Lotus software

IBM

Redbooks Paper

Kim Greene
Dave Johnson
Joe Peterson

IBM Lotus Domino for iSeries Performance and Tuning

You've heard about the outstanding performance of the IBM® @server® iSeries™ server with the NotesBench performance benchmarks. iSeries is clearly a leader in Lotus® Domino® performance and scalability. But how do you capitalize on this performance leadership in your own situation? As you deploy your Domino applications on the iSeries server, there are simple steps you can take to ensure that you achieve the best performance from your iSeries Domino servers.

The scope of this IBM Redpaper encompasses Domino for iSeries performance and tuning tips. It covers the following topics:

- Domino release level
- Application tuning
- NOTES.INI settings
- Number of documents in mail files
- Run-time priority tuning
- Network performance tuning
- Integrated file system (IFS) file attributes
- Collection Services

The purpose of this paper is to show you some of the most commonly used tuning tips. Before you implement any of these tuning tips, you must ensure that your iSeries is performing well and there are no other major bottlenecks such as CPU, memory, network bandwidth, or disk I/O. If any of these bottlenecks exist, they will most certainly impact performance of the Lotus Domino environment on your iSeries server.

If your system is not properly sized, you may also experience performance issues. This Redpaper assumes that your system is appropriately sized, and therefore CPU, disk, memory, and network bandwidth are not limiting factors.

Note: This Redpaper refers to several helpful IBM Redbooks™ and Redpapers. You can find these documents on the IBM Redbooks Web site at:

http://www.redbooks.ibm.com

Domino release level

IBM continues to improve performance and add features in its newer releases. Therefore, upgrading to the latest Domino release is one of the fastest and easiest ways to improve performance. To demonstrate this, let's look at the performance improvements when comparing Domino 6 to Domino R5. Additionally, performance improvements are shown for both the Lotus Instant Messaging and Web Conferencing (Sametime®) and Lotus Team Workplace (QuickPlace®) products.

Domino 6 offers some significant performance improvements for almost every environment. All of the performance improvements that we discuss in this paper assume an upgrade from Domino R5 on the iSeries server to Domino 6 on the iSeries server when comparing Domino server performance. The same hardware configuration is also assumed, with the only changes being an upgrade of the Domino server code and additional memory added to the iSeries server configuration. Domino 6 has more efficient algorithms to improve view performance. However these algorithm changes require more memory than the comparable Domino R5 configurations. In general, Domino 6 requires about 15% additional memory compared to Domino R5.

Domino 6 performance improvements

CPU utilization for Domino mail environments has reduced around 40%, and response times for end users has decreased about 30%. Lotus Domino Web Access (iNotes™) has gained in popularity because of the rich functionality it provides for Web-based access to user's mail files. This environment has also improved in Domino 6. Lotus Domino Web Access environments are experiencing around a 45% drop in CPU utilization and about a 35% improvement in response times compared to Domino R5 environments.

Custom-written Domino applications are also seeing quite substantial improvements. Several customers have seen CPU utilization drop by up to 30% to 40% simply by upgrading their Domino R5 servers to Domino 6.

For more details about Domino 6 performance compared to Domino R5 on the iSeries server, refer to the article *iSeries Performance with Domino 6, QuickPlace 3, and Sametime 3*, which you can find on the Web at:

http://www.lotus.com/ldd/today.nsf/Lookup/iSeriesPerf_D6_QP3_ST3

Lotus add-on products

Both the IBM Lotus Team Workplace and IBM Lotus Instant Messaging and Web Conferencing products have some fairly substantial improvements to report. The performance improvements for these two products assume the same level of OS/400® and the same hardware configuration. Only the version of Lotus Team Workplace and Lotus Instant Messaging and Web Conferencing that is used is changing.

QuickPlace 3.0 is seeing a reduction in CPU consumption of about 45% compared to QuickPlace 2.0.8 on the iSeries server. The improvement in response times is even larger, with a 55% reduction in end-user response times. These results are based on an Domino R5 base of the Domino code.

Note: Domino 6.5.1 is the first Domino 6 release to support IBM Lotus Team Workplace.

Sametime 3.0 response times have always been good. There is no measurable difference in end-user response times using the chat capability over Sametime 2.5. However, the CPU utilization for Sametime 3.0 has improved by approximately 45%.

Sametime 3.0a offers even more improvements over Sametime 3.0. The amount of improvement is another 30% to 40% reduction in CPU for chat function. This is a substantial improvement when added on top of the 45% reduction in CPU consumption when moving from Sametime 2.5 to Sametime 3.0.

