# Resolving Critical Server Issues

**RICH JENSEN**
Product Support Engineer
Worldwide Customer Services
and Support

The purpose of this document is to help network administrators become more proactive in resolving critical server issues (abends and hangs). In the past, recommended procedures for handling server crashes have not been clearly set down in writing. By creating this document, Novell Support hopes to minimize miscommunication when dealing with customers and educate them as to how they can best help Novell resolve server issues. This document provides a standardized way to obtain valuable feedback to questions that will help prioritize the issue, gain some historical perspective on the problem, and improve resolution time.

ACKNOWLEDGEMENTS

Special thanks to the following people for their assistance with this AppNote:

Alan Blauer, Software Engineer, Critical Problem Resolution Team
Tim Heaton, Software Engineer, Critical Problem Resolution Team
Carrie Labonty, Product Support Supervisor, Worldwide Customer Services and Support
Randy Lee, Technical Support Engineer, Worldwide Customer Services and Support
Dale Maughan, Product Support Engineer, Worldwide Customer Services and Support
Troy Rovig, Software Engineer, Critical Problem Resolution Team
Jim Sorenson, Software Engineer, Novell Consulting Services

DISCLAIMER

Novell, Inc. makes no representations or warranties with respect to the contents or use of these Application Notes (AppNotes) or of any of the third-party products discussed in the AppNotes. Novell reserves the right to revise these AppNotes and to make changes in their content at any time, without obligation to notify any person or entity of such revisions or changes. These AppNotes do not constitute an endorsement of the third-party product or products that were tested. Configuration(s) tested or described may or may not be the only available solution. Any test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state, or local requirements. Novell does not warranty products except as stated in applicable Novell product warranties or license agreements.

Novell, Inc.
122 East 1700 South
Provo, Utah  84606  USA

# Introduction

Few events strike terror into the hearts of network administrators as much as a server going down unexpectedly. But with sufficient troubleshooting information and a few advance precautions, you'll be better prepared to handle server problems proactively.

This AppNote presents recommended guidelines and procedures for customers to follow when resolving critical server issues. It begins with a discussion of server abends and lockups and their possible causes. It then gives some troubleshooting steps to follow to identify and resolve these types of server problems. An appendix gives instructions for capturing a server memory image for analysis by Novell.

By creating this document, Novell Support hopes to minimize miscommunication when dealing with customers and educate them as to how they can best help Novell resolve server issues. This document provides a standardized way to obtain valuable feedback to questions that will help prioritize the issue, gain some historical perspective on the problem, and improve resolution time.

**What is a Critical Server Issue?**
For the purposes of this AppNote, we define a "critical server issue" as a situation in which the server ceases operation unexpectedly. The server may simply stop running or become unusable, thus preventing any work from being done by clients connected to the server or with applications running at the server. Such conditions are generally described as server crashes, hangs, or "abends."

# What is an Abend?

The NetWare 3 and 4 operating systems continually monitor the status of various server activities to ensure proper operation. If NetWare detects a condition that threatens the integrity of its internal data (such as an invalid parameter being passed in a function call, or certain hardware errors), it abruptly halts the active process and displays an "abend" message on the screen. ("Abend" is a computer science term signifying an ABnormal END of program.)

The primary reason for abends in NetWare is to ensure the stability and integrity of the internal operating system data. For example, if the operating system detected invalid pointers to cache buffers and yet continued to run, data would soon become unusable or corrupted. Thus an abend is NetWare's way of protecting itself—and users—against the unpredictable effects of data corruption.

There are two basic types of errors that can cause abend messages to be generated:

- Errors detected by the CPU
- Consistency check errors (detected by the operating system)

## CPU-Detected Errors

When the server's CPU detects an error, the processor can interrupt program execution by issuing an interrupt or an exception.

Intel defines an *interrupt* as "an asynchronous event typically triggered by an external device needing attention."

Intel defines an *exception* as "a synchronous event which is the response of the processor to a certain condition detected during the execution of an instruction."

