware and drivers. See [LAN/SAN Problems](#) for more information.

Missing NDS Properties

The following screen is an example of what you see on the server console when there are missing NDS properties.

```

Copyright (C) 1999-2001 Novell, Inc. All Rights Reserved.
CLUSTER- INFO - 29 : ncslibResolveName: Now connected to .NODEA.novell.CIFSTREE.

CLUSTER- INFO - 29 : ncslibResolveName: Now connected to .CN=NODEA.O=novell.T=CI
FSTREE.
CLUSTER- FATAL - 65 : ncslibLoadClusterConfig: API called = DDCReadTOCB, error =
FFFFFDA5
ncslibLoadCluserConfig: could not initialize the cluster configuration
SERVER-5.00-1553: Module initialization failed.
Module CLSTRLIB.NLM NOT loaded
Loading module CLSTRLIB.NLM
Novell Cluster Configuration Library Build Number = 06082001
Version 1.01b June 8, 2001
Copyright (C) 1999-2001 Novell, Inc. All Rights Reserved.
CLUSTER- INFO - 29 : ncslibResolveName: Now connected to .NODEA.novell.CIFSTREE.

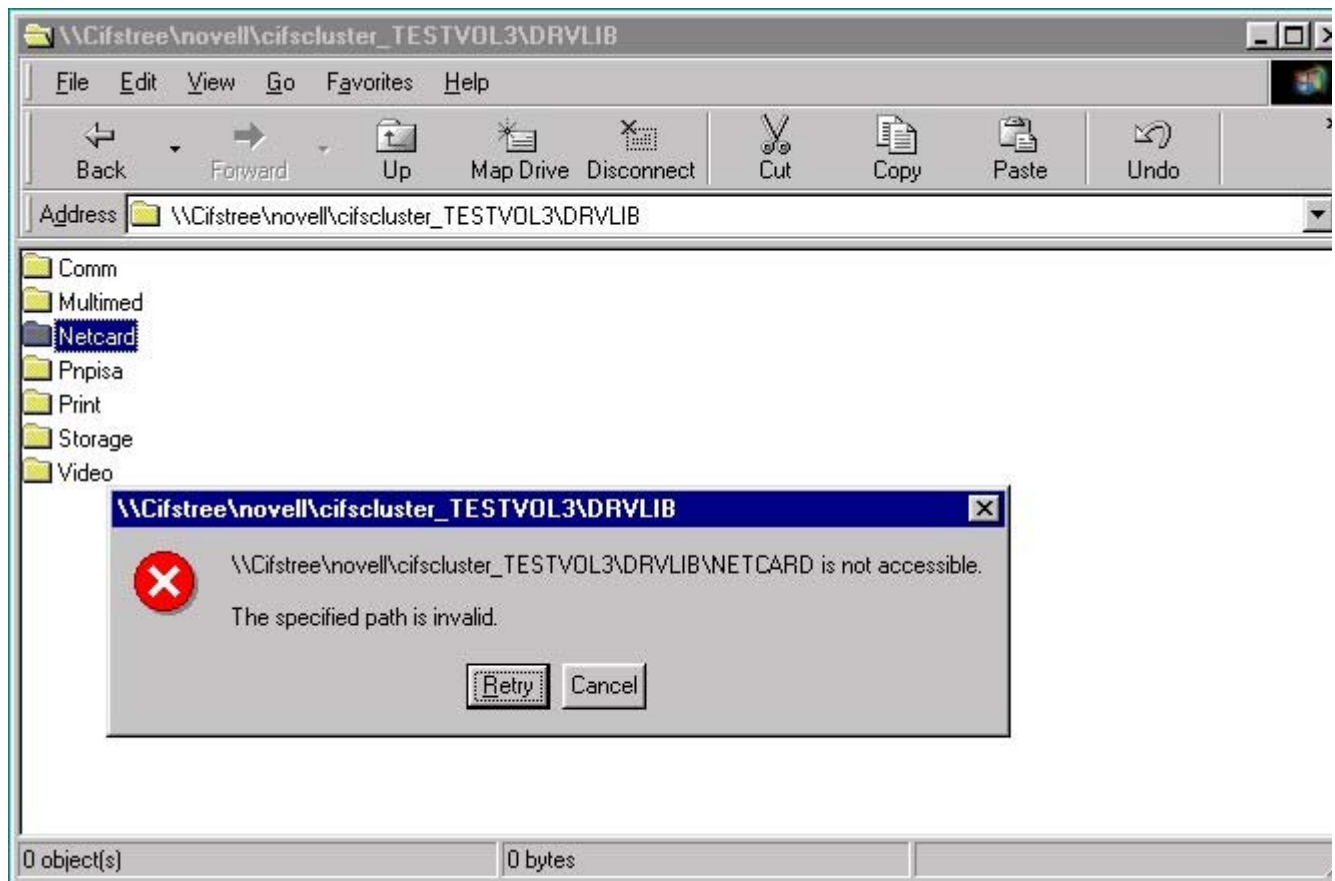
CLUSTER- INFO - 29 : ncslibResolveName: Now connected to .CN=NODEA.O=novell.T=CI
FSTREE.
CLUSTER- FATAL - 65 : ncslibLoadClusterConfig: API called = DDCReadTOCB, error =
FFFFFDA5
ncslibLoadCluserConfig: could not initialize the cluster configuration
SERVER-5.00-1553: Module initialization failed.
Module CLSTRLIB.NLM NOT loaded
??? Unknown command ???
NODEA:

```

One of the cluster nodes has lost its association with the cluster container.

Client Problems

After a cluster-enabled volume has failed over, failed back, or been migrated to another server in the cluster, you might occasionally see a message similar to the one in the following example.



