



This content is no longer being updated or maintained. The content is provided “as is.” Given the rapid evolution of technology, some content, steps, or illustrations may have changed.

Learn > Collaboration

Contents

Introduction

Troubleshooting Lotus Domino hangs and

What are server hangs and crashes?

crashes

Troubleshooting

Troubleshooting tools
Kiran Bellari

Published on January 17, 2006
New troubleshooting features in Domino 7



Conclusion

Downloadable resources
Lotus Domino is built to be very reliable. But even the best-built products may encounter problems that cause them to hang or crash. When this happens, the quicker you can isolate, analyze, and fix the problem, the quicker your user community will be happily up and running -- and the quicker you can go back to worrying about other things.

Comments

This article offers some ideas you can use to fix Notes/Domino problems. We start by defining the differences between a server hang and a server crash, and how you can go about solving examples of each. We conclude with an overview of new troubleshooting features included in Notes/Domino 7, the latest release of the product. We assume you're an experienced Domino administrator, and are familiar with basic Notes/Domino concepts and terminology.

What are server hangs and crashes?

Before we get into the technical details, let's define two commonly used terms, crash and hang, to ensure we're all on the same page.

Server crash

A Domino server **crash** is a situation where the server program has terminated and it is no longer running. You can often determine the task that the server was performing when it terminated by looking at the crash screen, or from the NSD/RIP log file (depending on which release of Domino you are running).

Common symptoms of a Domino server crash include:

- The Domino server is no longer running, but other programs on the system are still running.
- The Domino server console does not appear, even when tasks appear to be loaded.
- The Domino server loaded and abruptly came down without doing anything.
- A panic error appears on the console or in Log.nsf, and the system comes down.

NSD/RIP automatically runs and generates a file, and the server comes down and/or restarts by itself.

There are several different types of server crashes. For example, a **one-time crash**, as the name implies, may occur once and never appear again. A one-time crash may be caused by bad memory or a corrupted document accessed by a process that resulted in Domino crashing. For example, suppose a document deposited in Mail.box is corrupted. When the Domino router accesses Mail.box to route the document to its destination, this produces a Domino server crash. A similar situation may or may not occur in the future. In general, one-time crashes are the most difficult to analyze.

A **reproducible crash** is one that can be repeated by following a sequence of steps. One example is a form that includes a badly coded button that always results in a crash when pressed.

Contents

Repetitive crashes occur on a particular schedule. They don't seem to be associated with any specific actions; instead, they may happen at the same time every day. In such situations, you need to identify exactly what is getting executed on the server at that time that may be causing the problem. For instance, imagine that a Domino server has a scheduled agent enabled that runs every month. This agent may be producing the server crash. In such scenarios, you need to first disable the agent creating the problem and then review why the agent is causing the problem (and fix it).

Troubleshooting

An **ABEND** is a special form of server crash. The term ABEND is a combination of the words "abnormal end." ABEND crashes do not produce RIP or NSD files.

New troubleshooting features in Domino 7

Causes of crashes include:

Downloadable resources

A software problem in the code (either on the server or on the client).

Related topics

Corruption in a database.

Comments

A software problem in a third-party application accessing Domino.

Insufficient memory.

Restricted operations caused by customized code.

A memory leak.

An incomplete request.

Server hang

A Domino server **hang** is a situation where the Domino server is still running, but one or more tasks on the server are not responding to requests. These tasks may still be active, but they are not doing what they are supposed to do. The term "hang" also defines a state that sometimes occurs when computer programs do not run as designed. Most of the time, a hang occurs due to a low-level loop or a permanent unavailability of a resource, causing serious performance issues. (Server hangs are most commonly attributed to resource issues, so they are sometimes considered performance problems.)

During a hang, the program seems to be paralyzed, no error messages are displayed, and the screen freezes or the application does not respond to users' actions. Keyboard input or mouse clicking has no effect, regardless of where the cursor is placed, but the program is still running. Unlike an ABEND or crash, sometimes a hang will resolve itself, and the application resumes its normal execution without your involvement. Such a case might be considered more of a performance issue than a hang.

Symptoms of a Domino server hang include:

Domino is still running, but is not responsive to clients. In this case, users often report that they are receiving "Server not responding" messages.

