



# A Preventative Approach to Resolving Critical Server Issues

**Dirk A. D. Smith**

President

Alexander LAN, Inc.

**Allan P. Hurst**

Principal

KIS Computer Center

# Who are we?

## Allan Hurst / KIS Computer Center

Helps his clients by fixing weird (often catastrophic) problems on enterprise and corporate networks.

Favorite Quote: *"It's never done THAT before!"*

## Dirk Smith / Alexander LAN, Inc.

Creator of the Alexander SPK which automates recovery and diagnostics of system crashes.

Favorite Quote: *"My PC has Blue Screened 134 times. Is that normal?"*

# Who are you?

- This session is designed for network administrators who are responsible for installing, maintaining, and troubleshooting servers...because if you don't prevent this stuff from happening:



```
*** STDP: 0x0000000A (0xFFB0F004,0x00000010,0x00000000,0x00079D3)
IRQL_NOT_LESS_OR_EQUAL*** Address 800079D3 has base at 80007000 - SCSIPTORT.SYS

CPUID:GenuineIntel 5.2.4 irql:1F DPC SYSVER 0xF0000565

d11 Base DateStmp - Name
80010000 36224Cda - ntoskrnl.exe
80001000 35ca19db - smc810.sys
8027a000 353e319e - Disk.sys
80390000 36238303 - Ntfs.sys
f8eb8000 353e319c - Cdrom.sys
f9074000 35eb144b - KSecDD.sys
f8ee8000 353e3184 - 18042prt.sys
f9084000 31ec6c94 - Kbdclass.sys
f8c10000 353e3155 - S3.sys
f8f30000 353e31df - Msfs.sys
fe4bb000 362043ba - NDIS.sys
a0000000 36248f4f - Win32k.sys
f8c70000 353e3626 - CdFs.sys
f9132000 35fe29fd - Rasacd.sys
fe3ea000 36243C12 - tcpip.sys
fe49f000 35e5c7f3 - Rasarp.sys
fe493000 31ec6e15 - e1ink3.sys
fe3bb000 3610249c - afd.sys
fe47e000 31ec6c9b - Parport.sys
f913c000 31ec6c9d - ParVdm.sys
fe352000 35f03aa8 - rdr.sys
fe2de000 353e362c - mup.sys

d11 Base DateStmp - Name
80010000 35e72341 - hal.d11
80007000 35e5c313 - SCSIPTORT.SYS
8038c000 36269e3f - CLASS2.SYS
f8ea8000 31ec6c8d - Floppy.sys
f9214000 00000000 - Null.sys
f9215000 00000000 - Beep.sys
f907c000 353e318a - mouclass.sys
f8f00000 35646e19 - VIOEDPTT.SYS
f9094000 360ea154 - vga.sys
f8c30000 353e31d5 - Npfs.sys
f909c000 35fe17b4 - ndistapi.sys
fe482000 35d9fd5f - S3.d11
fe437000 35dde3d6 - Fastfat.sys
fe4a7000 31ec6e6c - TDI.sys
fe3cc000 36129a8d - netbt.sys
f8cfd000 3548ba13 - AsyncMac.sys
f8d00000 35fe1816 - ndiswan.sys
f8fbf000 353e35d4 - netbios.sys
fe476000 353e318f - Parallel.sys
f8d40000 35ef29c4 - Serial.sys
fe2ef000 35b7f615 - srv.sys

Address Word dump Build [1381]
801499f8 800079d3 800079d3 805f58c8 8047b000 805f5a06 805f1400 - SCSIPTORT.SYS
80149a0c 80149a3c 80140a3c 800079bf 805f58c8 8047b000 805f5a06 - ntoskrnl.exe
80149a10 800079bf 800079bf 805f58c8 8047b000 805f5a06 805f1400 - SCSIPTORT.SYS
80149a3c 80149a84 80149a84 00000000 800079d3 00000000 00010296 - ntoskrnl.exe
80149a64 800079d3 800079d3 00000000 00010296 00000000 805f1510 - SCSIPTORT.SYS
80149a80 80149a88 80149a88 80149a80 800039b6 805f5a48 fe26dd3c - ntoskrnl.exe
80149a84 80149a80 80149a80 800039b6 805f5a48 fe26dd3c 805732c8 - ntoskrnl.exe
80149a88 800039b6 800039b6 805f5a48 fe26dd3c 805732c8 80149a88 - smc810.sys
80149a96 80149a88 80149a88 805f5a48 fe26dd3c 805f14fd fe26dd3c - ntoskrnl.exe
80149ac0 80149a88 80149a88 8000442b 805f5a48 805f14fd fe26dd3c - ntoskrnl.exe
80149ac4 8000442b 8000442b 805f5a48 805f14c0 805f14c0 fe26dd3c - smc810.sys
80149afc 800023fd 800023fd fe26dd3c 805f5a48 fe26dd3c 805eed08 - smc810.sys
80149b14 800084eb 800084eb 805f5a48 fe26dd3c 80149b44 fe26dd4c - SCSIPTORT.SYS

Beginning dump of physical memory
Restart and set the recovery options in the system control panel
or the BIOS/UEFI system start option. If this message reappears,
contact your system administrator or technical support group.
```

# “You’ll have to update one of these...”



## THOMAS C. BURRELL, MCSE, CNE, CCNA

5687 Brookstone Drive  
Chicago, IL 60601  
Phone: (555) 786-8083 • Email: tom@emailaddress.com

### INFORMATION SYSTEMS DIRECTOR / NETWORK SERVICES MANAGER Proven Technical & Management Expertise in a Career Spanning 15+ Years

Technically sophisticated and business-savvy management professional with a pioneering career reflecting strong leadership qualifications coupled with “hands-on” IS and networking expertise. Maintain focus on achieving bottom-line results while formulating and implementing advanced technology and business solutions to meet a diversity of needs. Superior record of delivering simultaneous large-scale, mission-critical projects on time and under budget. Team-based management style and excellent interpersonal/communication skills.

*IT Strategic Planning / Business Solutions / Team Leadership / Budgeting / Project Management  
Capital Expenditure Planning / Contract Negotiations / Vendor Relations*

#### Professional Experience

INFORMATION SYSTEMS MANAGER, Clinic Health System, Chicago, IL 1993 - Present

Recruited to upgrade and replace obsolete technologies at this world-class health care organization with more than 2000 users in 15 remote locations. Hire, train, develop, and lead a 20-person technical team. Manage a \$2 million capital budget and \$1.2 million operating budget. Scope of position is expansive and includes departmental direction and full design, installation, engineering, implementation, support, training, administration, and management authority for:

- LAN/WAN Network Services
- 24x7 Data Center Computer Operations
- Applications Systems
- Web/Internet Design & Operations
- PC Desktop Systems
- UNIX Systems Administration
- Database Administration
- Help Desk Operations

Spearheaded transition from outdated organization-wide and departmental technologies to highly functional, streamlined, and cost-effective client-server technologies and business solutions that have dramatically improved efficiency, decreased expenses, and optimized data integrity and safety.

#### Key Projects & Achievements:

- Directed design and installation of the complete \$8 million LAN/WAN infrastructure. Utilized state-of-the-art technologies to provide network connectivity of disparate Mainframe, AS400, UNIX, Windows NT, Novell, and PC systems.
- Completed, in just 8 months -- 22 months ahead of schedule -- a complex \$15 million project forecasted to take 2.5 years and involving replacement of more than 30 systems.
- Delivered \$2 million in cost savings through aggressive negotiation of contracts and pricing on a budgeted \$10 million for hardware/software purchases and consulting services.
- Saved more than \$1.2 million in technical consulting fees by negotiating complimentary network design services from vendors.
- Performed the work of 3 full-time equivalents, slashing labor expenses substantially by expanding personal responsibility to include UNIX, network, and database administration.
- Decreased inventory, application pricing, and licensing expenses \$750K by establishing standardization for applications, PC desktops, and networking systems.
- Defused and resolved long-standing conflicts and department problems; elevated morale and decreased high employee turnover rates, achieving the best retention rate in the company.