Novell® Cluster Services™ supports transparent client reconnect to cluster enabled-volumes.

For the NetWare® Client 32™ for Windows* 95, Windows 98, and Windows Me, reconnect is supported at the file level. This means that applications and files will still be open after the client automatically reconnects.

For the NetWare Client for Windows NT* and Windows 2000, reconnect is supported at the drive mapping level only. This means that drive mappings will remain intact, but applications will have to be restarted and files will have to be reopened after the client automatically reconnects.

If the message in the example above appears, you might have to click the Retry button several times in order to reconnect to the cluster-enabled volume and restore your drive mappings.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

NSS Errors

The following screen is an example of what you see on the server console when there is a corrupted NSS block in the cluster.

```
Operational Mode 0
    Pool TSTPL000084 set to the DEACTIVE state.
Loading pool "TSTPL0086"
Operational Mode 0
    Pool TSTPL0086 set to the DEACTIVE state.
Loading pool "TSTPL0087"
Operational Mode 0

Jan 7, 2002  11:12:18 am  NSS ZLSS -3.01-1486: zlssLogicalVolume.c[4367]
    Error validating Pool Data Block 130949, status=20012.
Run Verify and if necessary, run Rebuild.

    Pool TSTPL0087 set to the DEACTIVE state.
Loading pool "TSTPL0088"
Operational Mode 0
    Pool TSTPL0088 set to the DEACTIVE state.
Loading pool "TSTPL0089"
Operational Mode 0
    Pool TSTPL0089 set to the DEACTIVE state.
Loading pool "TSTPL00000090"
Operational Mode 0
    Pool TSTPL000000C90 set to the DEACTIVE state.
Activating pool "TSTPL0000084"...
** Pool layout v4C.07
```

At the server console, run **NSS /PoolVerify = <poolname>** .

If necessary run **NSS /PoolRebuild = <poolname>** at the server console.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

Trustees Problems

Each server tracks file system trustees based on the Object ID of the trustee. With NetWare® 5, each server has its own view of Object IDs; any given NDS® object will have a different Object ID depending on which server is viewing the object. Because Object IDs are server centric, when you move a volume from one server to another, Novell® Cluster Services™ must adjust the Object IDs of each trustee based on the new server's perspective.

The TRUSTMIG.NLM module is used to adjust object IDs.

Note: NetWare 6 uses globally unique Object IDs, so TRUSTMIG.NLM is not required with NetWare 6.

TRUSTMIG.NLM normally functions well in accounting for and updating file system trustees. However, there might be times when something happens that causes trustee errors. Some of the more common problems include the following:

- An administrator manually moves a volume to another server without using the TRUSTMIG utility.
- A server fails in the middle of the trustee migration process.
- A trustee assignment is changed when TRUSTMIG is not monitoring the volume.
- An NDS object that has file system trustee assignments is deleted or moved when TRUSTMIG is not monitoring a volume but the volume is mounted. This could be caused by the volume resource being offline.
- NSS corruption occurs on the volume.
- An administrator has removed the TRUSTMIG command or the WATCH parameter from the load script.

Understanding the Function of TRUSTMIG

To help troubleshoot trustee issues, it is important to first understand the function of TRUSTMIG.NLM in a cluster. NetWare stores file system trustees in the metadata of the NSS volumes as a 4-byte Trustee ID (TID). These IDs are then resolved back to NDS objects based on the Entry ID table stored in the SYS:_NETWARE directory of the server. Each server has its own Entry ID table, which is different for each server.

When a volume moves from one server to another, the TIDs in the metadata are not valid for the new server. TRUSTMIG resolves this issue by keeping a TRUSTMIG.FIL file in a hidden _NETWARE directory on each shared volume that it is configured to monitor. The header record in TRUSTMIG.FIL has a marker that tells it which cluster node the file is currently valid on. It also contains a state marker, which tells the server the current migration status. This is used to recover from a failed migration. Finally, the TRUSTMIG.FIL file contains a list of all the Distinguished Names (DNs) of trustees for the volume, along with each trustee's assignments.

When a volume is mounted on a new server, TRUSTMIG goes through a routine to translate all of the DN's to the new server's TIDs. This process first produces temporary TIDs and then produces the actual TIDs, thus giving it the ability to recover even if the server were to crash in the middle of a migration.

Recovering from a Partially Migrated Volume

Any time a volume crashes in the middle of a trustee migration, Novell Cluster Services forces you to mount that volume back on the previous host server where the trustees were valid. This is necessary only when a node fails in the middle of the trustee migration process, which can happen only when a volume load script is being run on a node and that node fails. An example of this is when a volume fails over from one node to another and then the second node also fails. In this example, you have to repair the trustee assignments by mounting the volume on either the first or second node; do not mount the volume on any other node.

This problem happens only with a double-node failure. This means that one node fails, the second node runs TRUSTMIG, and then it also fails. The volume trustee assignments have to be restored on either the first or second node. If you attempt to mount the volume on a node other than the first or second, you will see an error similar to the following on the console of the node you are attempting to mount the volume on:

CLUSTER-<FATAL>-<18008>: You can't go to another node. Go back to either NODE1 or NODE2

It is extremely rare to get a double-node failure and rarer still that the second failure happens during the TRUSTMIG process. Novell Cluster Services is designed to deal with this kind of double-node failure. Simply move the volume back to the first node and then perform the following steps:

1. Bring the cluster volume resource offline.
2. Run the TRUSTOOL utility to finish the partial migration back to the previous host server.

The TRUSTOOL utility allows the migration to finish if you run it on the server where the volume was being moved to. If you run TRUSTOOL on the server that hosted the volume prior to the failure, it will back out of the migration. If it is run on a server other than the two described above, TRUSTOOL informs the administrator which server it must be run from.