Exceptions are classified as faults, traps, or aborts based on how they are reported and whether restart of the failed instruction is possible.

Here is a list of exceptions and interrupts:

1. Divide Error
2. Debugger Call
3. Nonmaskable Interrupt (NMI)
4. Breakpoint
5. INT 0-detected Overflow
6. BOUND Range Exceeded
7. Invalid Opcode
8. Device Not Available
9. Double Fault
10. Invalid Task State Segment
11. Segment Not Present
12. Stack Exception
13. General Protection
14. Page Fault
15. Floating-Point Error
16. Alignment Check
17. Maskable Interrupts

The types of exceptions that are related to abends are the nonmaskable interrupt (NMI) and the processor-detected exceptions.

For more complete details about exceptions and interrupts, refer to Chapter 9 of the *Intel486 Microprocessor Family Programmer's Reference Manual*.

**Paging and Segmentation Exceptions**
NetWare 4 takes advantage of Intel's segmentation and paging architecture. Each page of memory can be flagged present or not present, read-protected, write-protected, readable, or writable. These changes in NetWare 4 introduce new exceptions that are not seen in NetWare 3. One good example is the "Abend: Page Fault" error.

Exceptions caused by segmentation and paging problems are handled differently than interrupts. Normally, the contents of the program counter (EIP register) are saved when an exception or interrupt is generated. However, exceptions resulting from segmentation and paging give the operating system the opportunity to fix the page fault by restoring the contents of some of the processor registers to their state before interpretation of the instruction began. NetWare 4 provides SET parameters to enable and disable page fault emulation, giving you the choice between continuing pro-gram execution or abending.

## Consistency Check Errors

Consistency checks are internal tests which Novell software engineers have placed in the NetWare operating system code. The primary function of consistency checks is to ensure the stability and integrity of internal operating system data. Numerous consistency checks are interlaced throughout NetWare to validate critical disk, memory, and communications processes. The abend errors that result from failed consistency checks are code-detected errors, as opposed to CPU-detected errors.

As an example of a consistency check, imagine a function called XYZFreeMemory that is used to release a portion of memory so it will be available for other programs. To guard against possible problems, the programmer includes a check to see whether the pointer passed into the function points to a valid memory buffer. If this check fails, the system will generate an abend.

A failed consistency check is always a serious error because it indicates some degree of memory corruption. Consistency check errors might be caused by a corrupt operating system file, corrupt or outdated drivers and NLMs (NetWare Loadable Modules), bad packets formed at the client, or hardware failures. These errors can also be associated with defective memory chips, static electricity discharges, faulty power supplies, or fluctuations in commercial power (see *NetWare System Messages* manual, page 1).

## Analyzing Abend Messages

Before NetWare displays an abend message on the file server screen, several steps occur depending on whether the error was CPU-detected (exception generated) or code-detected (consistency checks). The type of information provided on the screen is identical in both cases:

(Line 1) Date and time the system halted
(Line 2) Abend message
(Line 3) Operating system version
(Line 4) Current running process
(Line 5) Current stack dump

For ease of reference, we'll refer to line numbers 1, 2, 3, 4, and 5 in the sample abend message screens below.

Note: In the NetWare 3.12 operating system, EIP was added to the information on exceptions generated by the CPU.

**Line 1: Date and Time.** NetWare first posts the date and time at which the system was halted.

**Line 2: Abend Message String.** The text of the abend message itself will help you determine whether it is a CPU-detected abend or a code-detected error. In many cases, it's easy to tell whether the message contains only information provided by the CPU or information from the operating system.

