

Iris Today Archives

[Copyright IBM](#)



Semaphores (Part 1)

by [Carol Zimmer](#) and
Laura Rutherford
(with Michelle Pflaum)



Level: Advanced
Works with: Domino 5.0
Updated: 11/01/99

It's rush hour at a busy intersection – but there are no stoplights. Do your thoughts conjure up an image of chaos? Most likely they do, because stoplights keep the flow of traffic on a road running smoothly. And similar to stoplights at an intersection, semaphores keep the flow of traffic (in another word, tasks) on the server running smoothly. Administrators need to understand semaphores, as they are part of the Domino server architecture, in order to maintain an efficient and reliable system.

The first in a two part series, this article discusses semaphores, provides information on why they produce timeout messages, and offers tips on how to read and troubleshoot messages. The [second article](#) gives advice on troubleshooting semaphore timeout messages based on real-life scenarios. It also offers tips on minimizing semaphore issues.

What is a semaphore?

A semaphore is a software flag that ensures a server completes certain tasks before it can begin another. In systems that multitask, semaphores are a key ingredient in making those systems run smoothly and reliably.

Notes/Domino locks in a semaphore during one task and unlocks the semaphore when the task completes. For example, when the Indexer rebuilds an index, it locks a semaphore to prevent other tasks from using the view until it finishes rebuilding. If another task then tries to open that view, it has to wait for the Indexer to complete rebuilding the view index and unlock the semaphore.

What if a task waits and waits for a semaphore to unlock? Just as a driver waiting at an intersection becomes curious if the red light is slow to turn green, a user task – through Domino server internals – records how long it waits and triggers a semaphore timeout notification (when the NOTES.INI setting [DEBUG SHOW TIMEOUT](#) is set) if the wait exceeds 30 seconds. The semaphore handling logic then logs a message to the server console. The user task continues to wait for the semaphore to unlock, timing out every 30 seconds, until the “semaphore holder” task ends. In complicated processes, the task may have to wait through several timeouts. While the semaphore timeout message displays, end users may notice sluggish system performance or an unresponsive server, but end users do not see semaphore messages.

Why do semaphore timeouts occur?

Semaphore timeouts can occur in a number of situations. For example, a task locks a semaphore and then does not release it (see possible reasons listed below), leaving tasks in the queue waiting for that semaphore to unlock. Not all semaphore timeout messages signal a serious problem, though they likely indicate an active, fully loaded or possibly overloaded system. Knowing these scenarios can help you differentiate between a real problem and a temporary situation.

Reasons for semaphore timeouts include the following:

- A heavy load on the server, causing processes to delay releasing semaphores.
- A semaphore deadlock – where two tasks are waiting on the resources the other uses and neither can break the loop. For example, Task A locks Semaphore 1 and then tries to lock Semaphore 2. In the meantime, Task B has already locked Semaphore 2 and is now trying to lock Semaphore 1. Task A is stuck waiting for Semaphore 2 and Task B is waiting for Semaphore 1.
- A process that fails to release a semaphore during execution, which then blocks another process awaiting the semaphore.
- A situation where poorly written scripts or programs are in contention for the same database at the same time. For example, in an end user environment, this type of situation can occur if a program is verified to work in a low-scale test configuration but encounters problems when rolled out to a much larger scale production environment.

You can check to see if the system has experienced a semaphore timeout in one of two ways .

- Type **Show Stat Sem.Timeouts** at the Domino server console. If a timeout has occurred, you receive a display similar to the following:
Sem.Timeouts= 430D:58 0A12:42 030B:28 0116:26 0A12:21
- Check the Domino server console for a message that reads:
Session semaphore held for [n] seconds
Make sure to enable the **DEBUG_SHOW_TIMEOUT** setting in NOTES.INI for these messages to display. These messages/errors do not print to the server log file.

Reading a semaphore timeout message

Domino/Notes records the type and count of semaphore timeouts in output from the **Show Stat Sem.Timeouts** console command. You can view the actual semaphore timeout message on the console as well or in the SEMDEBUG.TXT file. See the "Troubleshooting – Collecting information" section below for details on enabling the appropriate NOTES.INI parameters for message display.