To run the TRUSTOOL utility, execute the following command:

```
TRUSTOOL <VOLUME> FIX
```

Note: TRUSTOOL FIX deactivates the volume if it isn't already deactivated. Make sure the volume resource is offline before performing this action.

3. Activate and mount the volume.
4. Create a Trustool dump file by executing the following command:

```
TRUSTOOL <VOLUME> DUMP
```

This command writes all DNs and their migration status to the server console screen, as well as to the file SYS:ETC\TRUSTDMP.TXT.

5. Check the server console screen or the text file to verify that the trustees are correct.

You can also use NetWare Administrator, ConsoleOne[®], or Windows* Explorer to validate trustees. To do this you must manually access the volume through the server that has it mounted because you haven't yet brought the cluster-enabled volume online.

6. If necessary, add or edit any missing trustees or trustee assignments.
7. Purge any bad trustees.

The purging process deletes any bad DN's in the TRUSTMIG.FIL file. First it creates the TRUSTMIG.BAK file and then it re-creates the TRUSTMIG.FIL after purging the bad DN's.

Bad DN's occur whenever you delete an NDS object without first removing the object's file system trustee assignments. Bad DN's also occur when NDS doesn't synchronize quickly enough after new users with trustee assignments are created in the tree.

To purge bad trustees, execute the following command:

```
TRUSTOOL <VOLUME> PURGE
```

8. Create a new Trustool dump file by executing the following command:

```
TRUSTOOL <VOLUME> DUMP
```

9. Check the server console screen or the text file to verify that the trustees are correct.
10. Dismount and deactivate the volume.
11. Bring the volume online by using ConsoleOne to start the cluster volume resource.
12. Verify once again that the trustees are correct.
13. Migrate the volume to the server that you want it to run on.
14. Run TRUSTOOL DUMP on the new server and compare the dump file to the file you created in Step 8.

Preventing a Catastrophic Trustee Loss

Although the TRUSTMIG and TRUSTOOL utilities provide a great deal of protection and recovery in the event of trustee loss, you should still back up your trustees periodically just in case you have a serious volume error that prevents TRUSTOOL from recovering your trustees. Because the utilities described above rely on the TRUSTMIG.FIL file, any serious error that corrupts this file can cause an irreversible loss of trustees.

Novell has an unsupported utility called TBACKUP. This utility can be found at <http://support.novell.com/misc/patlst.htm#tools>. You can use TBACKUP to periodically back up your trustees. Many third-party companies also have tools that back up trustees. One of these is FSTRUST by DreamLAN Network Consulting Ltd. (<http://www.dreamlan.com/main.htm>).

Back up your trustees using your preferred method at intervals based on the frequency of file system trustees changes. If you have a lot of file system trustee activity, consider backing up your trustees weekly or even daily. For less active sites, consider doing it monthly.

A trademark symbol (®, TM, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

Upgrade Problems

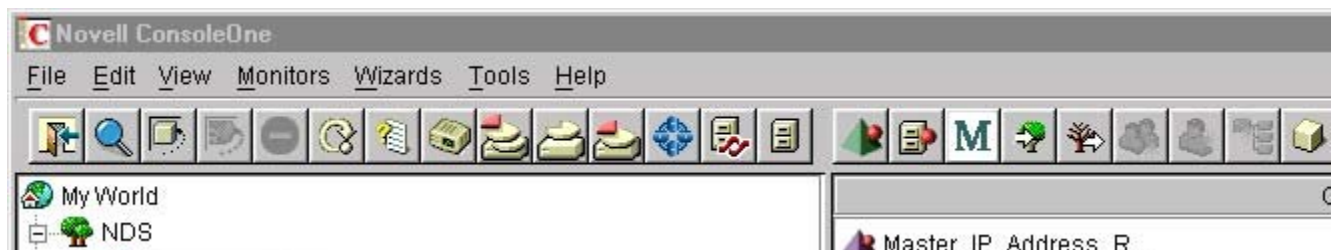
The following screen is an example of what you see during an upgrade on the server console when a cluster-enabled volume is offline or a cluster node just failed and the volume is in the process of failing over to another node in the cluster.