Here is an example of a CPU-generated abend:

```
(1)  System halted Friday, July 22, 1994   3:32:42 pm MDT
(2)  Abend: Page Fault Processor Exception (Error code 00000000)
(3)  OS version: Novell NetWare v4.02  June 8, 1994
(4)  Running Process: Server 03 Process
(5)  Stack:          02 72 00 00 D7 BB 02 F8 AC B9 EE 00 C0 B9 EE 00
                     60 70 2B 00 78 B9 EE 00 58 92 05 F1 D0 FF 08 00
                     94 B9 EE 00 97 D6 00 F1 D0 FF 08 00 00 00 00 00
   Press "Y" to copy diagnostic image to disk.  Otherwise
   Power off and back on to restart.
```

Notice the text of the message on line 2, "Abend: Page Fault Processor Exception (Error code 00000000)". This information is provided to the operating system by the CPU. The error code in the message is used to help determine additional information about the exception. Error codes are produced only for some exceptions. Under certain conditions, exceptions which produce error codes may not be able to report an accurate code.

Here is an example of a code-generated abend:

```
(1)  System halted Tuesday, October 4, 1994   9:59:08 am PDT
(2)  Abend: SERVER-4.00-3128: SubAllocFreeSectors given invalid FAT
     chain end that was already free.
(3)  OS version: Novell NetWare v4.02  June 8, 1994
(4)  Running Process: Console Command Process
(5)  Stack:            3C 9E 0D F8 AB 57 27 00 01 00 00 00 20 00 01 00
                       01 00 00 00 00 00 00 00 10 B7 B5 0E A0 3E 56 00
                       01 00 00 00 15 20 01 F8 01 00 00 00 00 00 00 81
     Press "Y" to copy diagnostic image to disk.  Otherwise
     Power off and back on to restart.
```

Notice how this abend message is different from the CPU-detected abend above which was generated by an exception. The message in line 3 refers to a consistency check found in the NetWare 4 operating system code (“SERVER-4.00-3128"), along with a short description of what that check was. In this example message, the error was found in the SubAllocFreeSectors routine which checks the FAT chain to see if it is a valid SubAlloc block.

**Line 3: Operating System Version.** This line identifies the version of the NetWare operating system running in the server.

**Line 4: Running Process.** This line indicates which process was running at the time of the abend. A “process” is a thread or path of execution that runs in the operating system. It can be an internal OS process or a process belonging to an NLM. Internal server processes can be referred to as *OS worker threads*. These are processes that take on a wide variety of tasks, such as handling packets, processing NCP requests, and performing work from the work-to-do list. Some of these tasks can be scheduled by other NLMs and carried out by file service processes. NLMs can also have their own dedicated threads.

Although the server message indicates which process was currently running at the time of the abend, you can’t assume that the running process is the cause of the abend. It may or may not be involved.

A good example of a case in which the running process is not the cause is when a process (call it Process A) receives an invalid pointer from a corrupt memory area and then tries to use this pointer. The memory area possibly became corrupt because some other process (Process B) issued a write over a valid structure or pointer. The running process simply tries to execute this pointer, which results in the abend. So even though Process A is identified as the running process in the abend message, the problem actually lies with Process B.

Another example is when the running process is passed invalid information from another NLM. File service processes fall under this scenario because they carry out work for other NLMs and service incoming packets that can pass invalid or corrupt information to the server process to execute.

Note: In abend messages, file service processes are identified as “Server XXX Process” where XXX can be any number between 0 and 100.

**stack:** an area of memory set aside for the temporary storage of values in a computing environment.

**Line 5: Stack.** The 30 hexadecimal bytes displayed at the bottom of the abend screen represent part of the CPU’s stack at the time of the abend for the current running process. All three lines of the stack dump may be useful to technical support people in diagnosing the cause of the abend.

## Server Hangs or Lockups

In the computer industry, people describe a machine that suddenly stops working with a variety of frightful terms. They say the computer has crashed, frozen, hung, or locked up. For the purposes of this discussion, we’ll distinguish between *full* and *partial* server lockups.

When a full server lockup occurs, no processes are allowed to run. No one can log in to do work on the server. Connections that are currently logged in or attached are dropped. Nothing can be done at the server console or other NLM screens, and there may be no response at all from the server keyboard.