For the latest information about which releases of Domino work with the different Lotus add-on products, such as Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace, refer to the *Lotus Software for iSeries Compatibility Guide* at:

http://www.ibm.com/servers/eserver/iseries/domino/domino6/releasesupport.pdf

Application tuning

Each release of Domino offers more enhancements to improve performance. Domino 6 is no exception. The previous Domino release-level section points out the performance improvements offered in Domino 6 compared to Domino R5 without any application or configuration changes. The release level of Domino can have quite an impact on the performance of a Domino application.

Another factor that can greatly impact the performance of any Domino application is the application design. If an application is poorly designed, or designed with one set of criteria and the application evolves into an expanded set of criteria, there are often resulting performance problems. One of the most common problems has to do with view indexing. If a Domino application has a view that contains many hundreds or thousands of documents, the view can often become a bottleneck in the application if its design was not based on containing a large number of documents.

One of the most common causes of poorly performing views is the use of time-sensitive formulas. Any view that uses the @Now formulas are always out-of-date and need to be rebuilt. Use of the @Today formula can also cause views to be rebuilt continuously. Using these formulas is good when there are 1000 documents or less in a Domino database. However when the number of documents exceeds 1000, performance of the views degrades. There are several alternatives to using time-sensitive formulas such as using an environment variable in place of the formula or using the @Text formula that contains the date.

A wide variety of additional application design trade-offs are commonly overlooked such as which operator is used in a division operation. Did you know that backslash (\) division is 60% faster than forward slash (/) division? That is because using a forward slash produces a floating point number, and using a backslash produces an integer.

When working with document collections, how you retrieve the documents in that collection can have a huge impact on performance. For example, consider the `GetNextDocument` and `GetNthDocument` methods. Using the `GetNextDocument` method is significantly more efficient than using the `GetNthDocument` method, because of how positioning is done in the document collection. When you use the `GetNextDocument` method, positioning to the next entry in the document collection is very fast. However, when you use the `GetNthDocument` method, positioning starts from the top of the document collection for each individual lookup. This adds a lot more processing time to an application that needs to access each document in the document collection.

To learn about these application tuning tips and many more, refer to the redbook, *Performance Considerations for Domino Applications*, SG24-5602.

NOTES.INI settings

Several variables in the NOTES.INI file can impact the performance of your Domino for iSeries server. You must explicitly add these variables to your NOTES.INI file. Otherwise, the default values for each of these variables is used. The following sections discuss several performance-related variables and the recommended settings.

Note: After setting these variables in the NOTES.INI file, you must restart the Domino server for the changes to take effect.

NSF_Buffer_Pool_Size_MB

Domino has its own memory management code included in the base Domino server code. This code is necessary because not all platforms that support Domino have appropriate memory management algorithms to page necessary data and program objects in and out of memory. On the iSeries server, storage management provides this function for all objects used when executing applications. However, Domino still does its own memory management because it is part of the core Domino server code.

The two pools of memory that are most critical to Domino performance are the *NSF buffer pool* and the *database cache* or *DB cache*. The size of the DB cache is directly related to the size of the NSF buffer pool. DB cache is three times the size of the NSF buffer pool. To change the size of the DB cache, you can either adjust the size of the NSF buffer pool or add the variable NSF_DbCache_MaxEntries to the NOTES.INI file.

You can use the **`show stat database`** command on the Domino server console to see the performance statistics related to how the NSF buffer pool and DB cache are performing.

Note: Allow the Domino statistics to settle before you look at the numbers. You want the Domino server to run for several hours and experience normal or peak workload demands before you use the statistics to adjust the tuning parameters.

You can use two variables in the NOTES.INI file to control the size of the NSF buffer pool. The NSF_Buffer_Pool_Size_MB variable sets the NSF buffer pool to a static size. The size is specified in megabytes. The PercentAvailSysResources variable sets the NSF buffer to a percentage of total memory available on the system. The valid ranges are between 2% and 100%.

The default size of the NSF buffer pool is set to 300 Mb in Release 5 of Domino on the iSeries and 750 Mb in Domino 6 for iSeries. To adjust the size of the NSF buffer pool, add either one of these variables to the NOTES.INI file of the Domino server and then stop and restart the Domino server.

Note: You can set the size of the NSF buffer and the DbCache smaller than the default values. If you do not need additional memory allocated to these buffers, you may adjust their sizes downward to more appropriate settings.