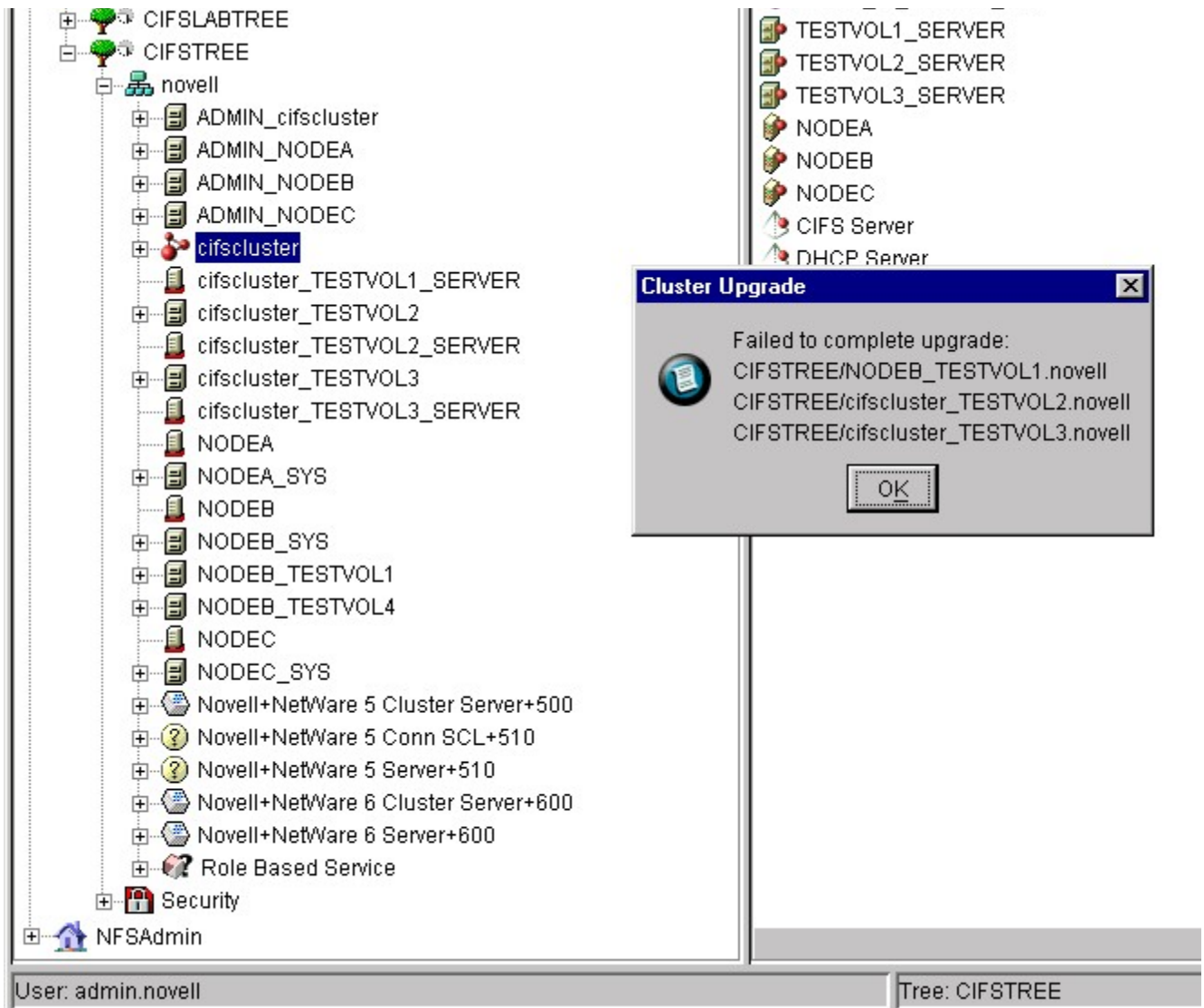


If the cluster-enabled volume is offline, bring the volume online and retry the upgrade process.

If the cluster node just failed and the volume is in the process failing over to another node in the cluster, wait for the failover to complete and retry the upgrade process.

The following screen is an example of what you see on the server console when the Pool objects for the volumes listed in the error message have not yet been added to NDS®.





Pool objects not being added to NDS are generally related to three causes:

- NDS is slow to add NSS Pool objects.

If you suspect this is the cause, wait and retry the upgrade process.

- Volumes are corrupted and, because of this, NSS Pools cannot be added to NDS.

If you suspect this is the cause, run NSS /PoolRebuild on the volume or volumes that failed. This command rebuilds the volume and creates the pool associated with the volume.

After running NSS /PoolRebuild, add the newly created pool to NDS via the Media tab in ConsoleOne®.

- NDS is unhealthy.

If you suspect this is the cause, go to [NDS and Time Synchronization Problems](#) to check NDS

health and time synchronization.

A trademark symbol ([®], [™], etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

Install Problems

If you have problems connecting to servers during the Novell® Cluster Services™ installation, map a drive to the servers from a Novell Client™ machine prior to installation.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

License Issues

The following screen is an example of what you see on the server console when the Cluster Server License (CSL) is not installed or is installed in the wrong location in your NDS[®] tree. You might also see this screen if NDS has health problems.

```
Deactivating volume "TESTVOLUME"...
Dismounting Volume TESTVOLUME
    Pool SHAREPOOL set to the DEACTIVE state.
CLUSTER- FATAL - 10 : Failed in RequestClusterServerLicense.
NCS PM: Leaving the cluster, status = FFFFFFFF.
CLUSTER- FATAL - 92 : Server CPQ3 was unable to obtain a valid Cluster Server Li
cense. Please contact your Network Administrator. Error # C0001002.

CLUSTER- WARNING - 17 : NCS PM can not obtain cluster server license. Another at
tempt will be made later.
CLUSTER- FATAL - 92 : Server CPQ3 was unable to obtain a valid Cluster Server Li
cense. Please contact your Network Administrator. Error # C0001002.

CLUSTER- WARNING - 17 : NCS PM can not obtain cluster server license. Another at
tempt will be made later.
CLUSTER- FATAL - 92 : Server CPQ3 was unable to obtain a valid Cluster Server Li
cense. Please contact your Network Administrator. Error # C0001002.

CLUSTER- WARNING - 17 : NCS PM can not obtain cluster server license. Another at
tempt will be made later.
CLUSTER- FATAL - 92 : Server CPQ3 was unable to obtain a valid Cluster Server Li
cense. Please contact your Network Administrator. Error # C0001002.

CLUSTER- FATAL - 10 : Failed in RequestClusterServerLicense.
NCS PM: Leaving the cluster, status = FFFFFFFF.
```

Install the CSL in the same context or container where the Cluster object is located.

If you suspect NDS health problems, see [NDS and Time Synchronization Problems](#).

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

ConsoleOne Issues

If the server where your eDirectory™ Master replica is located goes down, you might have to restart your NetWare® client machine. This applies only if you are accessing ConsoleOne from the server to run it on the client and is necessary for ConsoleOne to function properly on a client.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).

LAN and SAN Related Problems

LAN-Related Problems

The following list identifies the most common things to look for when troubleshooting LAN-related problems on a cluster.