After a partial server lockup, users might still be able to log in to the server and accomplish work. In some cases, you may be able to toggle to different server or NLM screens and do work. Partial hangs may eventually clear themselves up, or they may lead to a full system lockup.

**The Nonpreemptive Environment**
Because the NetWare operating system is nonpreemptive, it allows threads to access and control the CPU as they choose. The underlying assumption is that NLM processes will cooperate with each other and not monopolize the processor. In this type of environment, threads need not worry about being forced off the CPU unless they monopolize it. However, they can and should relinquish control frequently to allow other threads a chance to run.

One possible cause of a server lockup is a server or NLM thread which becomes caught in a tight loop and does not relinquish control of the CPU. The cause for this type of lockup can be related to either software or hardware problems.

Another example is a process which locks up resources (volumes, cache buffers, and so on) by blocking access to these resources. Other processes waiting on the release of these resources will not run until they are available. Again, the cause for this type of lockup can be software or hardware.

Server lockups can also be caused by some of the same problems that cause abends: corrupt operating system files, corrupt or outdated drivers and NLMs, bad packets formed at the client, or hardware failures.

Here's a sample case that involved the use of outdated software. The customer was using the BNETX NetWare shell on the client for packet burst communications with a NetWare 4.02 server. (The BNETX shell was developed for use with the original PBURST.NLM and was intended for use only with NetWare 3.11.) Because BNETX was out of date, the client was not communicating properly with the server. This miscommunication caused the server to hold resources and not release them for long periods of time. The longest period of delay time experienced was two hours. During that time, all any other processes could do was wait for the server resources to be freed up.

In diagnosing the cause of a server lockup, it is sometimes useful to generate a memory image file (or core dump) that lists the entire contents of server RAM. The steps for doing this are outlined in the Appendix of this AppNote.

## Steps for Troubleshooting NetWare Servers

Like any sophisticated piece of software, the NetWare operating system is very complex and dynamic. In a network, a large number of components work together to form a functional whole. Each component has one or more specific relationships to other components in the system. A network is dynamic because it is subject to change. These characteristics of a network can make it difficult to pinpoint the exact cause of problems.

By following the troubleshooting steps outlined below, you can eliminate some of the obvious problems and provide more accurate information for the support technician if needed.

**Server Troubleshooting Steps**

1. Gather information about the problem.
2. Understand the problem and identify probable causes.
3. Test possible solutions.
4. Use debugging tools, if necessary.
5. Resolve the problem.

### Step 1. Gather Information About the Problem

When faced with a critical server issue, you should gather the following facts:

A. All error messages that are generated.
B. Complete hardware configuration of the server.
C. Disk and LAN driver information for the server.

D.  Listing of current NLMs and NCF files on the server.
E.  The most recent changes made to the system.
F.  Events that occurred prior to the crash.

**A. Error Messages.** All error messages need to be gathered and analyzed near the time of the system crash. There are many places to gather error information. One of the first is the abend information screen. Another is the server console screen where some console message might still be displayed.

After the server is brought back up, the system error log is a good place to look for date and time information. Another often overlooked area is the volume error logs.

**B. Hardware Configuration.** List all hardware components that make up the server. Find certification and testing information on these components.

**C. Disk and LAN Drivers.** Put together a complete listing of LAN and disk drivers running on the server, along with their date and version information.

**D. NLMs and NCF Files on the Server.** Put together a complete listing of NLMs running on the server, along with their date and version information. Also obtain a listing of both the STARTUP.NCF and AUTOEXEC.NCF files to show how the NLMs were loaded.

**E. Recent Changes to the System.** Network administrators should maintain a log for each server to record both hardware and software changes. These records can help determine if the system has a history of stable operation, and whether or not this is a problem seen before on this system. This information could be very important in resolving the problem.