The reason the NSF buffer pool size is so critical to a Domino server is because it provides the I/O buffering mechanism between the Lotus Notes® indexing facility and the data and

programs residing on disk. For the most optimal performance from your Domino server, we recommend that you use the **`show stat database`** command on the Domino server console to see how much of the NSF buffer pool is actually required by the Domino server. If the size of the NSF buffer pool is set too high, then Domino is doing unnecessary work in managing the larger NSF buffer pool. If the size of the NSF buffer pool is too small, then Domino does not have access to the resources necessary to provide optimal view indexing and database caching performance.

The critical statistics that you need to see to determine the correct setting for the NSF buffer pool are:

- Database.Database.BufferPool.Maximum.Megabytes
- Database.Database.BufferPool.Peak.Megabytes
- Database.Database.BufferPool.PerCentReadsInBuffer
- Database.DbCache.CurrentEntries
- Database.DbCache.HighWaterMark
- Database.DbCache.MaxEntries
- Database.DbCache.OvercrowdingRejections

The two critical statistics to look for are Database.BufferPool.Maximum and Database.BufferPool.Peak. Database.BufferPool.Maximum shows what the current setting is for the NSF buffer pool. The Database.BufferPool.Peak statistic shows how large the NSF buffer pool has gotten since the Domino server was last started.

You also want to look the percentage of reads that occurred in the NSF buffer, Database.BufferPool.PerCentReadsInBuffer. You want this to be a relatively high number, generally 95%. If reads are occurring and they are not in the buffer, then the time to do those reads is longer.

Note: It may not be possible to increase the NSF buffer pool to a size that increases the PerCentReadsInBuffer to above 95%. While measuring the NotesBench benchmark, it has been found that in this environment where many thousands of users are concurrently accessing the server, the size of the NSF buffer pool is insatiable. The same holds for customer environments that have thousands of users accessing the server. In these scenarios, increases in the size of the NSF buffer pool are unable to increase the PerCentReadsInBuffer statistic above 95%.

Other statistics you need to pay attention to include Database.DbCache.MaxEntries and Database.DbCache.HighWaterMark. If Database.DbCache.HighWaterMark is equal to or exceeds the size of Database.DbCacheMaxEntries, then you will need to increase the size of the NSF buffer pool to optimize performance. Database.DbCache.MaxEntries is set to three times the size of the NSF buffer pool. The DB cache is used to cache databases. If you have a system where many different databases are accessed, such as a Mail workload, then the size of the DB cache may not be large enough. Because the size of the DB cache is based on the size of the NSF buffer pool, you need to increase the buffer pool if there is not enough room in the DB cache.

By looking at Database.DbCache.CurrentEntries and Database.DbCache.OvercrowdingRejections statistics, you can see whether the NSF buffer pool is set correctly. If Database.DbCache.OvercrowdingRejections is high and Database.DbCache.CurrentEntries is close to Database.DbCache.MaxEntries, then you need to increase the size of the DB cache because there is not enough room in the cache to house the number of databases that are regularly accessed on the Domino server.

Let's look at an example to help you understand better how all of these variables fit together. Here are the values for the key variables:

- Database.Database.BufferPool.Maximum.Megabytes = 170
- Database.Database.BufferPool.Peak.Megabytes = 168
- Database.Database.BufferPool.PerCentReadsInBuffer = 95.44%
- Database.DbCache.CurrentEntries = 295
- Database.DbCache.HighWaterMark = 765
- Database.DbCache.MaxEntries = 510
- Database.DbCache.OvercrowdingRejections = 8385

In this example, we can see that we must increase the size of the NSF buffer pool, because BufferPool.Peak.Megabytes is 168, which is within 5% of BufferPool.Maximum.Megabytes or 170. This information may lead us to increase the NSF buffer by only 5% or so. However, when looking at the other statistics, we see that the DB cache is undersized. DbCache.HighWaterMark is 765, with DbCache.MaxEntries only equal to 510 (3 x 170). Additionally, DbCache.OvercrowdingRejections is large and would obviously have a dramatic impact on overall Domino server performance because of the wait time to page databases into the DB cache.

It is also important to monitor the amount of paging and faulting that is occurring on the iSeries server. OS/400 performance statistics can be found by either using the Work with System Status (WRKSYSSTS) CL command or by using the Work Management features through the iSeries Navigator graphical interface. Because Domino databases reside in the OS/400 IFS, any paging and faulting related to Domino is considered non-database faulting.