The console behaves as if it is disconnected and won't accept any commands, not even a simple command such as quit.

Clients accessing the server (for example, opening databases) are experiencing slow response times.

Semaphore timeouts are occurring. The 'show stat' command will record semaphore timeout information. The following is an example of semaphore timeouts recorded in Statrep.nsf: Sem.Timeouts = 430D: 58 0A13:42 030B:28 0116:26 0A12:21. In this example, 430D is the semaphore name, and 58 is the number of timeouts. Note that semaphore timeouts do not always indicate a performance problem. It is common for semaphore timeouts to occur on a busy server. The statistic Sem.timeouts will not appear in Statrep.nsf if the server has not experienced any semaphore timeouts.

Performance-related error messages are reported, such as:

Insufficient memory.

Insufficient memory. NSF Folder Pool is full.

Maximum number of memory segments that Notes can support has been exceeded.

Network operation did not complete in a reasonable amount of time.

Server not responding.

Contents

Introduction

What are server hangs and crashes?

Note that in a server hang situation, an NSD/RIP is never generated automatically.

Troubleshooting

Causes of server hangs include resource problems (insufficient resources), third-party application conflicts, and hardware

Troubleshooting In general,

server hangs are more difficult to analyze than server crashes. One final note: crashes and hangs not

only occur on the Domino server, they can also happen on the Notes client.

New troubleshooting features in Domino 7

Conclusion

Troubleshooting

Downloadable resources

Related topics

In this section, we examine some general approaches to troubleshooting server crashes and server hangs.

Comments

Troubleshooting Domino server crashes

If Domino has crashed and is not able to restart, remove tasks from the Notes.ini variable Servertask and attempt to narrow down and identify the task causing the crash. When you suspect a particular task is causing the problem, open the server console and narrow down the possible error messages generated by task. For example, if the router crashed while accessing mail in Mail.box, rename Mail.box and allow the server to recreate Mail.box.

If you suspect the problem is caused by a corrupted database, run offline maintenance tasks on this database. If the crash is occurring on a scheduled basis, review the actions performed on the server at the time of the crash.

Consider the following questions:

Is the Domino server reporting error messages to the console or the log file?

What is the exact syntax of the error message?

Where is the error message being generated, on the Domino server or on the Notes client?

When did this problem first appear?

Did you implement any recent changes before the problem started appearing?

Troubleshooting Notes client crashes

First, find out whether or not the problem is specific to a single user. If so, check the configuration of that user and compare it to configurations for other users. Also, determine whether or not the problem happens due to a specific application being

accessed. If so, review the application with a developer.

If you suspect the problem is caused by a corrupted database or document, run the maintenance tasks Updall, Fixup, and Compact (with appropriate switches). Also, try to recreate the database's full-text index, if possible, if you think the problem is due to a bad index.

Troubleshooting Domino server hangs

If constant semaphore problems appear on the server console, check whether or not the tasks' schedule is conflicting. If the system is responding slowly, check your non-Domino applications to see whether or not they are also performing slowly. Additionally, as a general rule, make sure your operating system is updated with all the latest patches.

Introduction

NSD analysis

Determining the process that crashed the server is often the first step in resolving a server crash. In Domino 6 and later, the NSD file can be a good place to start. NSD gives you all current information about the state of the server (call stacks for all threads, memory information, and so on). In the event of a crash, an NSD log file will automatically be generated by the Domino server and stored in the data\IBM_TECHNICAL_SUPPORT directory. An NSD log will have a file name with a time stamp showing the time when the NSD was generated. For example: Nsd_W32I_KIRANTP_2006_01_17@17_17_18.log indicates this NSD was created on January 17, 2006. When NSD runs, it attaches to each process and thread, to dump the call stacks. This can help you determine the cause of a server or workstation crash.

Downloadable resources

The "heart" of an NSD file is the stack trace section. This section provides a breakdown of the code path each thread in a currently existing process traversed to put it in its current state. This is very helpful in examining hang or crash situations on a server. Also, by examining the NSD file, you can find any core files generated in a Domino data directory, and can do a base-level analysis to trace the final stack of calls that were made by the process that died and left behind the core. In a complex product such as Domino, a stack trace of the same type of action on two different servers can produce different results.