- LAN switch delays or drops of multicast/broadcasts packets
- Extremely high LAN utilization caused by a broadcast or multicast storm
- Unstable LAN drivers
- Non optimal placement of LAN cards on the PCI bus
- A misbehaving NIC that transmits but won't receive packets

SAN-Related Problems

The following list identifies the most common things to look for when troubleshooting SAN-related problems on a cluster.

- Gigabit Interface Converters (GBICs) that are improperly seated or not fully connected to the switch or fiber card
- Error lights on the switch or fiber card
- Damaged fiber cable.

Try replacing a damaged cable with a new one.

- A bad GBIC.

Try switching the GBIC to a different node if a spare GBIC is not available.

- A bad fiber switch port.

Try switching to a different port.

- Laser light components not working properly.

Try cleaning the laser light components.

- Version problems.

Verify that the SAN devices and drivers are certified for the version of NetWare[®] you are running.

Your hardware vendor can provide vendor-specific SAN diagnostics based on the model of hardware you are using.

Intentional Abends

The following links provide detailed information on how to avoid intentional abends.

- [Understanding the Heartbeat Process](#)
- [LAN Failure or Congestion Issues](#)
- [LAN Card Issues](#)
- [Tuning Server Configuration to Avoid False Split Brain Conditions](#)
- [Categorizing the Four Types of Split Brain Conditions](#)
- [Categorizing Various Split Brain / Poison Pill ABENDs](#)

Understanding the Heartbeat Process

The first node that joins the cluster automatically becomes the master node in the cluster. Subsequent nodes that join the cluster become slave nodes. The slave nodes periodically perform a heartbeat based on the heartbeat parameter configured for the cluster by sending a unicast TCP/IP packet to the master node. The master node periodically performs a heartbeat across the LAN based on the master watchdog parameter by sending a multicast packet to all slave nodes.

In addition, each node performs a heartbeat across the SAN by periodically (based on one half of the tolerance parameter) increasing a counter value stored in its own section of the cluster services partition. Each node writes in its own space and reads all other nodes' sectors prior to writing its own update.

Note: By default, the master and slave heartbeats take place every second, with a tolerance of eight seconds.

A failure detection algorithm is initiated any time a node experiences a continuous failure to a heartbeat equal to the tolerance parameter. The master and slave nodes then communicate over the LAN to form a new cluster based on the nodes chosen by the failure detection algorithm. A new master node is also elected during this phase. The new master node could be the same node as before the failure. Because only one node is allowed to control any given cluster resource at a time, any node that fails to perform the heartbeat is removed from the cluster.

LAN Failure or Congestion Issues

A frequently asked question is why clusters do not have dedicated TCP/IP networks for heartbeat traffic. Because heartbeat traffic is very light, the heartbeat process itself does not necessitate a dedicated network. Additional reasons you would not want a dedicated network for heartbeat traffic include the following:

- Production LAN card failures will not be noticed if you are using a heartbeat network, so nodes can't be removed when they should.
- A heartbeat LAN failure with the production LAN still functioning will cause a poison pill condition, even though the node was actively servicing clients.
- The workstation running ConsoleOne[®] to manage the cluster would need a route to the heartbeat network to manage the cluster.

Enabling IP Packet Forwarding on the cluster nodes negatively affects client reconnection. This poses a trade-off between managing and dedicating the cluster traffic (unless you have another method of routing to the heartbeat network other than a cluster node).

Because of these and possibly other reasons, many clients decide not to implement a heartbeat network. You need to understand your production network environment and the heartbeat tuning

parameters before deciding whether or not to implement a heartbeat network.

Based on the defaults, if eight seconds of consecutive heartbeat packets are dropped, a node is removed from the cluster. If your network is congested or your VLAN includes multiple switches between the cluster nodes, you might have cases where nodes are removed from the cluster simply because the heartbeats didn't get through your network. If your network is too congested to support the default settings, you can increase the heartbeat tolerance. If you increase heartbeat tolerance, be aware that this might adversely affect transparent client reconnections.

As a general rule, if a client can't reconnect within 60 seconds, it probably will never reconnect without performing a new login. You should calculate how long it takes your resources to migrate and determine the amount of time necessary to add to the tolerance. If you can't increase the tolerance sufficiently enough to overcome LAN congestion problems, you must either resolve LAN congestion problems or implement a dedicated heartbeat network.

LAN Card Issues

One of the most effective ways to resolve intentional abendss is to ensure that you have the latest NetWare Support Packs and the latest vendor-supplied LAN drivers installed. In most cases, updating LAN drivers will resolve intentional abends.

You should always manually set the LAN card driver to the same speed and duplex setting that is used on your switch. Also, manually set the switch to the proper speed and duplex setting for your network. Avoid using automatic speed or duplex detection on servers or the switch ports that the servers are connected to. This is even more important if you have hardware from different vendors. For example, you might be using an Intel* NIC with a 3COM* switch.

Another LAN card problem involves the improper implementation of the link-indicating counters. In a two-node cluster, Novell® Cluster Services™ watches the following counters to determine if a node is communicating on an Ethernet LAN:

- MLID_NUM_GENERIC_COUNTER
- NUM_GENERIC_MLID_COUNTERS
- NUM_ETHERNET_SPECIFIC_COUNTERS
- ETH_TX_ABORT_CARRIER_SENSE
- MLID_TOTAL_TX_PACKET_COUNT -- This should increment after the heartbeat packet is transmitted.
- MLID_TOTAL_TX_OK_BYTE_COUNT_LOW -- This should change after the heartbeat packet is transmitted.
- MLID_PACKET_TX_MISC_ERROR_COUNT -- This is monitored to see if it changes after the heartbeat is transmitted.