# Why are we all here today?

- As system administrators, we need to know:
  1. How to prevent crashes

*and since they're gonna crash anyway...*

## 2. How to diagnose crashes

(If you're reading this text, you're scaring us. We can't even read this small on the laptop screen.)

```
Abend on P00: Page Fault Processor Exception (Error code 00000000)
OS version: Novell NetWare 5.60 August 18, 2001

...Debug symbols are enabled!
Running Process: ABEND.NLM      1 Process
Stack: 00 00 00 00 80 54 B0 D3 7C D1 AC D3 00 97 B0 D3
      00 00 00 00 E8 03 00 00 00 00 00 00 00 04 04
      31 00 00 00 00 97 B0 D3 00 00 00 00 00 00 00
Additional Information:
The CPU encountered a problem executing code in LIBC.NLM. The problem may be
in that module or in data passed to that module by a process owned by
ABEND.NLM.

Press:
"S" to suspend the running process and update the ABEND.LOG file.
"Y" to copy diagnostic image to disk (COREDUMP).
"X" to update ABEND.LOG and then exit.

Writing diagnostic core dump to: C:\COREDUMP.IMG
(Press ESC to cancel)

Writing page 165 of 261989 (normal: 90, phantom: 75)
```

Interlude: “A funny thing happened on the way to the conference...”



quakecon 2002 - mesquite, texas, USA - august 2002 - photo: yossarian holmberg (yossman@yossman.net)

# Crashes Cost Money.

- What does a crash really cost your company?
- The per-hour cost of downtime is a lot higher than you may think.
- Consider a small company of 100 people being paid an average of \$30,000/year. That's  $100 \times \$15/\text{hour} = \$1,500/\text{hour}$  downtime cost! (And that's just salary, not including payroll taxes or the cost of lost business.)
  - *Many companies lose millions of \$ per hour in lost business...*

# “Stuff Happens.”

The best way to handle disasters? Prevent them from happening in the first place!

Know how to recover from what you can't prevent.

Generally speaking, these techniques apply to Windows, NetWare, Unix, and Linux systems.

*Prevention is always cheaper than recovery.*





# General Categories of server crashes

## Soft Crashes

When the OS can be prevented from needing to crash just by suspending a process and/or module.



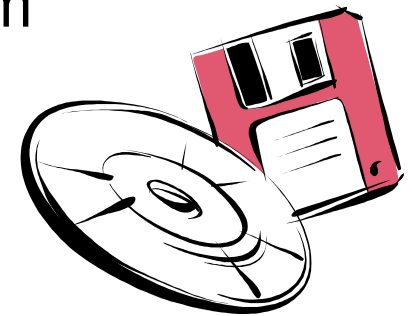
- ## Hard Crashes

When nothing can prevent the system from crashing and it will have to be fully restarted to regain its services



# What are the major causes of server crashes?

- Software, Software, Software. 95% of system crashes are software generated!
- Most crashes are repeat crashes.
  - This applies to servers *and* PCs
- The OS is rarely at fault (Honest!) It is usually a module from a third-party vendor.
- If the OS really looks guilty there's a strong possibility that a third-party module passed a bad instruction to the OS.



# Could it be a Hardware Crash?

- If your server experiences a hardware crash, run any diagnostic programs your hardware vendors provides to help locate the source of the problem (ex: memtest86).
- Try swapping hardware components to see whether the problem disappears when a particular component is replaced. Swap components in this order:

1. Memory



2. Power supply



3. System board



# Crashes can be very public...



# Prevention 101:

## Use “decent” hardware

Brand Names Count. (Yes, really.)

- Avoid “death clones” for production machines.
- Clone components change too rapidly to find again.  
(brand-name components are usually stocked for several years)
- Clone servers are certified only at the component level.
- Avoid servers that are glorified workstations.  
(Vendors: You know who you are. Stop it!)
- Use Microsoft/Novell certified platforms only. (Please.)
- Examples of “Brand Name” servers that have worked for us:  
[HP/Compaq...](#)
- REAL servers are ...
  - Built entirely from components intended for 7x24 use for 3 years or more
  - Optimized for high performance/throughput as a single machine
  - Certified as a cohesive unit, NOT as individual components
- Factory-Built or Assemble It Yourself Onsite?
  - Factory-built servers still require a systems check.
  - “DIY” servers take more time, but you’re certain of the result.





# Prevention 101: Use “decent” hardware

Add enough memory!

- Allan’s RAM Rule #1:

“If a server will use Java, start at 1GB”

- Allan’s RAM Rule #2:

“If a server will use Windows 2000/2003 or NetWare 6/6.5, start it at 2GB.”

- Allan’s RAM Rule #3:

“There is No Such Thing as ‘Too Much’ Server Memory.”

(The old, “too much memory” rule last applied to NetWare 2.x!)

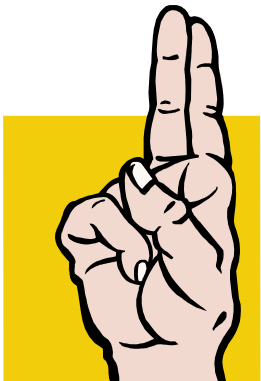
***Besides, RAM is cheap these days!***



# Prevention 101:

## Use “decent” hardware

- Caveat Emptor: Always buy the manufacturer’s extended warranty...it’s not a bogus add-on.
  - “Out of the box” warranties promise on-site service in one business day, with a dangerous caveat: “...or best effort”
  - The average difference between an out of box warranty call and a 7x24x4 hour response call? 3 to 6 business days!
  - Most manufacturers maintain separate stocks of spare parts for “contract” and “non-contract” customers. (The response difference can be measured in days!)
  - Buy a 7x24x4 hour response onsite warranty that will last at least 3 years. It’s cheaper than having a server down with bad hardware for one or more full business day(s).
  - Most warranty uplifts cover only components **INSIDE** the server box. If you have an external tape drive or storage array, it will need its own extended warranty



# Prevention 102:

## Build 'em right the first time.

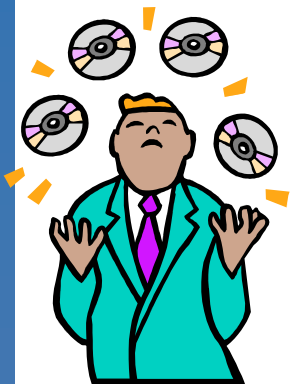
- Don't use third-party components unless you're adding functionality the server manufacturer can't provide. (E.G., HBA for a SAN, special NIC, etc.)
- After building the server, upgrade firmware on ALL components before loading any software.
- Configure and burn in server hardware at least 24 hours before you start loading software.
- Download the latest drivers (NIC, disk, tape, etc.) to floppy disk or CD-R before loading the OS.



# Prevention 102:

## Build 'em right the first time.

- Many people have been bitten on new installs and upgrades by early “press versions” of OS CDs.
- “Pre-patched” NetWare CD-ROM images are downloadable from <http://support.novell.com> Use these to configure servers whenever possible. Less time, less work.
- If you can’t install from a pre-patched CD, have the latest OS patch nearby on CD before you start installation.
- Don’t forget to create CDs with patches or updates for other components you may need, such as newer versions of eDirectory™, iManager™, or eGuide™.
- Copy updated and third-party NIC, disk, and other drivers to C:\NWUPDATE before starting the NetWare installation or upgrade, and NetWare will “find and grab” the updated drive automatically.



# Prevention 102:

## Build 'em right the first time.

### Use a UPS with auto-shutdown software



- You spent HOW much on the server, and you can't bother to protect it from a 2-second brownout?
- Either use a small UPS for each server, or one large UPS for several ... but each server has to have a connection back to the UPS and have auto-shutdown software loaded.

### Configure the OS with as few features as needed



- Try to NOT load every feature of the OS unless you need it
- Get the base OS working first
- Add bells and whistles later

### Document the final configuration

- When it's working just as you want it, use config.nlm (load config /ds) to write a config.txt file, and save it somewhere not on the server.





# Interlude: Commercial break (Survey & Giveaway)

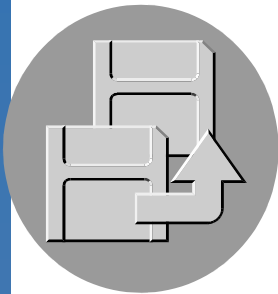


# Prevention 103:

## Plan for recovery

Back up the DOS boot partition. This can cut as much as an hour from your disaster recovery time.

- Use a parallel port ZIP drive (if your server still has a parallel port)
- Load DOSFAT.NSS (NW6) and copy down to a writable CD.
- Update your DOS partition backup every time you patch the server!
- If you're running mirrored drives, don't forget to manually synchronize the DOS boot partitions on each drive.
- Copy the server's config.txt file to the same place as your DOS boot partition backup –or- keep it handy on floppy (or printed out in your network documentation binder). You'll be glad you have it when you have to rebuild the server from scratch, and don't recall what sizes each volume was or what name spaces were loaded!



# Prevention 103: Plan for recovery

Get your “recovery kit” ready before deploying server!