Learning how to decipher a semaphore timeout message typically narrows down the focus area of concern. We provide a sample semaphore timeout message below, followed by detail on how to analyze it. This message is an example and the text may differ from message to message.

```
THREAD [009F:00BE-0012] WAITING FOR RWSEM0x4245
open database queue semaphore (@01070246)
(R=0,W=1,WRITER=009F:00AF-0025,1STREADER=009F:00AF-0025) FOR 5000 ms
```

The bold characters in the above message reflect new information added from Domino R4 to R5.

Note: The terminology within this article supports the platforms that have the process and thread architecture. These platforms include Windows NT, UNIX, and OS/2. The process ID corresponds to the Domino Add-in task or Domino (Main) Server task. As many of the tasks are multi-threaded, one or more threads may have an association with a given process. For platforms that do not support threads (but do support

processes), such as Netware and Windows 95 and 98, the handling is slightly different.

In the example, **0x4245** is the semaphore in question. The text following 0x4245 tersely describes what type of semaphore is being requested (in this case, open database queue semaphore). [Part two](#) of this article includes information on the activity with which specific semaphores are associated.

The table below analyzes specific identifiers in the sample semaphore message. See the "Troubleshooting – Reading information" section below for help corresponding thread IDs with specific tasks.

Identifiers	Meaning
[009F:00BE-0012]	009F is the process that requested the semaphore. More specifically, extract 009F and record that as the process ID. This ID represents the process that wants to lock the semaphore even though another process may own the semaphore.
[009F:00BE-0012]	00BE is the virtual thread ID. More specifically, extract 00BE and record that as the virtual thread ID. This information is the value used internally within Domino to identify the executing thread.
[009F:00BE-0012]	0012 is the thread ID that the platform operating system knows. This information is important to record for future debugging efforts and should be passed to Lotus Customer Support to help in troubleshooting. The physical thread information reporting is a result of the Domino R5 thread pool architecture enhancement.
009F:00AF	009F is also the process last locked the semaphore; AF is its virtual thread ID.

In the above example, the same process (009F) is trying to access the semaphore, but different threads are executing at different points in the logic. One thread has the semaphore and is blocking the other thread. Not all semaphore timeouts involve issues this complicated, but this example provides you with detail on one of the more challenging situations.

For more information on reading semaphore timeout messages, see the [Sem.Timeouts Analysis](#) sidebar and Lotus Customer Support Technote #112710, [Semaphores and Semaphore Timeouts](#).

Troubleshooting – Collecting information

To analyze why semaphore timeout messages occur and determine their level of severity, you need to collect information. In general, you should have an understanding of how your system works at different levels of activity. This section offers suggestions on methods of gathering and analyzing relevant information.

[Part two](#) of this article supplies additional troubleshooting advice, based on real-life semaphore timeout scenarios derived from customer observations.

Semaphore messages are not written to the server log file, so taking steps to collect information is even more crucial to troubleshooting. The following strategies collect information from your system that will help you troubleshoot semaphore timeouts.

Know your system behavior at a steady state and active state time

Steady state is the time that users are not stressing the system (such as the start of the day, when users start up at different times). Active state is the time when users are logging on simultaneously and are most active (such as mid-morning and afternoon).

Once you know the system profile at steady state, you can move on to learning how the Domino server behaves at a more active state. Thus, when changes occur gradually or as a spike, you can discern how the impact upon steady or active state performance has changed. Know the differences, the valid ranges, and the potential for change (based on projected future database usage, new applications, and so on) for your Domino server.

In particular, gather such information from the Domino server console using the following commands and their output:

- **Show Stat Sem.Timeouts** – Displays the total count and type of Sem.Timeouts messages that occurred since the start of the Domino server.
- **Show Stat Server.Users** – Displays the count of active users at the current time.
- **Show Stat Database.DbCache.AverageDbOpenTime** – Displays the average time in seconds it takes to open Domino databases when servicing internal and user requests.
- **Show Tasks** – Displays the status and activity of active tasks and their server.
- **Show Perf** – Displays the date, time, number of transactions executed, and number of currently active users on a continuous minute-by-minute basis.
- **Show Dbs** – Displays performance information about a database, such as the number of times a database has been opened, whether the database has been modified, and the number of times a user has had to wait for a lock on the database. This command is new for R5.