In the NSD file, you can identify the executable in the failing process by performing a word search for "fatal," "panic," or "segmentation." By finding the process, we can see what preceded it, and hopefully determine how the crash occurred. When neither "panic" or "fatal" are found, sometimes a core dump will contain a reference to a "segmentation fault" in a function. This indicates that the process tried to access a shared memory segment that was corrupted for some reason, and will crash without calling "fatal_error" or "panic."

The following is a sample excerpt from an NSD file where a server process is involved in a crash:

```
### FATAL THREAD 39/83 [ nSERVER:07c0: 2764]
### FP=0743f548, PC=60197cf3, SP=0743ebd0, stksize=2424
Exception code: c0000005 (ACCESS_VIOLATION)
#####
@[ 1] 0x60197cf3 nnotes._Panic@4+483 (7430016,496dae76,0,496dace8)
@[ 2] 0x600018a4 nnotes._OSBBlockAddr@8+148 (1153f38,2000000,743f608,1)
@[ 3] 0x6000bd92 nnotes._CollectionNavigate@24+610 (0,743fc74,f,0)
@[ 4] 0x600626cc nnotes._ReadEntries@68+2860 (4c5440e8,4cfb8dba,800f,1)
@[ 5] 0x600b9f6f nnotes._NIFReadEntriesExt@72+351 (0,4cfb8dba,800f,1)
@[ 6] 0x10032d40 nserverl._ServerReadEntries@8+1424 (0,8d0c0035,4b64b5bc,4ae46dd6)
@[ 7] 0x100191fc nserverl._DbServer@8+2284 (41b0383,cb740064,0,23696f8)
@[ 8] 0x1002b8c8 nserverl._WorkThreadTask@8+1576 (4711d68,0,3,563fb10)
@[ 9] 0x100016cb nserverl._Scheduler@4+763 (0,563fb10,0,10ec334)
@[10] 0x6011e5e4 nnotes._ThreadWrapper@4+212 (0,10ec334,563fb10,0)
```

When the failing process has been determined, you can focus on troubleshooting that particular process.

ServerTasks

If a server is crashing continuously (for example, every five minutes), a useful troubleshooting step is to temporarily remove the ServerTasks= line from the server's Notes.ini file. The server can then be restarted and tasks can be loaded individually to determine which process is causing the crash.

Panic messages

When Domino detects an internal consistency error, or a condition that may lead to corruption of data or some other problem, it immediately calls a subroutine called Panic. This is a special construct used to continually monitor critical parts of the code as it operates. This helps catch problems as early as possible, before they escalate and possibly destroy data. When a panic takes place, it brings the system to a stop (and thus can be considered a controlled crash). Panics generate messages, sometimes in English and sometimes in code (for example: PANIC: 04:3C). You can give this code to Lotus Software Technical

Support for further troubleshooting.

Troubleshooting tools

This section reviews some of the troubleshooting tools available to you when you encounter a Domino server crash or hang.

Before using any of these tools, be sure to consult the [Domino administration documentation](#).

RIP (Domino R5)

A RIP file is generated when a server crashes. This file contains information about what the server was doing when it crashed. It reports any crash on the system, not just ones related to Domino. RIP files are generated only in Domino 5.x. In Domino 6 and later, NSD serves the purpose formerly performed by RIP, and also includes additional capabilities not included in RIP.

For a RIP file to be generated, QNC.EXE needs to be loaded on the Domino server. The QNC.EXE program (often called "quincy") is the default debugger program that ships with Domino. The QNC.EXE program is usually located in the \Domino directory. To enable QNC.EXE, type "qnc -I" at the operating system's command prompt. You can also enable QNC.EXE by typing "qnc nserver" at server launch. If RIP files are not generated when the server crashes, check whether QNC.EXE is enabled. Normally, RIP files get created in the data directory.

NSD (Domino 6 and later)

As mentioned previously, Domino 6 and later provides the NSD feature. This is a file that contains information about the state of the server at the time of a crash. For more information, see the section, "NSD analysis," earlier in this article.