In a two-node cluster, Novell Cluster Services needs to determine the proper node to bring down in the event heartbeats are not getting through; it can't just assume that the master node is the good node and bring down the slave. It determines which node is good by monitoring the LAN counters and determining which node is actively communicating. If it can't determine which node is good, it will then bring down the slave node. Unfortunately, not all LAN drivers implement counters. If the LAN driver doesn't implement counters, the master node will always survive, and the slave node will always fail when the heartbeat is not received.

Note:

The feature to detect LAN traffic was added to the Novell Cluster Services 1.01 two-node tiebreaker patch. Without this patch, the master node always wins. This feature also exists in NCS 1.6 and will exist in later versions.

Tuning Server Configuration to Avoid False Split Brain Conditions

NetWare servers by default are tuned to support 200 to 400 client connections. If you have more than 400 connections, you need to modify several parameters to help avoid poison pill conditions.

Some of the parameters you might want to increase are listed in the following table:

Parameter	Explanation
Service Processes	If the server does not have enough service processes for all the processes running, performance might degrade to the point where a heartbeat is not transmitted within the specified tolerance.
Packet Receive Buffers	If the server does not have an available packet receive buffer when an incoming packet arrives, the server drops the packet and increments the packet receive buffers until it reaches the maximum. When the packet receive buffer reaches its maximum, it drops packets until it catches up and empties the buffer. By default, dropping eight seconds of consecutive packets is enough to assume that the monitored server is down.
LAN Speed	If the switch is set to 100 Mb and the server is set to 10 Mb (or vice versa), sporadic communications occur which will eventually cause an abend.
LAN Duplex	If a server is set to full duplex but the switch is set to half (or vice versa), slow and sporadic communications occur which will eventually cause an abend.

Categorizing the Four Types of Split Brain Conditions

In general, split brain conditions fall into one of four categories:

- Fatal SAN errors
- False node failure detections
- Split brain conditions
- Stalled self leave

For each of these categories, the following sections provide some basic troubleshooting ideas.

Fatal SAN Errors

Any cluster node that cannot read or write to the shared storage is essentially useless to the cluster and must be removed. If the node cannot read or write to the split brain detection partition, it will intentionally remove itself by eating one of the following poison pills:

- CLUSTER: Node castout, fatal SAN read error
- CLUSTER: Node castout, fatal SAN write error
- CLUSTER: Node castout, fatal SAN device alert

Each of these abends is caused by a fatal I/O error or device alert, which is signalled by the SAN device driver when invoked by the SBD.NLM module. As with nodes that can't communicate on the LAN, clean shutdowns to misbehaving nodes are problematic, so the node must force itself out of the cluster immediately by eating a poison pill. If you are receiving any of these fatal SAN errors, start by troubleshooting your hardware and the device drivers.

These abends are caused by the device driver passing a fatal error to Novell Cluster Services, which means that you have either a hardware fault or a problem with the driver. Check with hardware manufacturers for tools to help you troubleshoot hardware devices. A fatal SAN error generally signifies an error with the SAN hardware or the SAN driver on the server. For more information on SAN errors, see [SAN Related Problems](#).

In some cases, the fiber channel card tries to use High Memory, which causes major instability. Increasing the FILES and/or BUFFERS statements in the CONFIG.SYS beyond 100 to 150 prevents fiber channel cards from doing this and stabilizes the system.

Another common error with SAN implementations is a failure to match the fiber channel Host Bus Adapter (HBA) to the SAN topology. Many vendors have generic cards that work for PPP, FC-AL, and Fabric SAN implementations. In these cases, there might be a jumper or BIOS setting that can be used to configure the card for the proper SAN topology. In other cases, the hardware vendor requires different cards to match the topology. If this is the case, you can't use a PPP card if your server is attached to a fabric switch.

False Node Failure Detection

False node failure detection is different from a split brain condition in that the cluster thinks the node is dead due to a lack of heartbeat packets when, in reality, the node is alive. In a classical split brain condition, each side of the split brain thinks that the other side has failed and that it needs to start the other side's resources. In a false split brain condition, one side believes the other side has failed, while the other side thinks everything is fine.

There are two categories of false node failure detection:

- The node goes to sleep for a period of time, the cluster removes it, and then it wakes up thinking it's still in the cluster (sleepy node syndrome).
- The cluster thinks the node is dead, while the node still sees the cluster (divergent view syndrome).

Understanding the Sleepy Node Syndrome

There are currently three known situations where a node can appear to have failed, not perform its required abend, and then appear to come back to life. You should be aware of these situations so you can avoid them. Because other nodes take over the resources from the node that appeared to fail, if the node comes back to life and continues on, file system corruption will likely occur because the node believes it is still a member of the previous cluster and has no reason to believe it doesn't still

own the resources it had prior to going to sleep. The node continues on as before only until the next time it sends a heartbeat to the shared storage (four seconds, by default), at which time the node realizes that it was given a poison pill and abends.

Since there is potential for data corruption by a node writing to a volume that is already mounted elsewhere, the following three known cases of this occurring should be avoided.

The first case of a sleepy node occurs when a node enters and stays in Real Mode for a period of time equal to or greater than the heartbeat threshold parameter. Because the NetWare floppy disk driver can execute in Real Mode longer than the threshold period, avoid using the floppy drive from a cluster node. If you must use the floppy disk, copy any NLM™ programs from the floppy to the server, and then run them from the server. Loading an NLM directly from a floppy will cause the server to stay in Real Mode too long to respond to the heartbeats from other nodes. It will likely cause a false node failure detection problem.

The second case occurs when an administrator suspends a node by bringing it into the system kernel debugger and then restarts the node after the threshold parameter has passed. If you need to use the kernel debugger, use the CLUSTER DEBUG cluster console command, which halts all nodes, or use the HTML-based NetWare Management Portal and its nonintrusive debugging tools.