Make use of various tools

The following tools can give you information on active processes and their activity level and physical thread ID values. You can use the process information from the semaphore message described in the previous section as an index to locate the process of interest from the tool output.

- **Perfmon** – On the Windows NT platform, use Perfmon to help augment the information Domino Administrator collects for

performance of specific system and Domino components. In particular, you can view the activity level at which Add-in tasks operated via the Process stats.

- **Windows NT Task Manager** – This tool helps you identify the process ID for each task. To view IDs, click the Processes tab and match the process in the Image Name column to the process ID in the PID column. When you view the process ID in a semaphore timeout message, you can then associate it with a task. For more information, see Lotus Customer Support Technote #175384, [How To Utilize SEMDEBUG.TXT and Task Manager](#).
- **Qslice** – On the Windows NT platform, this tool can show the active tasks at any point in time. It exposes physical thread ID values under the associated process ID.
- **NSD** – Developed by Domino engineers, this tool is currently available for the UNIX platform as part of the Domino server package. It shows the stack for the thread in question. For more information, see the [Troubleshooting recommendations for UNIX](#) sidebar.

For more information on other platform tools and on how to improve server performance, download the Lotusphere presentation "How to Deploy Domino R5 for Performance and Scalability" from the [Lotus IT Central Performance Zone Technical Library](#).

Turn on NOTES.INI settings

Various NOTES.INI settings, when enabled, may provide additional insight into a problem situation. In particular, turn on the following:

NOTES.INI Setting	Description
DEBUG_SHOW_TIMEOUT=1	Displays timeout messages on the server console.
DEBUG_CAPTURE_TIMEOUT=1	Writes timeout messages to a SEMDEBUG.TXT file. These messages do not write to the server log file.
DEBUG_SEM_TIMEOUT= <i>n</i>	Rate at which the Domino server console reports semaphore timeout messages in milliseconds. Default is 30000 milliseconds or 30 seconds. Do not change the default setting without contacting Lotus Customer Support.
DEBUG_THREADID=1	Use to correlate messages from the DEBUG_OUTFILE with specific process and thread IDs.
LOG_UPDATE=2 LOG_VIEW_EVENTS=1	Use to delve deeper into database and Indexer related issues.
NSF_BUFFER_POOL_	Use only if your system is in a partition configuration. If not,

SIZE= <i>value</i>	accept the default setting (the underlying code makes a calculation based on a fixed overhead and total memory available). For information on calculating a setting for your partition system, see Optimizing server performance: Port encryption and Buffer Pool settings or the Lotusphere presentation Deploying Domino R5 for performance and scalability .
SERVER_MAX_CONCURRENT_TRANS= <i>n</i>	Use to achieve greater yield from Domino and to guarantee a certain level of activity when many users access the server simultaneously. Default is 20. This can also increase the semaphores available and enables the server to be more efficient when dealing with large bursts of activity. This applies only to Domino configurations that do not support the latest thread pool architecture enhancements (which are found in R5).
SERVER_SHOW_PERFORMANCE=1	Enables additional information on the server console output, including the number of transactions executed in the past minute and the current number of server users.
DEBUG_OUTFILE= <i>path file name</i>	Creates a file that captures server console output, which you can use to correlate messages with specific process and thread IDs. Use this file in conjunction with DEBUG_THREADID. This file does take up disk space and CPU to execute, so you should disable after use. For more information, see Lotus Customer Support Technote #162400, How Should the DEBUG OUTFILE Parameter Be Implemented?

For more information on Domino server commands and NOTES.INI settings, see [Domino 5 Administration Help](#).

Troubleshooting – Reading information

In many cases, you may have to deduce problems by eliminating certain causes. The following strategies offer advice on reading information to troubleshoot the cause of a semaphore timeout.

Read the server log file

A good place to start analyzing semaphore behavior is in the server log file, where output from console commands appears. In particular, the output from the commands **Show Stat Server.Users** and **Show Stat Sem.Timeouts** provide insight into activity that has changed over a given time period.