- If you configured your server manually ... have a boot floppy handy which contains the same version of DOS (MS-DOS/DR-DOS) as was used to create the server. (If you installed by booting from a NetWare OS CD or a NetWare license disk, have a copy of that in the kit.)
- A floppy with the correct CD-ROM and ZIP drivers on it. (Could be the same as your boot floppy.)
- A ZIP cartridge or CD-R containing a backup of the server boot partition.
- Your NetWare OS CD (pre-patched if that's what you used originally, or pre-patched to the current level).
- Copies of your NetWare license diskette(s)
- A CD-ROM containing the OS patches matching the patch level of the DOS boot partition (if not using a pre-patched OS CD)

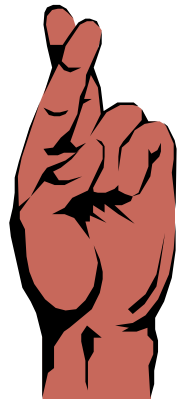


# Prevention 103:

## Plan for recovery

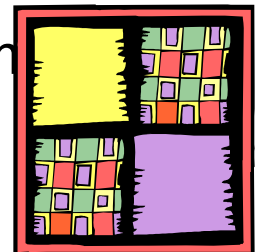
Other stuff to have in your recovery kit:

- A copy of your tape backup/restore software (if applicable).
- Your tape software license diskette or serial number needed for installation.
- A copy (printed or on floppy or on the ZIP drive!) of the most recent config.txt file.
- A checklist that works for you – especially at 3:00 AM with people yelling at you.
- Pack your kit-bag: Put it ALL in a box/bag so that you look in ONE place, find it all, and can carry it to another room or site immediately.



# Prevention 201: Patch that puppy!

- Microsoft and Novell don't issue new OS patches simply to torture us.
- Current support packs usually fix a lot more than they break
- It pays to keep current: if you're not patched to the current level of support pack, don't bother calling Support. Both Novell & Microsoft will tell you to apply the latest support pack and call back when you've reproduced the problem
- Edirectory needs patches, too! Go to:  
<http://support.novell.com/filefinder/>, and search for "ds.nlm"





# Prevention 202: Document & Maintain

Investigate the wonders of the cron utility.  
Here are some tasks to consider automating:

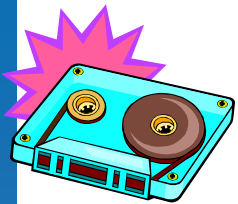


- Run “**config.nlm /d**” automatically each night, prior to backup.
- Run “**dsrepair -rc**” on each server containing DS replicas each night, prior to your tape backup routine, to create backup dib sets. (This will make Novell DS Support very happy with you when you need to call them.)
- **Toolbox.nlm** is a terrific way to purge volumes after the backup runs ... especially when the volume hosts print queues or applications that create and delete a million little stinkin’ “lock” files. (Remember Paradox?)
- Chronic DS problems? Consider running unattended dsrepair operations each night.
- Document the heck out of the autoexec.ncf and startup.ncf files. (This is automatically inserted into your config.txt file.)



# Prevention 203:

“Just back the darn thing up, willya!?”



- We don't care what the tape software vendors say, nothing beats “everything, every night.”
- Many current applications use temporary “journal” files. Overlaying multiple incremental restore sets can honk up applications -- especially critical vertical market apps such as accounting, HR, practice management, etcetera.
- If you can't back everything up overnight, you need to figure out why.
- For most shops, the existence of cheap gigabit switches and NICs and large, fast tape drives means that unless you're running a very large disk array, you *should* be able to back up everything every night.

# Prevention 203:

“Just back the darn thing up, willya!?”

- Change the tapes out regularly. One year's service, max. (They stretch, get dirty, and drop bits all over the floor!)
- Use a simple rotation that you can easily understand at 3 o'clock in the morning when you're trying to restore during an emergency.
- Cleaning tapes are good things. Listen to your tape drive. Clean it when the little light flashes. You and your tape drive will both be happier with the result.

# Interlude: Roadside sights.



# Troubleshooting 101:

So...what do you do when all heck breaks loose?

- Grab a blank pad of paper and pen ... you're going to need it, both for yourself and in case you need to call Microsoft or Novell support.
- Keep a chronology. As you go through each of the below steps, note the current time (and date, if needed). It really will help you track the situation better.

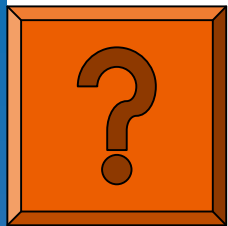




# Troubleshooting 101:

Determine the nature and scope of the situation.

(Write all of this down on your blank pad, please)

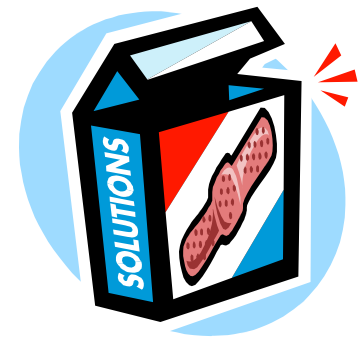


- *What* are the symptoms?
- *Who* is affected by the problem?  
(Users, groups, buildings, campuses, etc.)
- *Which systems* are affected?
- *When* did this start? Today? Last night? Last week? (If the problem has been going on for a while, why is it just now being reported?)
- *What was changed* before the problem started?
- Politely inform people that this is NOT a time to “cover their tracks”. Total truth is essential during network emergencies.

# Troubleshooting 101:

## Determine Possible Causes.

- Hardware (e.g., bad memory)
- Software (e.g., misbehaved driver)
- Infrastructure (network, power, WAN, internet, flooding)
- Configuration of any of the above items
- Operational/Procedural Error  
(our old friends “Fred” and “Simon”)



# Troubleshooting 101:

## Isolate the variables.

### Hardware

- component swap
- firmware update
- configuration change

### Software

- component update/upgrade
- driver upgrade
- configuration change

### Procedural

- if you change the order of operations leading up to the problem, does it still happen?
- can you change the entry point?
  - Workstation(s) used
  - Login ID(s) used



# Troubleshooting 101:

Start to make changes...carefully.



**WRITE IT DOWN** - At each step, record what you've done, and what happened. It's far too easy to lose track of where you are if you don't write it down. (If you have a second person in the room, have them scribe for you while you're knee-deep in network blood and guts.)

**CHANGE ONLY ONE VARIABLE** at a time!

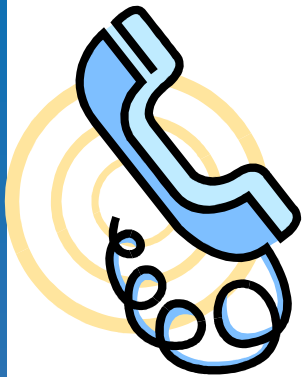
**CHANGE EACH VARIABLE BACK** to the original before trying a new variable. (There are exceptions, such as when changing a variable improves the problem.)

**TEST THE SOLUTION** - Using your notes of what the original problem was, try to replicate the problem

# Troubleshooting 102:

When it's time to call Microsoft or Novell Support ...

- Please have all of this information ready:



- Support PIN ...and password
- OS version and patch level
- DS version and patch level
- List of third party products (version and patch level!) running
- A copy of minidump or config.txt lists all vital server information ... including system module dates and sizes.
- A concise description of the problem
- Your chronological notes...so support knows the exact sequence of events
- A workstation, logged in as admin or equivalent
- An abend.log file or an SPK Crash Report



- Remote access methods to consider (if needed):

- PCAnywhere via dial-up or TCP/IP
- Control-F1, Desktop Streaming, or similar web-based technologies

# Interlude: Abends Without Borders.

```
(c) Copyright 2000-2003 Computer Associates. All Rights Reserved

PFC: NLMVersionInformation OK
PFC: Information is saved to file SYS:¥ARCSERVE¥NLM¥PFC.LOG
PFC.NLM Unloaded
モジュール CATIRPC.NLM をロード中
  CA RPC Interface (Build 218.000 11/20/00)
  バージョン 7.00 2000 -11- 20
  (C) Copyright 1991-2003 Computer Associates. All Rights Reserved.
モジュール ASDB.NLM をロード中
  ARCServe 7.0 Database (Build 218.000 11/29/00)
  バージョン 7.00 2000 -11- 29
  (C) Copyright 1990-2003 Computer Associates. All Rights Reserved
モジュール ARCSERVE.NLM をロード中
  ARCServe 7.0 Scheduler (Build 218.000 11/22/00)
  バージョン 7.00 2000 -11- 22
  (C) Copyright 1990-2003 Computer Associates. All Rights Reserved.

E0129 Failed to create console screen, 稼働中のプロセスが停止されます

2002/06/26 10:36:45 : SERVER-5.0-4631 [nmID=1001C]
  警告! サーバ NW51-01 に致命的なエラーが発生しました。処理中のプロセスが一時停止ま
  たは回復しました。このサーバが別のサービスに影響を与える恐れがあります。

NW51-01 <1>: _
```

# Troubleshoot Server Crashes

- If you follow Allan's advice, you **WILL** prevent system failures.
- The problem is that, like a lot of things in life, even if you do all the right things...**[STUFF]** happens and systems crash  
*which is a good way to ruin the day if you're the one to call*





# Troubleshoot Server Crashes

- Many crashes take 6 – 8 weeks to resolve during which time the server will likely fall over another dozen times
- Now your bad day becomes a stress-ridden couple of months
- This scenario is common and completely unnecessary
- ***Well, perhaps 1/2 of the time***
- This is because 50% of the crashes that you see, server or desktop, can be solved, by you, in less than a minute with a tool that costs nothing
- Sound too good to be true?