**F. Events Occurring Prior to the Crash.** Gather a sampling of what activities were taking place on the network at the time of the abend or hang. These might include events such as system maintenance (backups, database rebuilds, and so on), installation or changes in software or hardware, system failures, errors and warnings. Also make a note of user activities (high workload, atypical activities such as month-end closing, and so on).

Using CONFIG.NLM. To help in the gathering of this information, Novell Support provides an NLM called CONFIG.NLM. CONFIG.NLM creates a text file called CONFIG.TXT in SYS:SYSTEM. This file contains a list of all modules loaded on the server at the time CONFIG.NLM is run. It also contains the contents of the STARTUP.NCF, AUTOEXEC.NCF, CONFIG.SYS, and AUTOEXEC.BAT files for the server. A directory of SYS:SYSTEM and your local drive is also placed in CONFIG.TXT.

Download this NLM from the NSD area of NetWire. The self-extracting file is named CONFIG.EXE. (For more information on CONFIG.NLM, refer to Technical Information Document

TID021808 entitled “CONFIG.NLM”; the Research Index at the back of this AppNote issue gives availability information on Novell technical bulletins.) To run this module, you must have the latest CLIB.NLM loaded on your server. (Updates to CLIB can be found on NetWire in LIBUPX.EXE.)

## Step 2. Understand the Problem and Identify Probable Causes

Understanding the problem comes by answering questions about the information and facts gathered in Step 1. Some of the types of questions you might ask are the following:

- Can I draw any conclusions from the information gathered?
- What information from the server error log file, volume error log file, and other audit-type files, could relate to the abend message or hang?
- Is the hardware configuration different from one that has been certified and tested?
- How are the drivers and NLMs loaded for this hardware configuration?
- Are the drivers and NLMs on the file server up to date and current?
- Have all the tested and approved patches been applied to the operating system?
- When did this problem occur? For example, did it occur while trying to boot the file server, and if so, at what point did the failure occur?
- What can I still do at the server? For example, if the system is in a hung state, can I toggle to different screens? Is the server partially or totally locked up?

Once you have a good understanding of the problem, try to identify some probable causes by drawing conclusions from the information gathered and forming one or more hypotheses.

As an example, suppose you just finished adding a new network card to the server and the server hangs next time you bring it up. After going through the information-gathering suggestions listed above, you arrive at two possible causes:

*Hypothesis 1.* Since you’ve just added a new network card, there’s a pretty good chance that this is the cause of the problem.

*Hypothesis 2.* The server might be experiencing file corruption resulting from a power outage or drive failure.

The above questions and hypotheses are just a few examples of many that could be determined from the information provided.

## Step 3. Test Possible Solutions

There are several methods or techniques you can use to test your hypotheses. Following are some of the most common ones.

**Apply Known Patches and Fixes.** Over half of the server abends and lockups reported to Novell Support are resolved by patches that have already been written. This should be one of the first areas to check in testing possible solutions to a problem, as it can save you many hours of troubleshooting previously resolved issues.

Be sure to apply *all* approved and tested operating system patches, *regardless of the problem*. A self-extracting EXE file for each operating system is available on NetWire and on the NSEPro CD-ROM. Novell uses the following naming convention for these files: The first three digits represent the OS version, followed by PT or IT (which stand for Passed Test or In Test), and a revision number. For example, 311PTD.EXE, 312PT1.EXE, 401PT1.EXE, and so on.

**Component Swapping.** One technique that is often used is swapping or replacing the suspected faulty component with a similar component that is known to be good. This method is most effective when you are familiar with the expected behavior of each component and already have a good idea of what could be causing the problem.

It's vital that you swap out only *one* component at a time. This technique is effective for both hardware and software problems.

**Divide and Conquer.** To make it easier to isolate a problem, remove components from the system. For example, unload unneeded NLMs and hardware components to simplify the system.

**Discuss the Problem with Others.** A good way to gain valuable feedback about a problem is to discuss possible solutions with other experienced CNEs and Novell support engineers.