To determine trend patterns, analyze the server log file on a daily and weekly basis. When analyzing a situation that needs closer monitoring, you may want to capture the Domino server console information within a one to four hour range. This information makes you better prepared to correlate output from **Server.Users** and **Sem.Timeouts** console commands to output from the log file. Also, remain aware of how the values change as usage on the Domino server increases. Note that even though semaphore timeout messages do not appear in the server log file, the total count of **Sem.Timeouts** increases with each semaphore message displayed on the console.

View output from LOG_UPDATE and LOG_VIEW_EVENTS

In the server log file, view the output from the **LOG_UPDATE** and **LOG_VIEW_EVENTS** commands when you believe the issue is database or Indexer related.

The **LOG_UPDATE** command tells you about indexing activity that the Domino Update task performs, providing the name of the database, the name of the view being updated, and status information. If the **LOG_UPDATE** command is set, and the messages regarding the updating views display quickly on the Domino server console, the views are current (that is, the messages display as each view is checked for recent activity).

The output from the **LOG_VIEW_EVENTS** command tells you the reason for activity, such as rebuilding a view, which the Update (Indexer) task also performs. For example, you can see when and which views are being rebuilt in the more popular databases such as the Domino Directory or Mail.Box. Also, you can start to put some of these clues together. For example, if you notice that a given view is frequently listed as being rebuilt, you might want to check into whether available disk space is a limiting issue. If there is not enough disk space, the view rebuild process never completes successfully, and the Domino Administrator notices that the system keeps trying to rebuild the views.

View output from DEBUG_THREADID

In the server log file, view the output from **DEBUG_THREADID** to correlate the **DEBUG_OUTFILE** file messages with a process ID or a thread ID. You need to cross-reference **DEBUG_THREADID** output with your OS listing of processes and their associated IDs. See the "Troubleshooting – Collecting information" section above for more information on process IDs. We recommend that you contact Lotus Customer Support when using **DEBUG_THREADID** as they can guide you through the analysis.

Read Database.DbCache.AverageDbOpenTime information

This statistic keeps track of the average time, in seconds, it takes to open Domino databases when servicing internal and end user requests. You should review this value on a continuous basis, noting its trends over time (if your system typically assumes a consistent behavior over time). This value can serve as an indication of system problems, including

system overload. Always keep in mind that one statistic does not provide you with an overall picture of the health of the server. You should review other statistics as well for a more complete picture.

View activity from Domino Add-in tasks

When analyzing the semaphore timeout changes in synchronization with the server log output, you can also see activity the other Domino Add-in tasks performed. These tasks, on their own or in conjunction with another Domino component, may also have a role in the semaphore timeout.

Reviewing the comment field displayed next to the various Add-in tasks (viewed through output from **Show Tasks** command) gives you additional information about what the task is doing. You can also monitor if a task is stuck, which is the case when a status message doesn't change over a reasonable period of time. Some exceptions, such as when a very large view is being rebuilt, may occur here. Again, your familiarity with the system and its behavior can help you spot a true problem situation.

View Domino server user count

When reviewing messages from the Domino server console or from the server log, view the Domino server user count along with the other reported transactions. To turn on display of the user count, first make sure the NOTES.INI setting SERVER_SHOW_PERFORMANCE is set, or issue the Domino server console command **Show Perf**. From then on, the Domino server user count automatically appears on the server console and becomes part of the log. This information display occurs on a continuous minute-by-minute basis. It contains the date, time, number of transactions executed over the past minute, and number of currently active users.

Analyze Sem.Timeouts information

The Sem.Timeouts output provides a report of the type and the count of semaphore timeouts. The output line can list more than one semaphore timeout type. A large jump in a timeout count over the monitoring period provides good reason to investigate further. As a gauge, an increase in a given semaphore count by approximately 5 to 10 in an hour is an indication that an area needs further investigation.

With the enhanced output in SEMDEBUG.TXT, STATREP.NSF (where statistics are captured), and the server log file, you can isolate the approximate time when there was a higher rate of timeouts for a particular semaphore. Also, try to determine in what area of the product the semaphore timeout occurred (such as, a task within the Domino server attempts to execute down a given path and is blocked). See the [Sem.Timeouts Analysis](#) sidebar and Lotus Customer Support Technote #112710, [Semaphores and Semaphore Timeouts](#), to understand Sem.Timeouts information, particularly how it keeps track of which timeouts occurred and their frequency.