***It would be if it worked 100% of the time***

# Troubleshoot Server Crashes

- We're talking about the debugger  
Familiarity with the debugger and data raises the bar:
  - increases your abilities to resolve critical issues
  - decreases your need for outside support
  - increases your abilities to help them resolve issues for you when you do call on them.

# Troubleshoot Server Crashes

- *A 30 Second Answer*  
I looked at a dump file. Here is the result.
- Crash Cause:           A driver  
         Driver Name:        vdriver.dll
- The entire process took me about 30 seconds
- This is what we'll spend the next hour preparing for

# Troubleshoot Server Crashes

***SO, IT'S NOT THAT BAD***

***...well...usually anyway!***

- The most laborious effort you will need to face is...  
*listening to me for an hour*
- ***In minutes*** you can set up a system to debug memory dumps
- ***In seconds*** you can find the cause of more than half the crash events

# Troubleshoot Server Crashes

- Unlike Allan who is remarkable at a wide range of system issues, I specialize in crashes...
  - *preventing and resolving, not causing*
- This session is the result of an article that I have written and your comments in our survey card will help me make improvements and final edits
- SURVEY CARD
- Give away: Alexander SPK Windows Enterprise Edition

# Troubleshoot Server Crashes

- No network administration course teaches about crash management yet you are responsible for it!
- It is a vicious circle: It is what at Alexander LAN we call the



## Crash Data Dilemma

*The tools and data needed to resolve system crashes are Greek to the people who need them the most*

# About Debuggers

- A debugger is a low level tool that enables you to look at the details of the state of a system at a point in time. According to the Microsoft online glossary:

“The origin of this definition is in some dispute, computer folklore attributes the first use of bug in this sense to a problem in the Harvard Mark I or the Army/University of Pennsylvania ENIAC that was traced to a moth caught between the contacts of a relay in the machine (although a moth is not entomologically a bug.)”

<http://support.microsoft.com/default.aspx?scid=/support/glossary/B.asp>



# About Debuggers

- There are two basic kinds of debuggers
  - Application Level Debuggers
    - For troubleshooting user mode programs like MS Word or any 3<sup>rd</sup> party application
  - Kernel Level Debuggers
    - For working with kernel (system) level programs like the operating system itself or drivers

# About Debuggers

- Difference between an Application Crash and System Crash
  - Application Crash
  - System Crash



# About Debuggers

- Today we will be
  - working with Windows  
*The principals here also apply to NetWare, Unix, and Linux*
  - use WinDbg/KD which is a free download from Microsoft
  - troubleshoot system (as opposed to application) crashes

# Set Up the Debugger

- System Requirements
  - OS Version
    - Windows Server 2003/2000/NT4
    - Windows Workstation XP/2000
  - Space
    - Windows: About 25 MD hard disk space

**NOTE:** Remember that this does not including dump files!

# Set Up the Debugger

- Download and Install

<http://www.microsoft.com/whdc/ddk/debugging/installx86.msp>

One debugger for Windows. Continuously updated.

# Set Up the Debugger

- Symbol Table Files
  - Before using WinDbg you MUST ensure it has access to the Symbol Table Files
  - What are symbol files?
  - Symbol files are not included
  - Example of a Windows Symbol: MmAccessFault
  - Use the **CORRECT SYMBOLS**! Using wrong symbols is like using the wrong map...
  - A very cool guy at MS set up a Symbol Table Server

# Set Up the Debugger

- You need a memory dump
  - You may
    - Have one
    - Find one
    - Make one (NOT recommended on Windows systems)
    - Download one from:

<http://www.alexander.com/Download/SampleDump.zip>



# Set Up the Debugger

- The Memory Dump
  - Dump Size
    - Windows 2000/XP/Server 2003 can produce three sizes of memory dumps
      - Small/mini dump (64K)
      - Kernel memory dump (10-33% RAM)
      - Complete/full memory dump (Size of RAM)

# Set Up the Debugger

- Memory Dump
  - General points about dump files
    - Allow loads of hard drive space, ESPECIALLY FOR FULL DUMPS!
    - If you plan to save files for later...
    - With multiple events and multiple systems it can be confusing so consider naming conventions to help
    - NT4 only does Full Dumps

# Set Up the Debugger

- The Memory Dump
  - General points about dump files
    - Allow loads of hard drive space, ESPECIALLY FOR FULL DUMPS!
    - If you plan to save files for later...
    - With multiple events and multiple systems it can be confusing so consider naming conventions to help
    - NT4 only does Full Dumps

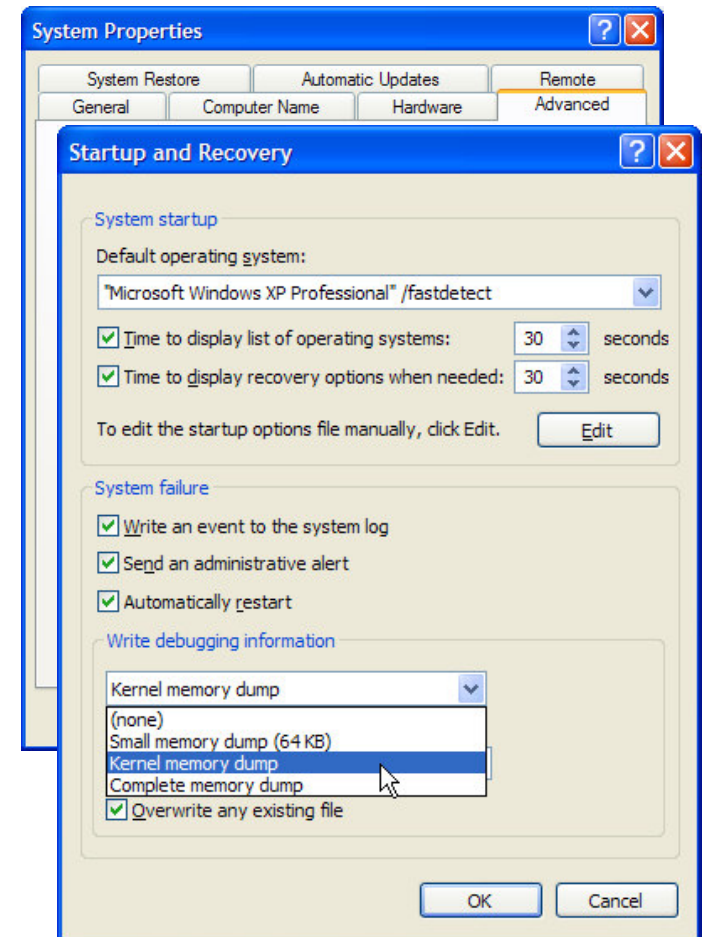
Naming Windows Memory Dump Files	
ORIGINAL:	MEMORY.DMP
SYSTEM:	MailServer03.DMP
CRASH DATE:	20030817.DMP
COMPANY:	WorldWideBearings.DMP <i>(when sending to vendor for remote analysis)</i>

# Set Up the Debugger

- The Memory Dump

## *Windows*

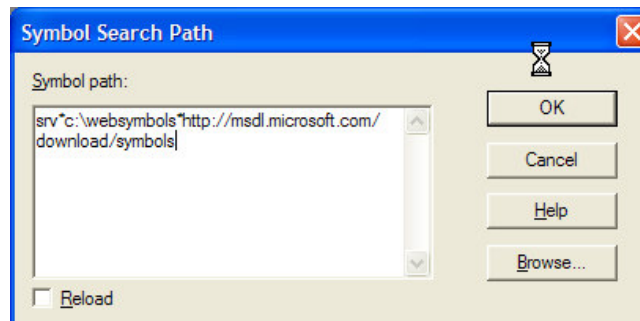
- Set System to Save Dumps
- Other considerations



# Set Up the Debugger

- Launch the Debugger
  - Start *WinDbg*
- Set Symbol File Path. Enter:

`SRV*c:\local cache*http://msdl.microsoft.com/download/symbols`



- Space Needed for symbol files:
  - 5MB Single system, more for numerous
- On my PC:  
`SRV*c:\websymbols*http://msdl.microsoft.com/download/symbols`

# Set Up the Debugger

- Set Executable Path. Enter:

System Root %System32; %SystemRoot/System32/Drivers;  
<http://www.alexander.com/SymServe>

- 

On my XP box, this is my path:

c:\windows\System32; c:\windows\system\System32;  
<http://www.alexander.com/SymServe>

- NOTE: Even if the debugger finds the drivers on your c: drive they may be incorrect because they may have been changed by Windows Updates!