## Step 4. Use Debugging Tools

If you have not been able to gather enough information to make conclusions about the abend or hang, the use of additional debugging tools such as network analyzers, along with a memory image from the server, can help in resolving server abends or hangs.

**Network Analyzers.** Network analyzers (such as Novell's LANalyzer, Network General's Sniffer, and so on) are great tools for gathering troubleshooting information. In many cases, knowing about the behavior of protocols and packets on the network can help speed up the resolution of the problem.

**Memory Image File.** If the problem still exists after you have taken all of the above steps, there is another useful tool available to you. That is to create a memory image or "core dump" of the server and send it to Novell Technical Support for analysis. This memory image provides a snapshot of your server at the time of the abend.

Note: Before sending in a memory image, make sure all the tested and approved NetWare patches have been applied to the server.

Although a memory image shows what was occurring at the time of the abend, it does not provide much of a history. Often, though not always, the memory image provides enough information for Novell engineers to correctly diagnose your problem. Sometimes they can learn enough from the memory image to duplicate the issue on an identical machine in Novell's server lab.

The Appendix of this AppNote contains information on how to obtain a memory image file and how to send it to be analyzed.

The Information Sheet. To assist in the problem resolution process, an Information Sheet is included with this AppNote. Fill in the information requested and send it in along with your server memory image and LANalyzer or Sniffer trace. If you can recreate the problem and describe exactly what steps led up to the abend, record this on the Information Sheet as well. This information will help speed up resolution time, reduce the chance of miscommunication, and keep the technical representative focused on the problem.

If Novell's engineers are able to correct the problem, and if the problem has been caused by a software bug in the operating system, they will debug the program and send you a patch for the problem.

## Step 5. Resolve the Problem

Once the problem has been isolated and you have proven your hypothesis correct, it is time to resolve the issue. For software issues, you can resolve problems with patches, workarounds, new drivers, and so on. For hardware, repair or replacement are the options.

The troubleshooting steps outlined above can be used for most abend errors on a NetWare server. If these steps do not resolve the problem, contact your Novell Authorized Dealer or Novell Technical Support for assistance.

## Additional References

Here are some useful references if you desire further information about microprocessor exceptions and NetWare Loadable Modules:

*Intel486 Microprocessor Family Programmer's Reference Manual*, Intel Corporation, 1992

Michael Day, Michael Koontz, Daniel Marshall. *Novell's Guide to NetWare 4.0 NLM Programming*, Novell Press, 1993

NOVELL®

# Critical Server Issue Information Sheet

Please fill out this information sheet and send in prior to Memory Image (Core Dump).

TO: FROM:

INCIDENT # (an open incident is required):

1. When did the abend problem start? Did it coincide with installation time, adds/changes to the hardware or software etc.?

2. Is the abend message always the same (same abend type and/or running process)?

3. Does it happen at a certain time of day or during a certain activity? What events occur prior to the abend or hang? Are there any noticeable changes in monitor statistics?

4. Can the problem be duplicated at will or is it intermittent?

5. How often does the abend occur? How many times?

6. Have the current standard patches for your version of the NetWare operating system been applied?

7. Are all of the NLMs and drivers up to date? Are they all certified?

8. What are the details on the server hardware and configuration (type of machine, RAM, controllers, NICs, NCF files, types of clients, and so on)? Run CONFIG.NLM and send in CONFIG.TXT.

9. What other symptoms or error messages occur prior to the abend?

**NUMBER OF PAGES (INCLUDING COVER SHEET):**

Novell, Inc. • 122 East 1700 South • Provo, Utah 84606 USA
Phone (800) NETWARE • Fax (801) 429-5200

# Appendix: Memory Images (Core Dumps)

The term “core dump” comes from the mainframe world, where RAM memory was (and still is) referred to as core memory because of the way data was stored in ferrous magnetic cores--little round doughnut-shaped objects made out of ferrous (iron-based) material. Today’s micro-computers don’t store data in this manner, but a PC’s system RAM is still occassionally referred to as the core.