Memory dump (Domino 6 and later)

In Domino 6 and later, you can use the command "sh memory dump" on the server console to create a memory dump file. A memory dump contains information on memory currently used by Domino. This is very useful when troubleshooting performance problems and memory leaks. Normally, memory dump files get collected in the data\IBM_TECHNICAL_SUPPORT directory. A memory dump file name includes a time stamp for the time when the NSD was generated. For example:

Note: To record the available memory to a file instead of viewing it on the server console, enter the following server console command: `sh memory dump >memory.txt`

HTTP request logs

To troubleshoot issues related to Domino Web server crashes and hangs, Lotus Software Technical Support will often ask you to create an HTTP request log. To enable the default settings for request logs, edit the server's Notes.ini file and add the line `HTTPEnableThreadDebug=1`. This sets HTTP request logging at the default level. (To set the logging level to record more details, see the [Domino administration documentation](#).) You can also enable HTTP request logging dynamically by entering `"set http debug thread on | off"` at the Domino server console. With HTTP request logging enabled, Domino creates a series of files with the name `htthr*.log`. For example: `htthr_a40_10_20050914@171556.log`.

What are server hangs and crashes?

HTTP request logging should be used only for troubleshooting specific issues, and usually at the direction of and with assistance from Lotus Software Technical Support. Do not use request logging for other purposes, such as general administration. These log files grow in size over time, so you should not leave this setting enabled for long periods or you could consume all available drive space.

New troubleshooting features in Domino 7

Conclusion

Automatic Data Collection

Downloadable resources

Notes/Domino 6.0.1 introduced the automatic diagnostic data collection tool, also known as Automatic Data Collection, or ADC for short. Automatic Data Collection simply means that, when a Notes client or Domino server crashes, the program gathers all the necessary data to debug the crash and sends it to a mail-in database when the client or server restarts. Administrators then have one location per domain in which they can see all the crashes that have occurred for all clients and servers. This will help eliminate the instances where an administrator or user may not be able to capture the proper data on a client or server crash.

Notes.ini settings

To troubleshoot performance and crash issues, you can enable the following Notes.ini debugging parameters:

`Debug_threadid=1` logs each process and thread ID for each server operation.

`Debug_show_timeout=1` turns on semaphore timeout messages to the console, and creates a semaphore text file called `semdebug.txt`.

`Debug_capture_timeout=10` time stamps each semaphore timeout message.

`CONSOLE_LOG_ENABLED=1` (Domino 6 and later) enables Domino console logging.

Fault recovery for server crashes

You can set up fault recovery to automatically handle Domino server crashes. When the server crashes, it shuts itself down and then restarts automatically, without any administrator intervention. Domino records crash information in the data directory. When the server restarts, Domino checks to see if it is restarting after a crash. If it is, an email is automatically sent to the person or group in the "Mail Fault Notification to" field.

A fatal error (such as an operating system exception or an internal panic) terminates each Domino process and releases all

associated resources. The startup script detects the situation and restarts the server. If you are using multiple server partitions and a failure occurs in a single partition, only that partition is terminated and restarted.

New troubleshooting features in Domino 7

This section briefly discusses some new Domino 7 features that can help you analyze and correct server hangs and crashes.

Domino Domain Monitoring

Contents

One of the most significant and useful server maintenance and troubleshooting features in Domino 7 is Domino Domain Monitoring (DDM). This provides one central location for monitoring all the servers in a domain (or multiple domains). DDM uses programs called probes to gather server information from the individual servers, and then report back to a special database (DDM.nsf) where you can view the collected data. This allows you to monitor, analyze, and troubleshoot a large number of servers from a single Domino Administrator console.

Troubleshooting tools

Activity Trends

New Troubleshooting Features in Domino 7

Conclusion

The Activity Trends feature lets you analyze "historical" server data, to help spot trends that can only be identified over an extended period of time. You can review this data to help predict and avoid future issues. This data is collected from the log file (Log.nsf) and the Catalog task, and stored in the Activity Trends database (Activity.nsf). The Activity Trends Collector task processes this data, and produces "trended" data that you can use for charting and resource balancing.