# Set Up the Debugger

- Unexpected value of minidump files:

## ***HISTORY?***

- Yup. Windows XP, by default, saves a Minidump file for EVERY crash the system has ever had (unlike kernel and full dumps)
- On typical systems there will be a handful of minidumps
  - Some have 100s!
- Since BSODs have long been awkward for most people to troubleshoot, the drivers that caused the crashes are often still in use
- Note XP will also save the most recent full dump as well



# Set Up the Debugger

- Open Dump File
  - Open: *File/Open Crash Dump*
  - Kernel and Full Memory Dumps
    - - Always named MEMORY.DMP by the system but you can open any renamed dump files too
    - - Always in the same folder when saved by the system

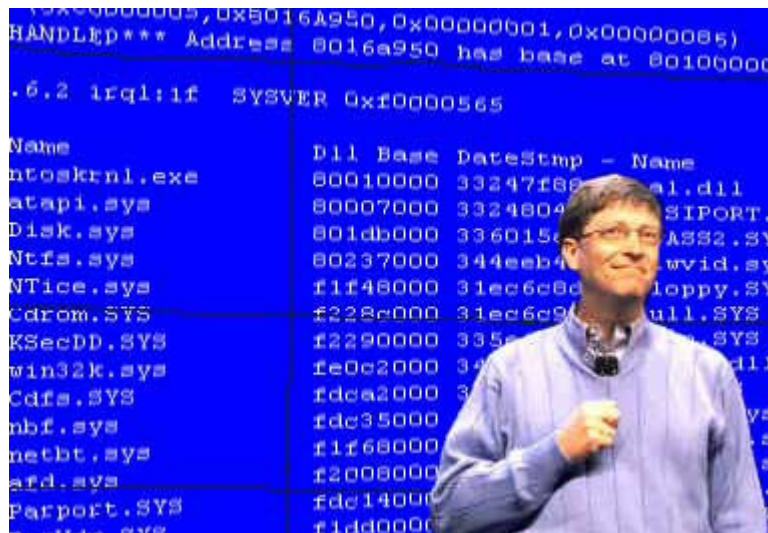
# Set Up the Debugger

- Open Dump File
  - Upon opening:
    - If they are incorrect or not available, you will see a message like this:  
  
\*\*\* ERROR: Symbol file could not be found. Defaulted to export symbols for ntoskrnl.exe
    - If you see a message
      - *Only kernel address space available*
  - Save Base Workspace Information

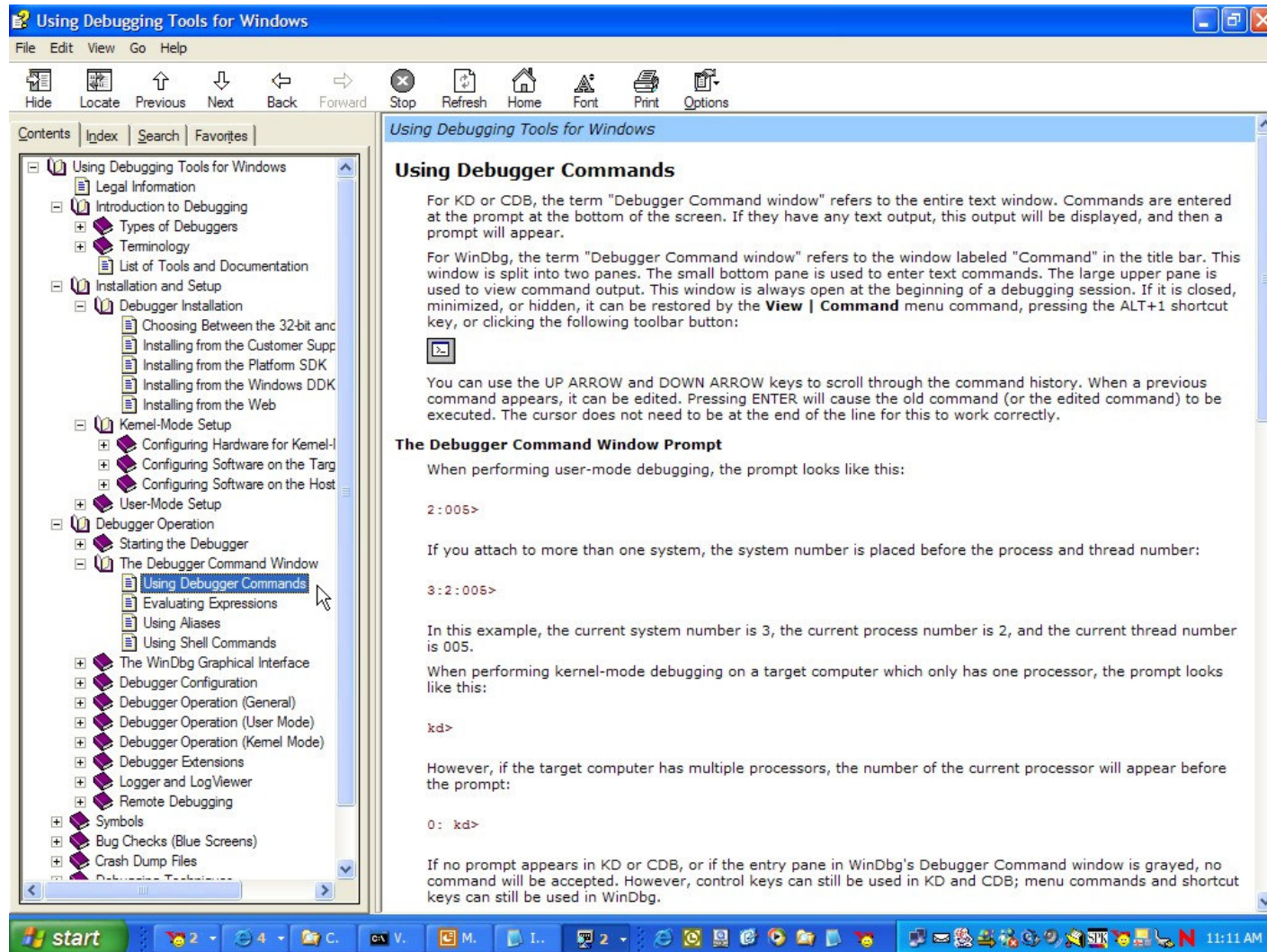
# Debugger Commands

# Debugger Commands

- With the debugger(s) set up, let's now look at what commands are needed.



# Debugger Commands

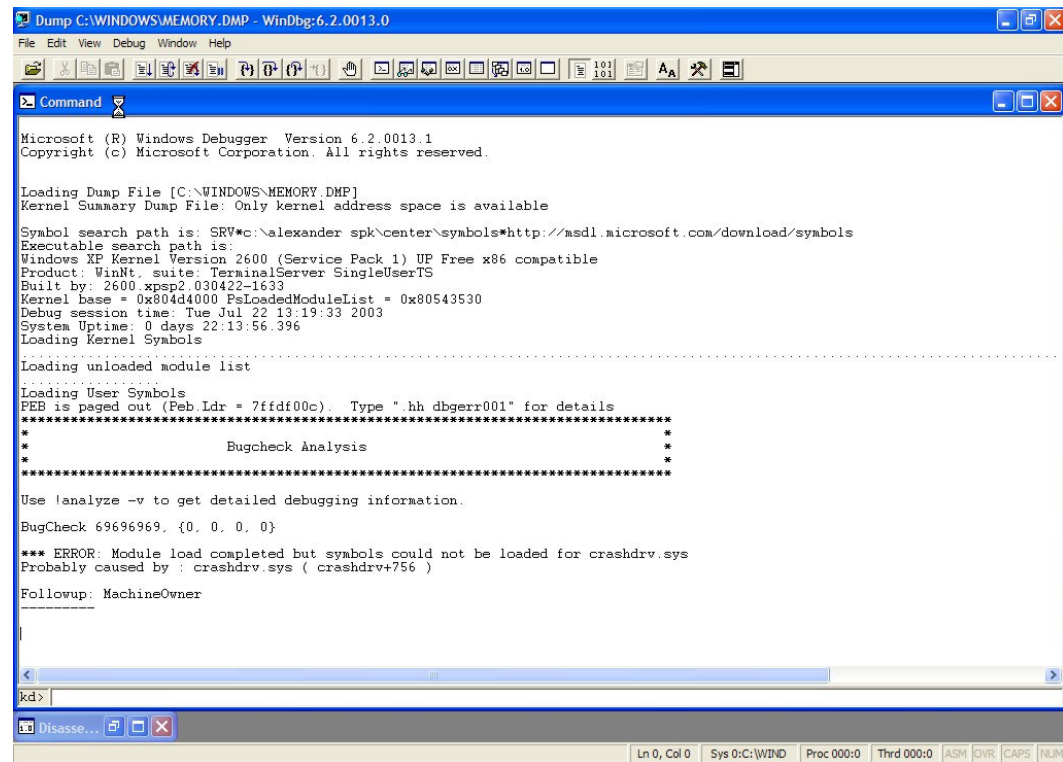


# Debugger Commands

- With the dump file successfully opened, the debugger runs a preliminary analysis
  - It may report

\*\*\* ERROR: Module load completed but symbols could not be loaded for [some driver]