A memory image or core dump is a byte-for-byte image of a NetWare server’s memory—a “snapshot” of a server’s RAM at the time it abended. The terms “core dump” and “memory image” are interchangeable; for consistency’s sake, we use “memory image” in this AppNote.

On a NetWare server, a memory image copy can be initiated in one of three ways:

1. By answering the prompts generated by NetWare after a server abend has occurred (see the example abend screens depicted earlier in this AppNote).

2. By manually activating the NetWare debugger (simultaneously press <LeftShift>+<RightShift>+<Alt>+<Esc>) and issuing the “.c” command.

3. By causing the CPU to issue an NMI exception, using an approved method from the PC hardware vendor.

Next, you need to choose the path and the method you’ll use to copy the memory image file to a storage medium. These steps are described under “Choosing the Path” and “Choosing the Method” below.

**Forcing a Memory Image Copy.** You might need to force a memory image copy under the following circumstances:

- You encounter an error (such as a server hang or lockup) and do not see the option to do the memory image copy.
- A server may exhibit strange behavior but not display any errors, and a Novell Service Representative may request the memory image file to see some internal details.

Here are the steps to follow to force a memory image copy:

1. If the server is running, press the following keys simultaneously to enter the debugger: <LeftShift> + <RightShift> + <Alt> + <Esc>. If the keyboard does not respond, generate an NMI as described above.

2. From the debugger, type “.c” to start the diagnostic image copy.

3. When the copy is finished, press “G” to enable NetWare to continue (provided the server is not locked up). Otherwise press “Q” to quit to DOS.

## Choosing the Path

With NetWare 3.12 and later (including all versions of

NetWare 4), the user can specify the drive letter to which the memory image file will be copied. This drive can be any writable DOS device, even a network drive on another file server that was mapped under DOS prior to booting the server. The size of the image file will be approximately equal to the total RAM installed in the server.

Note: For NetWare 3.11, a NetWare Loadable Module called HDUMP.NLM is available which allows you to write the image file to a local DOS partition or network drive instead of to a floppy drive. Download this NLM from the NSD forum on NetWire. The self-extracting file is named HDUMP.EXE. (For more information on how to use HDUMP.NLM, refer to Technical Information Document TID350119 entitled "Dumping Memory Dumps to DOS"; the Research Index at the back of this AppNote edition gives availability information for Novell technical bulletins.)

## Choosing the Method

Four methods are available for copying the memory image file to a storage medium.

**Floppy Drive Method.** If the image is copied to a floppy drive, the user will be prompted to insert formatted diskettes. Be sure you have sufficient diskettes on hand to copy all of your machine's RAM. For example, to copy 12 MB of RAM, you'd need nine 3 ½-inch high-density (1.4 MB) diskettes.

This method is not necessarily a good one to use because bad sectors on a floppy diskette could cause the image to be invalid or unusable when being analyzed.

**Hard Drive Method.** When the image is copied to a local hard drive on the server, the name of the image file is COREDUMP.IMG by default. Once the image file is on the hard disk, it can be com-pressed, copied to diskettes, backed up to tape, or sent by FTP to novell.com (see the instructions under "Sending the Image File to Novell").

The image file can also be copied to a NetWare drive later, after the server is up and running. This can be done by using a NetWare Loadable Module called IMGCOPY.NLM or any other third-party NLM that provides this functionality.

Note: IMGCOPY.NLM is included in the self-extracting file HDUMP.EXE which can be found in the NSD forum of NetWire. For more information about using this module, refer to the readme file included with the download.

**Network Drive Method.** If this method is used, some advance setup is required prior to the abend or hang. To wit, the problem server must have an extra LAN card installed, and a *client* ODI driver must be loaded for this card. (This is possible as long as you load the client driver in DOS conventional memory. The server drivers load in extended memory, so both types of drivers can be loaded at the same time.)