Comments

Writing status bar history to a log file

You can now enable Notes client logging of status bar messages to the local log file (Log.nsf) or to an external file that you designate. This can help you troubleshoot Notes client crashes. Use the Notes.ini setting `logstatusbar=1` to enable logging of status bar messages to Log.nsf. To view the logged messages, open Log.nsf and then click the Miscellaneous Events view. Status bar messages are appended with Status Msg. To write the status bar messages to an external file, use the Notes.ini setting `Debug_Outfile=<path to file>` with the Notes.ini setting `logstatusbar=1`. For example:

```
logstatusbar=1
```

```
Debug_Outfile=c:\temp\StatusBarLogging.txt
```

This logs status bar messages to the file StatusBarLogging.txt.

The Log.nsf file can also provide a snapshot of actions logged in the status bar before the Notes client crashed.

Fault Analyzer

Fault Analyzer is a new server feature that processes all new crashes as they are delivered to the Automatic Data Collection mail-in database. The Fault Analyzer task searches the database configured for Fault Report documents and determines whether or not the stack matches a crash that has already been seen by a user or server. It adds to the functionality of the Automatic Data Collection feature by analyzing the call stacks that are located in the Fault Report mail-in database, and evaluating them to determine whether or not there are other instances of the same problem.

Fault Analyzer is configured at the same time that you set up Automatic Data Collection (see figure 1). Use the Server

Configuration document to set up Automatic Data Collection on the server and to enable or disable Fault Analyzer.

Figure 1. Configuring Fault Analyzer

Diagnostic Collection Options	
Mail-in Database for diagnostic reports:	Lotus Notes/Domino Fault Reports
Maximum size of diagnostic message including attachments (in MB):	5
Maximum size of NSD output to attach (in MB):	2
Maximum amount of console output file to attach (in KB):	10240
Diagnostic file patterns:	
Remove diagnostic files after a specified number of days:	No
Fault Analyzer	
Run Fault Analyzer on Fault DBs on this server:	Yes
Run Fault Analyzer on:	All mail-in databases on this server
Remove attachments from duplicate fault reports:	No

Contents

Introduction

What are

Troubleshooting

Troubleshooting tools

New troubleshooting features in Domino 7

Conclusion

The Fault Analyzer locates duplicate fault reports, the new crash is reported as a response to the original crash, and attachments are either removed from the response document to save space in the database, or they are saved with the response document.

Automatic Data Collection enhancements

Downloadable resources

When you use the Automatic Data Collection tool to gather information about server crashes, the server is now first checked to see if it is being run under the Domino Controller and, if so, uses the Controller logs. If not, the server is checked to see if console logging is enabled and, if so, uses the console output. Finally, data is extracted from Log.nsf if neither the Domino Controller nor console logging has been set.

Comments

Now you can select which files (using wildcards) will be collected by the Automatic Data Collection tool when it runs on clients or servers. On Notes clients, it is configured using a Desktop Policy Settings document (see figure 2).

Figure 2. Configuring Automatic Data Collection on the Notes client

Policy Settings Lock Down	
<input checked="" type="checkbox"/> Allow users to change the settings on this tab:	
Diagnostic Collection Options	
Mail-in Database for diagnostic reports:	<input type="text" value="r"/> ▾
Prompt user to send diagnostic report:	<input type="text" value="r"/> Yes ▾
Prompt user for comments:	<input type="text" value="r"/> Yes ▾
Maximum size of diagnostic message including attachments (in MB):	<input type="text" value="r"/> 5 ▾
Maximum size of NSD output to attach (in MB):	<input type="text" value="r"/> 2 ▾
Maximum amount of console output file to attach (in KB):	<input type="text" value="r"/> 10240 ▾
Diagnostic file patterns:	<input type="text" value="r"/> ▾
Remove diagnostic files after a specified number of days:	<input type="text" value="r"/> No ▾

On Domino servers, it is configured using the Server Configuration document (see figure 3).