No worries



```
Dump C:\WINDOWS\MEMORY.DMP - WinDbg:6.2.0013.0
File Edit View Debug Window Help

Command

Microsoft (R) Windows Debugger Version 6.2.0013.1
Copyright (c) Microsoft Corporation. All rights reserved.

Loading Dump File [C:\WINDOWS\MEMORY.DMP]
Kernel Summary Dump File: Only kernel address space is available

Symbol search path is: SRV*c:\alexander spk\center\symbols*http://msdl.microsoft.com/download/symbols
Executable search path is:
Windows XP Kernel Version 2600 (Service Pack 1) UP Free x86 compatible
Product: WinNt, suite: TerminalServer SingleUserTS
Built by: 2600 xpp2.030422-1633
Kernel base = 0x804d4000 PsLoadedModuleList = 0x80543530
Debug session time: Tue Jul 22 13:19:33 2003
System Uptime: 0 days 22:13:56.396
Loading Kernel Symbols
Loading unloaded module list
Loading User Symbols
PEB is paged out (Peb.Ldr = 7ffdf00c). Type ".hh dbgerr001" for details
*****
*                               *
*                               *
*                               *
*                               *
*****

Use !analyze -v to get detailed debugging information.

BugCheck 69696969, {0, 0, 0, 0}

*** ERROR: Module load completed but symbols could not be loaded for crashdrv.sys
Probably caused by : crashdrv.sys ( crashdrv+756 )

Followup: MachineOwner
*****

kd>
```

# Debugger Commands

- Commands

**!analyze -v** State of the system when it crashed, the fault encountered, and who is the primary suspect

**!drivers** List of all drivers loaded when the system crashed, along with summary information about their memory use

**!mv** Driver path, version, vendor (if they were thorough), and description

# Debugger Commands

- If you want to sound like you know what you're talking about, this is how to say it:

`!analyze -v`

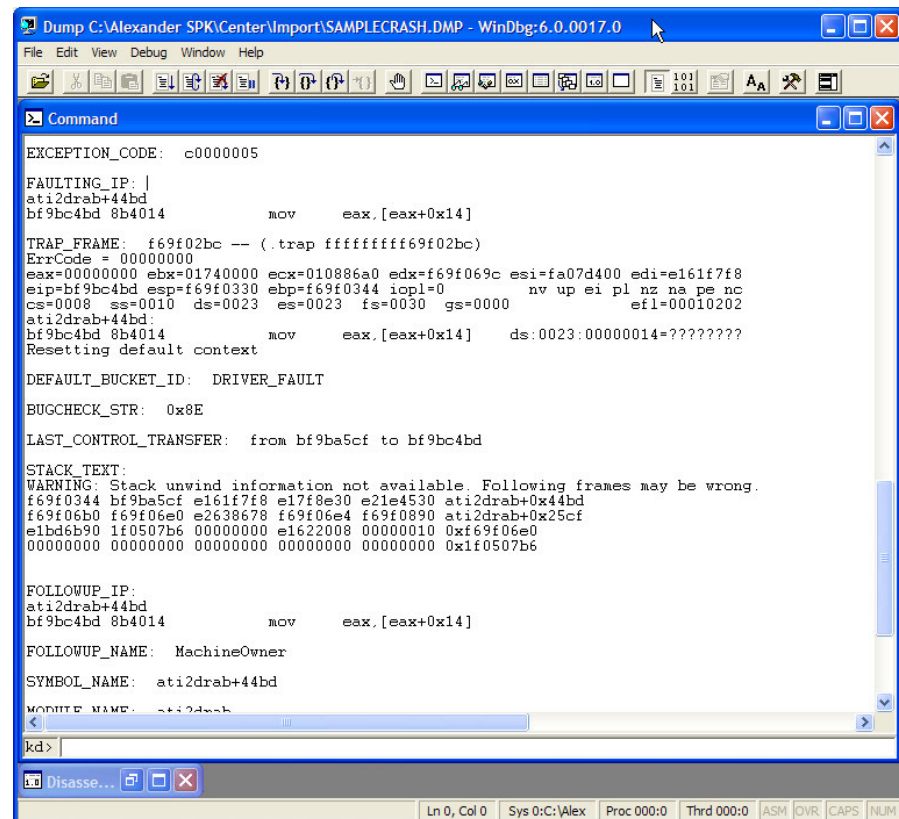
*bang analyze dash vee*

*“v” = verbose or “show me the detail”*



# Debugger Commands

- Output from *!analyze -v*
  - *So much info, you may not need more*



The screenshot shows the WinDbg interface with the Command window displaying the output of the `!analyze -v` command. The output provides detailed information about a system crash, including the exception code, faulting IP, trap frame, error code, register values, bugcheck string, and stack text. The stack text indicates a warning that stack unwind information is not available for the following frames.

```
Dump C:\Alexander SPK\Center\Import\SAMPLECRASH.DMP - WinDbg:6.0.0017.0
File Edit View Debug Window Help
[Icons]
Command
EXCEPTION_CODE: c0000005

FAULTING_IP: |
ati2drab+44bd
bf9bc4bd 8b4014          mov     eax,[eax+0x14]

TRAP_FRAME: f69f02bc -- (.trap ffffffff69f02bc)
ErrCode = 00000000
eax=00000000 ebx=01740000 ecx=010886a0 edx=f69f069c esi=fa07d400 edi=e161f7f8
eip=bf9bc4bd esp=f69f0330 ebp=f69f0344 iopl=0         nv up ei pl zr na pe nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010202
ati2drab+44bd
bf9bc4bd 8b4014          mov     eax,[eax+0x14]  ds:0023:00000014=????????
Resetting default context

DEFAULT_BUCKET_ID: DRIVER_FAULT

BUGCHECK_STR:  0x8E

LAST_CONTROL_TRANSFER:  from bf9ba5cf to bf9bc4bd

STACK_TEXT:
WARNING: Stack unwind information not available. Following frames may be wrong.
f69f0344 bf9ba5cf e161f7f8 e17f8e30 e21e4530 ati2drab+0x44bd
f69f06b0 f69f06e0 e2638678 f69f06e4 f69f0890 ati2drab+0x25cf
e1bd6b90 1f0507b6 00000000 e1622008 00000010 0xf69f06e0
00000000 00000000 00000000 00000000 00000000 0x1f0507b6

FOLLOWUP_IP:
ati2drab+44bd
bf9bc4bd 8b4014          mov     eax,[eax+0x14]

FOLLOWUP_NAME:  MachineOwner

SYMBOL_NAME:  ati2drab+44bd

MODULE_NAME:  ati2drab

kd>
Disassembler [Icons]
```

# Debugger Commands

- kd> !analyze -v

**FAULTING\_IP:**



The *FAULTING IP* (Instruction Pointer) indicates an address where software screwed up

# Debugger Commands

- `kd> !analyze -v`

**FAULTING\_IP:**

**vdriver+44bd** ←

WinDbg says that this IP address is owned by a (fictitious) software product called VDriver

# Debugger Commands

- kd> !analyze -v

FAULTING\_IP:  
vdriver+44bd

**DEFAULT\_BUCKET\_ID:**



The *DEFAULT BUCKET ID* identifies the kind of fault that occurred

# Debugger Commands

- `kd> !analyze -v`

FAULTING\_IP:  
vdriver+44bd



**DEFAULT\_BUCKET\_ID: DRIVER\_FAULT** ← The problem was caused by a faulty driver

# Debugger Commands

```
kd> !analyze -v
```

```
FAULTING_IP:  
vdriver+44bd
```

```
DEFAULT_BUCKET_ID: DRIVER_FAULT
```

```
FOLLOWUP_IP:
```

**vdriver+44bd location** ←

Note that 44bd represents the offset from the beginning of this address where it crashed

# Debugger Commands

```
kd> !analyze -v
```

*[ Lines of debugger spew omitted ]*

FAULTING\_IP:

vdriver+44bd

*[ More spew tossed ]*

DEFAULT\_BUCKET\_ID: DRIVER\_FAULT

*[ Still more spew tossed ]*

FOLLOWUP\_IP:

vdriver+44bd location

*[ Yep. More tossed ]*

SYMBOL\_NAME: vdriver+44bd

**MODULE\_NAME:**



The name of the guy holding  
the smoking gun

# Debugger Commands

```
kd> !analyze -v
```

*[ Lines of debugger spew omitted ]*

FAULTING\_IP:

vdriver+44bd

*[ More spew tossed ]*

DEFAULT\_BUCKET\_ID: DRIVER\_FAULT

*[ Still more spew tossed ]*

FOLLOWUP\_IP:

vdriver+44bd location

*[ Yep. More tossed ]*

SYMBOL\_NAME: vdriver+44bd

**MODULE\_NAME: vdriver**



The smoking gun was  
VDriver



# Debugger Commands

kd> !analyze -v

STACK\_TEXT:

```
ecbecc1c f7e17756 69696969 f7e17410 04515f10 nt!KeBugCheck+0x10
ecbecc34 804ea221 852daf18 84bada80 806abfe0 VDriver+0xbd44
ecbecc44 8055d0fe 84badaf0 84e09538 84bada80 nt!IopfCallDriver+0x31
ecbecc58 8055de46 852daf18 84bada80 84e09538 nt!IopSynchronousServiceTail+0x5e
ecbecd00 80556cea 000000bc 00000000 00000000 nt!IopXxxControlFile+0x5c2
ecbecd34 8052d571 000000bc 00000000 00000000 nt!NtDeviceIoControlFile+0x28
ecbecd34 7ffe0304 000000bc 00000000 00000000 nt!KiSystemService+0xc4
0012fb4c 00000000 00000000 00000000 00000000 SharedUserData!SystemCallStub+0x4
```