You'll also need to obtain and load a NetWare Loadable Module called NETALIVE.NLM, which can be found in the NSD forum on NetWire, in the self extracting file NETALV.EXE. This module keeps a client connection alive underneath an active server when two LAN cards are used. (For more information on using this module, refer to Technical Information Document TID021885 entitled "NETALIVE.NLM"; the Research Index at the back of this AppNote edition gives availability information for Novell technical bulletins.)

When an abend occurs, proceed as follows:

1. Boot the problem server as a client (when using VLMs, you'll need to make one small change to the CONFIG.SYS file: that is, set LASTDRIVE=Z).

2. From this "client," log in to a healthy server elsewhere on the network.

3. Map a drive to a *volume:directory* on the healthy server. The volume must have enough free disk space to copy the problem server's memory image file. Record the complete path (for example, f:\sys:\cdump).

4. Now boot the client as a server by running SERVER.EXE from the DOS partition or boot diskette. The server comes up, but DOS is still loaded. Therefore, until NetWare's watchdog function kills the connection to the extra LAN card, a connection to the healthy server can be maintained via NETALIVE.NLM.

5. If you are running NetWare 3.11, load HDUMP.NLM with the recorded path on the command line. For example:

   ```
   load hdump f:\sys:\cdump <Enter>
   ```

   With post-3.11 NetWare, the abend screen gives you the opportunity to specify a different path. This is where you would specify the path you recorded in Step 3 (for example, f:\sys:\cdump).

By default, the name of the image file is COREDUMP.IMG. Once it is copied to the specified drive on the other server, this file can be renamed, compressed, copied to diskettes, backed up to tape, or sent via FTP to novell.com (see the instructions under "Sending the Image File to Novell").

This method can speed up the memory image copying process by as much as four to five times over the other outlined methods. By way of comparison, for a server with 128 MB of RAM it can take 5 to 6 hours to copy the memory image file to diskettes.

The network drive method has been tested in Novell's server lab and has been used by several customers to obtain valid memory image files.

**Parallel Port Method.** This method requires the purchase of a drive that uses a parallel port as its method of connectivity. These drives require a device driver be loaded in the CONFIG.SYS file. Attach the drive, then boot the server. Because the drive is a DOS device and is ignored when NetWare loads, you can remove the drive once the server is started. As long as the REMOVE DOS command is not issued, DOS thinks the drive is still there.

When a server abends, reattach the drive onto the server's parallel port and dump the memory image to the device. The drive can then be moved onto a DOS PC, from which the image file can be compressed and sent to Novell.

The parallel port method was recommend by one of Novell's customers and beta sites as their current method of obtaining memory image files.

## Sending the Image File to Novell

To send a memory image file to Novell, an open support incident number is required. To obtain this number, work through your Novell Support Representative, or call 1-800-NETWARE and open a support incident. The incident is billable, and you will need to provide a credit card to open the incident. You will not be charged until the incident is resolved or closed. If it turns out that the problem is a NetWare bug and no patches were previously available, there will be no charge.

A Customer Support Representative will assign you a Technical Support Engineer who will help you analyze the memory image file. He or she will make arrangements to receive the image either in the mail or through the Internet.

Note: Before sending the image file, rename it with the first eight numbers of the incident number assigned to you. This will help Novell process the image file. Also, consult with your Technical Support Engineer to determine the best media format to use. Novell does not return floppy diskettes or backup tapes sent in with memory image files.

**By Mail.** To send the image file by mail, zip the file first and copy it to the agreed-upon media. Mail to:

Novell Support
Attention: *Technical Support Engineer's Name*
E-34-2
Novell Inc.
122 East 1700 South
Provo, Utah 84606
U.S.A.

**By Internet.** Customers with access to the Internet can send the memory image via anonymous FTP to novell.com. The file should be zipped first and then placed in the "incoming" directory. This method can save both parties time and money.

If you use this option, be sure to make arrangements with the Technical Support Engineer who will be receiving the file. Again, an open support incident is required. Files received on the Internet with no open support incident number will be deleted.