Figure 3. Configuring Automatic Data Collection on the Domino server

Diagnostic Collection Options	
Mail-in Database for diagnostic reports:	Lotus Notes/Domino Fault Reports
Maximum size of diagnostic message including attachments (in MB):	5
Maximum size of NSD output to attach (in MB):	2
Maximum amount of console output file to attach (in KB):	10240
Diagnostic file patterns:	
Remove diagnostic files after a specified number of days:	No
Fault Analyzer	
Run Fault Analyzer on Fault DBs on this server:	Yes
Run Fault Analyzer on:	All mail-in databases on this server
Remove attachments from duplicate	No

Contents:

This allows you to collect diagnostic files from other IBM products, as well as third-party add-ins.

Introduction

There is a possibility that the output sent by Automatic Data Collection could be very large. If this becomes a problem, you can configure Automatic Data Collection to restrict the size of attachments sent by NSD and the console log to the Fault Reports

database (see figure 3).

Troubleshooting tools

Shutdown Monitor

Conclusion

It often takes a long time for the Domino server to actually shut down after you issue a quit or restart server command. To avoid this delay, the Shutdown Monitor task ensures that Domino terminates when requested to do so. If the server doesn't

terminate in the allotted time, the server will forcefully terminate and an NSD log will be generated before termination. The time limit is specified in the Server Shutdown Timeout field of the Automatic Server Restart section of the Server document,

on the Basics tab (see figure 4).

Figure 4. Setting the Server Shutdown Timeout

Automatic Server Recovery	
Run This Script After Server Fault/Crash:	<input type="checkbox"/> (This script must not run NSD)
Run NSD To Collect Diagnostic Information:	<input checked="" type="checkbox"/> Enabled
Automatically Restart Server After Fault/Crash:	<input checked="" type="checkbox"/> Enabled
Cleanup Script / NSD Maximum Execution Time:	600 seconds
Server Shutdown Timeout:	300 seconds
Maximum Fault Limits:	3 faults within 5 minutes
Mail Fault Notification to:	

The default Server Shutdown Timeout setting is 5 minutes. This feature can be disabled using the Notes.ini setting shutdown_monitor_disabled=1.

Process Monitor (Windows platforms only)

The Process Monitor task monitors the processes that should be running as part of the Domino server environment. (This task runs on Microsoft Windows platforms only; this functionality is implemented in Domino for Unix platforms without using a separate server task.) If any of these processes is missing, or if one terminates unexpectedly without completing the usual

Domino server termination routines, this task causes the server to panic and identify which process has prematurely terminated. The Process Monitor task works with Nprocmon.exe, which monitors the Nserver.exe process for abnormal terminations.

This feature can significantly reduce the number of abnormal termination problems, which otherwise are difficult to analyze (because it's often difficult to determine which process has terminated and caused the server problem). To disable the Process Monitor task, set the variable process_monitor_disabled=1 in the server's Notes.ini file.

Conclusion

In this article, we have defined the differences between a Domino server hang and a crash. We have discussed some troubleshooting procedures and tools you can follow when analyzing and fixing Notes/Domino problems. We also looked at new troubleshooting features introduced in Notes/Domino 7. You can consult this article whenever you encounter a hang or crash with the Notes client or Domino server -- which hopefully won't be very often!

What are server hangs and crashes?

Troubleshooting

Downloadable resources

Troubleshooting tools

 [PDF of this content](#)
New troubleshooting features in Domino 7

Conclusion

Downloadable resources

Related topics

Related topics
The developerWorks Lotus article, "[New features in Lotus Domino 7.0](#)," provides an overview of all the new server features introduced in Domino 7.

Comments

Before using any of the troubleshooting tools mentioned in this article, consult the [Domino administration documentation](#).

Comments

Subscribe me to comment notifications

IBM Developer

About

Site Feedback & FAQ

Submit content

Report abuse

Third-party notice

Follow us

Select a language

[English](#)

[中文](#)

[日本語](#)

[Русский](#)

[Português \(Brasil\)](#)

[Español](#)

[한글](#)

[Code Patterns](#)

[Articles](#)

[Tutorials](#)

[Recipes](#)

[Open Source Projects](#)

[Videos](#)

[Newsletters](#)

[Events](#)

[Cities](#)

[Developer Answers](#)

[Contact](#)

[Privacy](#)

[Terms of use](#)

[Accessibility](#)

[Feedback](#)

[Cookie Preferences](#)

United States - English