- The OS Kernel (any OS kernel) is rarely at fault
- VDriver is the only 3<sup>rd</sup> party driver on the stack making him the prime suspect
- Note that the stack ends with a KeBugCheck (fault) just after VDriver ran

# Debugger Commands

- Output from *!drivers*
  - Then type in *!drivers* which lists all drivers that were loaded on the machine when it crashed, where they are, as well as their versions, dates and times



# Debugger Commands

- `kd> !drivers`

# Debugger Commands

- Output from *!drivers* (HIGHLY truncated)

```
kd> !drivers
```

```
System Driver and Image Summary
```

Base	Code Size	Data Size	Image Name	Creation Time
804d0000	17e400 (1529 k)	4b500 (302 k)	ntoskrnl.exe	Mon Feb 25 18:32:36 2002
8069a000	f900 ( 63 k)	3580 ( 14 k)	hal.dll	Fri Aug 17 16:48:11 2001
f9f4d000	1100 ( 5 k)	780 ( 2 k)	KDCOM.DLL	Fri Aug 17 16:49:10 2001
f9e5d000	1800 ( 6 k)	1500 ( 6 k)	BOOTVID.dll	Fri Aug 17 16:49:09 2001
<b>ff9b8000</b>	<b>4a980 ( 299 k)</b>	<b>af80 ( 44 k)</b>	<b>VDriver.dll</b>	<b>Fri Sep 28 10:12:47 2001</b>
...				
..				
.				

***Common to expect 150 drivers listed!***

# Debugger Commands

- Output from *!drivers*

```
kd> !drivers
```

System Driver and Image Summary

Base	Code Size	Data Size	Image Name	Creation Time
<b>804d0000</b>	17e400 (1529 k)	4b500 (302 k)	ntoskrnl.exe	Mon Feb 25 18:32:36 2002
<b>8069a000</b>	f900 ( 63 k)	3580 ( 14 k)	hal.dll	Fri Aug 17 16:48:11 2001
<b>f9f4d000</b>	1100 ( 5 k)	780 ( 2 k)	KDCOM.DLL	Fri Aug 17 16:49:10 2001
<b>f9e5d000</b>	1800 ( 6 k)	1500 ( 6 k)	BOOTVID.dll	Fri Aug 17 16:49:09 2001
<b>ff9b8000</b>	4a980 ( 299 k)	af80 ( 44 k)	VDriver.dll	Fri Sep 28 10:12:47 2001



Base

- The beginning of the address range

# Debugger Commands

- Output from *!drivers*

```
kd> !drivers
```

System Driver and Image Summary

Base	Code Size	Data Size	Image Name	Creation Time
804d0000	17e400 (1529 k)	4b500 (302 k)	ntoskrnl.exe	Mon Feb 25 18:32:36 2002
8069a000	f900 ( 63 k)	3580 ( 14 k)	hal.dll	Fri Aug 17 16:48:11 2001
f9f4d000	1100 ( 5 k)	780 ( 2 k)	KDCOM.DLL	Fri Aug 17 16:49:10 2001
f9e5d000	1800 ( 6 k)	1500 ( 6 k)	BOOTVID.dll	Fri Aug 17 16:49:09 2001
ff9b8000	4a980 ( 299 k)	af80 ( 44 k)	VDriver.dll	Fri Sep 28 10:12:47 2001



- Code Size  
The amount of space for driver code

# Debugger Commands

- Output from *!drivers*

```
kd> !drivers
```

System Driver and Image Summary

Base	Code Size	Data Size	Image Name	Creation Time
804d0000	17e400 (1529 k)	<b>4b500 (302 k)</b>	ntoskrnl.exe	Mon Feb 25 18:32:36 2002
8069a000	f900 ( 63 k)	<b>3580 ( 14 k)</b>	hal.dll	Fri Aug 17 16:48:11 2001
f9f4d000	1100 ( 5 k)	<b>780 ( 2 k)</b>	KDCOM.DLL	Fri Aug 17 16:49:10 2001
f9e5d000	1800 ( 6 k)	<b>1500 ( 6 k)</b>	BOOTVID.dll	Fri Aug 17 16:49:09 2001
<b>ff9b8000</b>	<b>4a980 ( 299 k)</b>	<b>af80 ( 44 k)</b>	<b>VDriver.dll</b>	<b>Fri Sep 28 10:12:47 2001</b>

- Data Size  
The amount of space for driver data

# Debugger Commands

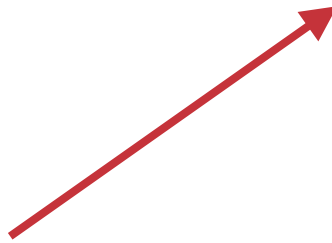
- Output from *!drivers*

```
kd> !drivers
```

System Driver and Image Summary

Base	Code Size	Data Size	Image Name	Creation Time
804d0000	17e400 (1529 k)	4b500 (302 k)	<b>ntoskrnl.exe</b>	Mon Feb 25 18:32:36 2002
8069a000	f900 ( 63 k)	3580 ( 14 k)	<b>hal.dll</b>	Fri Aug 17 16:48:11 2001
f9f4d000	1100 ( 5 k)	780 ( 2 k)	<b>KDCOM.DLL</b>	Fri Aug 17 16:49:10 2001
f9e5d000	1800 ( 6 k)	1500 ( 6 k)	<b>BOOTVID.dll</b>	Fri Aug 17 16:49:09 2001
<b>ff9b8000</b>	<b>4a980 ( 299 k)</b>	<b>af80 ( 44 k)</b>	<b>VDriver.dll</b>	<b>Fri Sep 28 10:12:47 2001</b>

- Image Name  
Address owner





# Debugger Commands

- Output from *!drivers*

```
kd> !drivers
```

System Driver and Image Summary

Base	Code Size	Data Size	Image Name	Creation Time
804d0000	17e400 (1529 k)	4b500 (302 k)	ntoskrnl.exe	Mon Feb 25 18:32:36 2002
8069a000	f900 ( 63 k)	3580 ( 14 k)	hal.dll	Fri Aug 17 16:48:11 2001
f9f4d000	1100 ( 5 k)	780 ( 2 k)	KDCOM.DLL	Fri Aug 17 16:49:10 2001
f9e5d000	1800 ( 6 k)	1500 ( 6 k)	BOOTVID.dll	Fri Aug 17 16:49:09 2001
<b>ff9b8000</b>	<b>4a980 ( 299 k)</b>	<b>af80 ( 44 k)</b>	<b>VDriver.dll</b>	<b>Fri Sep 28 10:12:47 2001</b>

- Creation Time
  - Compile date of the driver
  - This is **not** necessarily the same as the driver date

# Debugger Commands

- Output from *lmv*

```
kd> lmv
```

```
bf9b8000 bfa0dc00  VDriver      (no symbolic information)
Loaded symbol image file: VDriver.dll
Image path: \SystemRoot\System32\VDriver.dll
Checksum: 00058BD5  Timestamp: Fri Sep 28 10:12:47 2001 (3BB4855F)
File version:      5.20.10.1066
Product version:   5.20.10.1066
File flags:        8 (Mask 3F) Private
File OS:           40004 NT Win32
File type:         3.4 Driver
File date:         00000000.00000000
CompanyName:       Video Technologies Inc.
ProductName:       VDisplay Driver for Windows XP
InternalName:      VDriver.dll
OriginalFilename:  VDriver.dll
ProductVersion:    5.20.10.1066
FileVersion:       5.20.10.1066
FileDescription:   Video Display Driver
LegalCopyright:    Copyright© Video Technologies Inc.2000-2001
```

# Debugger Commands

Output from *lmv*

```
kd> lmv
```

```
bf9b8000 bfa0dc00  VDriver      (no symbolic information)  
Loaded symbol image file: VDriver.dll  
Image path: \SystemRoot\System32\VDriver.dll
```