If you switch to the system kernel debugger, make sure you either remove the server from the cluster with the CLUSTER LEAVE command or execute the Quit (Q) command to exit to DOS.

Note:

Novell Cluster Services SP2 for NetWare 5 and Novell Cluster Services 1.6 for NetWare 6 address kernel debugger problems. These versions do not allow you to type G or T in the debugger unless the SET parameter for the Developer Option is set to On.

The third case which can contribute to sleepy node syndrome is CPU Hog. Software that hogs the CPU can contribute to a poison pill abend.

It is important to understand CPU Hog, because unless you make some modifications to your configuration, you will never know the problem is a CPU Hog, or which module is causing the problem.

An application is considered a CPU Hog if it fails to voluntarily release the CPU as required. If an application takes longer to release the CPU than the heartbeat tolerance, a poison pill occurs because the cluster was not allowed to transmit the required heartbeats. Because the default CPU Hog timeout interval is set to 60 seconds, you would never know that the cause of the abend was a specific application hogging the CPU.

To help determine if one of your applications is hogging the CPU for too much time, adjust the CPU Hog Timeout Amount server parameter to a value less than the heartbeat tolerance parameter. This will cause cluster servers to abend with a CPU hog in the problematic module rather than abending due to a poison pill. Once you identify the problematic module, you can determine the best way to resolve the problem.

Due to the single-threaded nature of bindery services, you should also eliminate bindery emulation on all of your cluster nodes. Excessive bindery contexts could contribute to a CPU hog problem, but

would not point you to a specific module that is causing the problem.

To deal with a CPU hog of another sort, we highly recommend that you eliminate IPX™ from your cluster nodes because all resources must be serviced via TCP/IP to allow automatic reconnection. If you cannot eliminate IPX, then you should at least eliminate the use of the IPXRTR module.

This module has a tendency to periodically hog the CPU for 10 or more seconds, which normally would not be a problem but with a cluster, it is sufficient to cause false poison pill conditions. In many cases, stability problems are completely resolved by eliminating the use of IPXRTR.

Note:

Using INETCFG to disable IPX routing does not remove the IPXRTR module. Consider placing the load and bind commands for IPX in the AUTOEXEC.NCF instead of using INETCFG if you can't eliminate IPX altogether.

Understanding the Divergent View Syndrome

The second category of false node failure detection is the case where the cluster doesn't see the node's heartbeat but the node does see the rest of the cluster. This can be caused when the node's LAN transmit is not functioning but the receive is. Similar issues could arise if the master node's multicasts can't get through a switch but a slave's unicast packets can. This would result in slaves thinking the master is dead, while the master sees all of the slaves as alive.

This situation typically results in an abend with a message similar to "Ate poison pill in XXX given by some other node," with XXX varying depending on the specifics of the communications problem. See the table on [Split Brain/Poison Pill ABENDS below](#) for more information.

For the condition described above, consider the following potential solutions:

- Replace the misbehaving node's NIC
- Replace the misbehaving node's LAN cable
- Verify that the spanning tree is not interfering in the communications
- Use a protocol analyzer to help pinpoint the communications issue

Split Brain Conditions

Split brain conditions generally occur as a result of LAN hardware or software problems. A split brain is a condition where not all of the nodes agree on which servers should be members of the cluster. There is a split in agreement on the view of the cluster membership, with each side of the split thinking that the other side failed.

The following list provides recommendations to help you troubleshoot split brain conditions:

- Check on the server's LAN driver and protocol stack statistics with the LSLSTAT command

Look for excessive NO ECB Available errors as well as any other types of packet errors.

- Check all connectivity devices between the cluster nodes
 - Are the cables properly seated and connected?

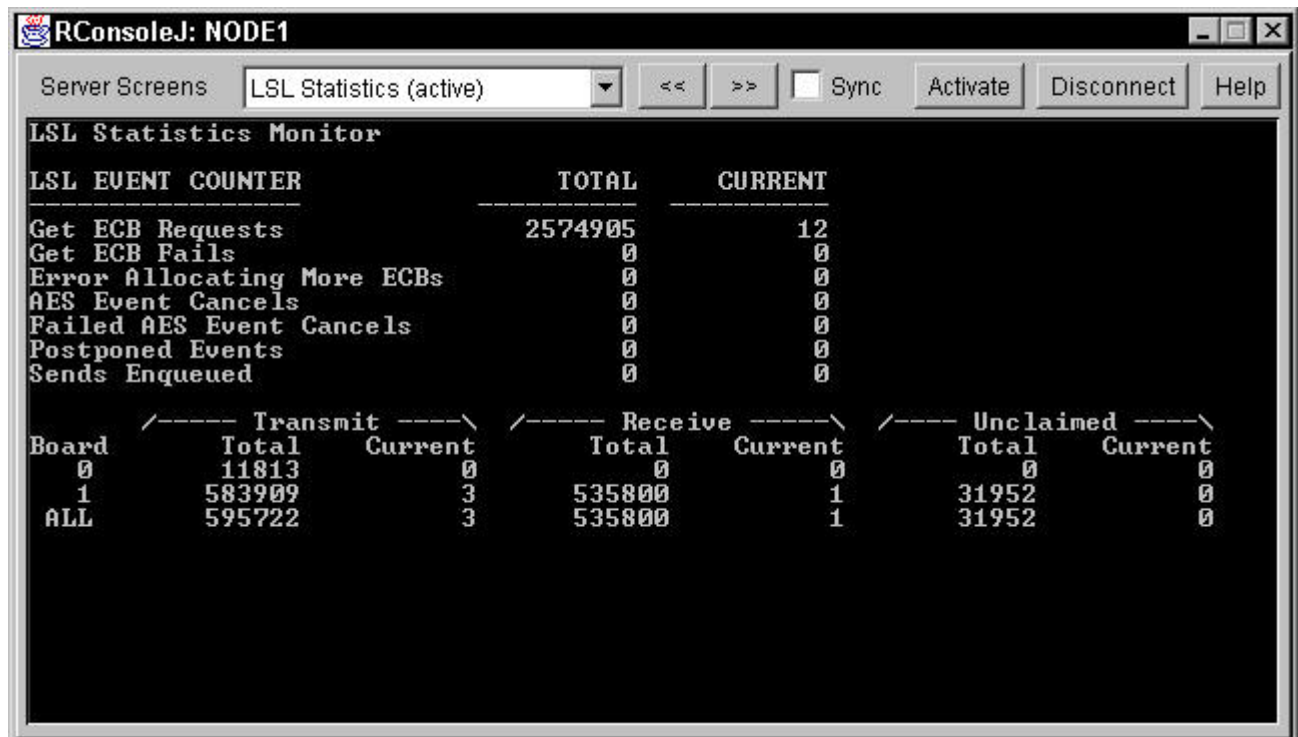
- Are all the hubs powered on?
- Did an administrator change the configuration of a switch, which required the switch to shut down and restart?
- Check the LAN switch for settings that can delay packets
 - Is the switch port manually set for speed and duplex?
 - Does the switch port configuration match the settings on the server's driver?
 - Is PortFast/spanning tree configured properly for your environment?
 - Did spanning tree reset all bridge ports and cause packet delay?
- Check that the server has sufficient resources to communicate on the LAN
 - Are packet receive buffers at their maximum? If so, increase the maximum.
 - Are service processes at their maximum? If so, increase the maximum.