Note that you cannot re-initialize Sem.Timeouts stats, so any change or analysis occurs in terms of the first value captured.

Analyze semaphore timeout messages

The actual semaphore timeout messages are written to the SEMDEBUG.TXT file when you have enabled the NOTES.INI setting DEBUG_CAPTURE_TIMEOUT. Using one of the platform tools (for example, the Windows NT Task Manager) to help you translate the

information reported in these messages, you can learn more about the semaphore under contention, including the tasks involved. See the "Reading a semaphore timeout message" section earlier in this article for help in deciphering a message.

Eliminate extraneous Add-in tasks

If possible, eliminate any extraneous Domino server Add-in tasks, either internal or external (such as an end-user-developed application or program) to Domino. This action helps to minimize the server load and to reduce opportunities for contention, especially in cases where the semaphore messages act as a warning for an overloaded system.

You may want to treat the removal of externally developed Add-in tasks differently than those developed internally. Sometimes externally developed tasks interfere or conflict with the overall processing of Domino Add-in tasks. You can remove tasks temporarily to see if the system experiences any changes. If the semaphore messages disappear, appear at a different rate, or if different messages appear, you now have additional insight as to the cause and effect.

Review NSF_BUFFER_POOL_SIZE

We recommend accepting the default value of the NOTES.INI setting NSF_BUFFER_POOL_SIZE. If you do specify a setting, you should review this setting after upgrading the system to a new Domino release or when system configuration changes. This guideline of accepting the default specification applies to single partition configurations.

Check SERVER_MAX_CONCURRENT_TRANS

Look to see if this NOTES.INI setting is too high. The default for this setting is 20, and we recommend increasing it in increments of 20. If Notes clients are timing out, then this value is set too high. [Part two](#) of this article contains more information on this setting.

Consider changing the DEBUG_SEM_TIMEOUT default

Before you change this NOTES.INI setting, contact Lotus Support for assistance. DEBUG_SEM_TIMEOUT controls the rate in milliseconds at which the semaphore blockages are reported and enables an administrator to be told, at a given time threshold, how long a semaphore request is blocked. For your environment, there might be reason to change from the default from 30000 (30 seconds).

We recommend the next reasonable value as 60000 (60 seconds), but you should base final adjustment on your understanding of the server's behavior and what you want to accomplish. For example, you may want to adjust the default value for DEBUG_SEM_TIMEOUT to something else when running an overused server, because an overused server is more prone to displaying semaphore timeout messages. And you presumably already have the expectation to see semaphore timeout messages (as the messages act as warning about system overuse) based on the workload, so increasing the time interval minimizes the acknowledged situation.

Conclusion

Again, it is worth restating that the presence of semaphore timeouts are not always a bad thing for the system. In fact, a healthy system is likely to encounter semaphore timeouts too. Learning about semaphore notifications and how they occur can help you maintain a reliable and

efficient system.

In this article, you have learned what semaphores are, why they produce timeouts, and how you can check if your system has encountered any. You also know how to read timeout messages and how to troubleshoot the cause of the timeout messages. [Part two](#) of this article provides additional troubleshooting advice, this time based on real-life scenarios. The second part of this article also gives tips on minimizing semaphore issues in your system.

ABOUT CAROL

Carol Zimmet started working at Iris in 1994. She is responsible for evaluating performance and performance tool development on the server team. Carol continues to search for the one-step solution to everyone's performance problems. Carol enjoys bicycling with her kids in the trailer and playing racquetball. She has a longing to return to stained glass!

ABOUT LAURA

Laura worked as a user assistance writer for Lotus until she had her daughter, Kate, in January 1999. Now she spends a good amount of time with some sort of baby food mush on her clothing, daydreaming about the days when "sleeping late" involved an hour later than 7:00 am. Some of Laura's favorite things include her daughter (of course), her husband, her two dogs, and taking long hikes with all of them. Other favorite things include frozen cappuccinos, cheese pizza, and margaritas (plain, on the rocks, and without salt) from Sierras restaurant in Sudbury.

SPECIAL THANKS

Special thanks to Nirmala Venkatraman and Andy Nolet for their help with this article.

What do you
think about
this article?