**Location of Culprit**

# Debugger Commands

*Output from lmv*

*kd> lmv*

*bf9b8000 bfa0dc00 VDriver (no symbolic information)*

*Loaded symbol image file: VDriver.dll*

*Image path: \SystemRoot\System32\VDriver.dll*

*Checksum: 00058BD5 Timestamp: Fri Sep 28 10:12:47 2001 (3BB4855F)*

**File version: 5.20.10.1066**



**Version Data**

# Debugger Commands

Output from *lmv*

kd> lmv

```
bf9b8000 bfa0dc00  VDriver      (no symbolic information)
Loaded symbol image file: VDriver.dll
Image path: \SystemRoot\System32\VDriver.dll
Checksum: 00058BD5  Timestamp: Fri Sep 28 10:12:47 2001 (3BB4855F)
File version:      5.20.10.1066
Product version:   5.20.10.1066
File flags:        8 (Mask 3F) Private
File OS:           40004 NT Win32
File type:         3.4 Driver
File date:         00000000.00000000
CompanyName:      Video Technologies Inc.
```

**Who Built It**



# Debugger Commands

Output from *lmv*

kd> lmv

```
bf9b8000 bfa0dc00  VDriver      (no symbolic information)
Loaded symbol image file: VDriver.dll
Image path: \SystemRoot\System32\VDriver.dll
Checksum: 00058BD5  Timestamp: Fri Sep 28 10:12:47 2001 (3BB4855F)
File version:      5.20.10.1066
Product version:   5.20.10.1066
File flags:        8 (Mask 3F) Private
File OS:           40004 NT Win32
File type:         3.4 Driver
File date:         00000000.00000000
CompanyName:       Video Technologies Inc.
ProductName:        VDisplay Driver for Windows XP
InternalName:       VDriver.dll
OriginalFilename:   VDriver.dll
ProductVersion:     5.20.10.1066
FileVersion:        5.20.10.1066
FileDescription:  Video Display Driver that doesn't work very well
```

**Description of Product**

# Debugger Commands

- OK. Done (...maybe)
  - ***!analyze -v*** told you that VDriver was (most likely) your problem
  - ***!drivers*** confirmed its location
  - ***Lmv*** told you its version, who made it, and what it does
- But, what if there was no or too little vendor information?
- Go a Little Deeper

# Debugger Commands

- Now go to the vendor web site
  - Has there been an update?
  - Are there any TIDs/Knowledge Base Articles?
  - If no helpful info
    - Report the event to the vendor
    - They'll appreciate you being able to send along supporting debug information
    - Or they may want the whole dump file itself



# Debugger Commands

- Go a little deeper
  - What if there is no 3<sup>rd</sup> party driver on the stack?
  - If you still don't have a good answer, walk the stack
    - If it crashed in a 3<sup>rd</sup> party driver, you likely have the answer
    - If not but there is a 3<sup>rd</sup> party driver on the stack, he is probably guilty
    - If no 3<sup>rd</sup> party driver on the stack...it gets a lot tougher

# Stacks and Stuff

- Stacks and other debugger spew
- 3 primary things active:
  - Processes
  - Threads
  - Stacks
- Key OSs are similar in this way

# Stacks and Stuff

- Stacks and other debugger spew
  - Process
    - A *Process* occurs when running applications that perform tasks that you request of them
  - Threads
    - When processes are created, one or more *threads* are created

# Stacks and Stuff

- Stacks and other debugger spew
  - Process
  - Threads
  - Stacks
    - A *stack* keeps track of what a particular thread is doing

# Debugging Strategy

STACK\_TEXT:

```
ecbecc1c f7e17756 69696969 f7e17410 04515f10 nt!KeBugCheck+0x10
ecbecc34 804ea221 852daf18 84bada80 806abfe0 VDriver+0xbd44
ecbecc44 8055d0fe 84badaf0 84e09538 84bada80 nt!IopfCallDriver+0x31
ecbecc58 8055de46 852daf18 84bada80 84e09538 nt!IopSynchronousServiceTail+0x5e
ecbecd00 80556cea 000000bc 00000000 00000000 NastyDiskDriver+0x5c2
ecbecd34 8052d571 000000bc 00000000 00000000 nt!NtDeviceIoControlFile+0x28
ecbecd34 7ffe0304 000000bc 00000000 00000000 UntestedUSBDriver+0xc4
0012fb4c 00000000 00000000 00000000 00000000 SharedUserData!SystemCallStub+0x4
```

- Note in this case, there were three guys in the room when the gun went off...

# Debugging Strategy

- So here are some final thoughts
  - After gathering what evidence you have
    - Check for updates
    - Did any change occur at the beginning of the crash events?
    - Were any drivers were NOT present for other similar events?
    - Try swapping hardware brand to test device and driver
    - If the crash was in NTOSKernel, he's VERY unlikely the culprit, look for someone else (this goes for other OS kernels as well)

# Debugging Strategy

## *So like I said in the beginning*

- The most laborious effort you will need to face is listening to me for an hour
- Finding a misbehaved driver in a memory dump often takes seconds
- OK. Many files are beyond the level of what we are doing here, but the point is that ***some are not***

# Debugging Strategy

## *Command Recap*

!analyze -v	Analyze dump file
!drivers	Loaded driver info
!mv	Loaded driver detail



# Debugging Strategy

- Quick demonstration of the debugger...

# Books on Debugging

- Books:  
*Out of print but available*
  - Windows
    - Debugging Windows Applications
      - John Robbins
      - ISBN: 0-7356-0886-5
- Other BSOD-related Enterprises:

**Errorwear.com**



# Geek Question

- How much of a geek are you?  
How can Halloween equal Christmas?
  - 31 Oct = 25 Dec

*Huh?*

$$- \quad 3 \times 8 + 1 \quad = \quad 2 \times 10 + 5 \quad = \quad 25$$

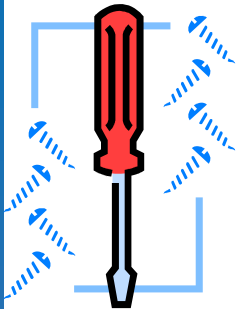
# Tool Time:

## Let's review our tool box!



### Config.nlm (DS) - (Loaded with NetWare OS.)

- Config.nlm creates a static file that isn't much help diagnosing a crash because it doesn't track system changes. However, it's critical for disaster recovery preparation.

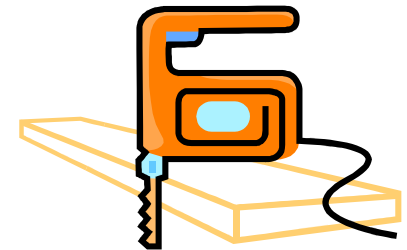


### Toolbox (AH) - (Loaded with NW51SP5 or NW6SP3 or later.)

- Allows you to purge deleted files and execute DOS-like commands and batch files at the server prompt.

### Dsrepair (AH) (Loaded with NetWare OS.)

- This, or NDS iMon – learn it, live it, love it.



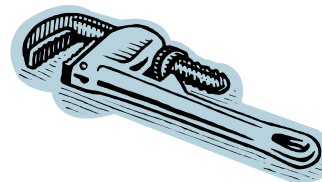
### SPK (DS) - (Available at <http://www.alexander.com>)

- NetWare
- Windows



### RecoverySafe (DS) - (Available at <http://www.alexander.com>)

- Windows Servers
- Windows PCs



# Submit Your Stories & Topic Requests

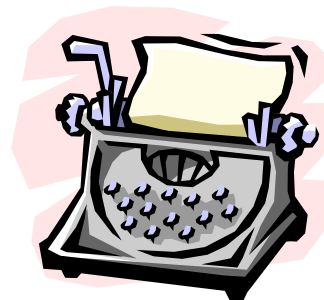
We're gathering stories of woe and wonder from network admins, resellers, and consultants, for a series of articles for publication we're writing.

(We're also looking for article topic requests. )

Please let us know if you wish to remain anonymous or if it's OK to publish your name as a source.

Send your stories to:

[stories@alexander.com](mailto:stories@alexander.com)



# Q&A, Survey Card, & SPK Give Away

- Survey card
- Draw for Software Raffle
- Hand outs
- Write to me if you want:
  - This PPT
  - PDF for setting up WinDbg Follow up questions:  
[dirk@alexander.com](mailto:dirk@alexander.com)

# Thank you!

## Contact us

Dirk Smith:  
Alexander LAN, Inc.:

[dirk@alexander.com](mailto:dirk@alexander.com)  
<http://www.alexander.com>

Allan Hurst:  
KIS Computer Center:

[allanh@kiscc.com](mailto:allanh@kiscc.com)  
<http://www.kiscc.com>

## Resources

Microsoft Downloads:  
Microsoft Debugger:

<http://www.microsoft.com/downloads>  
<http://www.microsoft.com/ddk/debugging/default.asp>

Novell Downloads:  
Novell File Finder:

<http://download.novell.com>  
<http://support.novell.com/servlet/filefinder>

*Merci*

*Obrigado!*

*Gracias*

Bedankt

شكراً

תודה

*Vielen*  
**Dank**

ขอบคุณ

# HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