In addition to the above configuration issues, also check and resolve any LAN connectivity issues. Use the appropriate diagnostic tools for your LAN (sniffer, probes, etc.) to determine if your LAN is causing delays due to congestion, misbehaving hosts, etc.

You can also try increasing the cluster tolerance and slave watchdog parameters, which cause the servers to be more tolerant of delayed packets. However, be aware that increasing the parameters excessively might negatively affect automatic client reconnection.

If you can't diagnose or resolve the above-mentioned problems, consider implementing a dedicated heartbeat network to isolate the heartbeat traffic. You might have too much traffic on the production LAN for the heartbeat to function efficiently.

NetWare 5.1 and NetWare 6 include a new diagnostic tool that can help you troubleshoot LAN errors at the LSL™ layer. This tool produces an LSL Statistics Monitor screen that can help you determine if you are having issues with dropped packets due to Event Control Block issues. An example of the LSL Statistics Monitor screen is shown below.



LSL Statistics Monitor

LSL EVENT COUNTER		TOTAL	CURRENT
Get ECB Requests		2574905	12
Get ECB Fails		0	0
Error Allocating More ECBs		0	0
AES Event Cancels		0	0
Failed AES Event Cancels		0	0
Postponed Events		0	0
Sends Enqueued		0	0

Board	Transmit		Receive		Unclaimed	
	Total	Current	Total	Current	Total	Current
0	11813	0	0	0	0	0
1	583909	3	535800	1	31952	0
ALL	595722	3	535800	1	31952	0

Categorizing Various Split Brain/Poison Pill Abends

The following table lists some of the most common split brain/poison pill abends, their categories, and descriptions.

ABEND	Category	Description
CLUSTER: Node castout, fatal SAN read error.	Fatal SAN error	The SAN device driver detects a fatal (nonrecoverable) error while reading from the shared storage.
CLUSTER: Node castout, fatal SAN write error.	Fatal SAN error	The SAN device driver detects a fatal (nonrecoverable) error while writing to the shared storage.
CLUSTER: Node castout, fatal SAN device alert.	Fatal SAN error	The SAN device driver detects a fatal (nonrecoverable) error while communicating or attempting to communicate with a shared storage device.
Ate poison pill in sbdProposeView given by some other node.	False Node Failure Detection	Communications problems cause a divergent view between this node and the cluster.
Ate poison pill in sbdWriteNodeTick given by some other node.	False Node Failure Detection	Communications problems cause a divergent view between this node and the cluster.
Ate poison pill. Link is down. Other node is alive and ticking.	Split Brain Condition	A two-node cluster condition where the LAN counters are incrementing on the other node but aren't incrementing on this node. Indicates a LAN card or driver failure.
This node is in the Minority partition and the node in the Majority partition is alive.	Split Brain Condition	A split brain condition where this server is on the losing side of the vote.
At least one of the nodes is alive in the old master node's partition. This node is not in the old master node's partition.	Split Brain Condition	A split brain condition where there is a tie vote. In the case of a tie, the side with the master node wins. This server is on the side that does not contain the master node.

The alive partition with the highest node members should survive. This node is not in the alive partition with highest node number.	Split Brain Condition	A split brain condition where the master node is not available because it left the cluster or failed. The side that contains the most nodes wins. This server is not on that side.
This cluster node failed to process its self-leave event in a timely fashion and will be forced out of the cluster.	Stalled self-leave	The node tries to leave the cluster, but for some reason it stalls. Because it does not leave cleanly, it is impossible to guarantee that the resources are safe to start on new nodes. The node must bring itself down so that resources safely start elsewhere in the cluster.
CRM: CRMSelfLeave: Some resources went in comatose state while SelfLeave.	Stalled self-leave	This situation is similar to the previous one except that a failure is detected while running a resource unload script. It isn't safe to assume this node cleanly stopped the resources that were running on it, so it removes itself from the cluster via an abend to allow the clean start of resources on another node.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark. For information on trademarks, see [Legal Notices](#).