Domino servers run out of memory pool number two, *BASE, on the iSeries server by default. Unless you change the routing entries to route the Domino subsystem to run out of another memory pool, this is the memory pool in which you need to monitor the faulting rates. The best guideline to use is to have less than 100 faults occurring per processor in the memory pool where the Domino server is running. With this guideline, a one-way processor is considered acceptable if the faulting rate in the *BASE memory pool is less than 100 faults per minute. Likewise, on a 4-way processor that has not been logically partitioned, it is acceptable to have faulting rates up to 400 faults per minute in the *BASE memory pool.

Important: If you increase the size of the NSF buffer pool too large, you can create more paging and faulting issues. If there is not enough memory on the system, increasing the size of the NSF buffer pool too large can negatively impact memory paging and faulting rates. It is necessary to balance the size of the NSF buffer pool according to the memory available on the system.

NSF_Buffer_Pool_Size_MB verses PercentAvailSysResources

We mentioned earlier that there are two ways to set the size of the NSF buffer pool: using the NSF_Buffer_Pool_Size_MB variable or using the PercentAvailSysResources variable. The PercentAvailSysResources variable works differently than the NSF_Buffer_Pool_Size_MB variable in that it assigns a percentage of the total memory available on the whole system to the NSF buffer pool as appropriate.

Valid values for PercentAvailSysResources are from 2% to 100%. On an iSeries server, you never want to allocate 100% of the memory to the NSF buffer pool. You need to look at the total memory available in the memory pool in which Domino is running and then determine how much of that memory pool to allocate to the NSF buffer pool.

For more details about how to appropriately set the size of the NSF buffer pool or the DB cache, refer to the redbook *Domino 6 for iSeries Best Practices Guide*, SG24-6937.

Note: The recommendations provided in this section are general principles to tune NSF buffer pooling. There may be situations where changing the NSF buffer pool size does not noticeably affect the percent reads. Be sure to monitor end-user response time performance and other system resource metrics when making tuning adjustments.

Notes_SHARED_DPOOLSIZE

DPOOL is used for buffering databases so that they are accessible from memory rather than having to be read from disk. The size of the DPOOL was 256 Mb in Domino R5. It is now 512 Mb in Domino 6. The default memory segment size is 12 Mb for both Domino R5 and Domino 6. This is often more than what is required by most Domino servers.

We do not include details about this tuning tip in this Redpaper. Rather we want you to be aware that this NOTES.INI variable exists. To adjust the size of the DPOOL appropriately, you must perform a more thorough analysis and examination than what this Redpaper covers. For more details, see the redbook *Domino 6 for iSeries Best Practices Guide*, SG24-6937.

ServerTasks

The SERVERTASKS variable controls the tasks that are automatically started when a Domino server is started. By limiting the number of tasks that start, you can improve the performance of that Domino server. For example, if you partition your Domino servers to handle different Domino functions, only start the tasks that are necessary for a particular server. A typical Domino 6 default ServerTasks variable looks like this:

```
ServerTasks=COLSRV400,IMAP,POP3,Amgr,Adminp,CalConn,Replica,Router,Sched,DECS,HTTP,Update
```

Important: Be careful when removing the UPDATE task from the ServerTasks line. If you remove the UPDATE task, you may experience functional problems in your Domino server if your application depends on view and index updates occurring on the fly.

Table 1 contains a list of the different Domino server tasks that can be started with the ServerTasks variable. With a better understanding of each of these tasks, you can limit the number of tasks automatically started with the Domino server. It is likely that a few of these are already in your NOTES.INI file since many are not there by default. However, the defaults have changed with various Domino releases. They also change depending on the options chosen when you configure your iSeries Domino servers. The following tasks are all added at the time a Domino for iSeries server is configured:

- Internet Cluster Manager (ICM)
- Hypertext Transfer Protocol (HTTP)
- Internet Inter-ORB Protocol (IIOP)
- Internet Message Access Protocol (IMAP)
- Post Office Protocol 3 (POP3)
- Lightweight Directory Access Protocol (LDAP)
- Domino Enterprise Connection Services (DECS)
- Net News Transfer Protocol (NNTP)

Also, the ServerTasks variable is preserved when upgrading Domino servers to a new release, so newer functions may not have been added automatically.

Table 1 Domino server tasks

Server task	Command to run task	Description
Activity Trends Collector	Trends	Performs historical and trended analysis on Domino activity data.
Administration Process	Adminp	Automates a variety of administrative tasks. You should not remove this entry.
Agent Manager	Amgr	Runs agents on one or more Domino databases. Generally, you should not remove this entry. Doing so makes unavailable all scheduled agents on the Domino server.
Billing	Billing	Collects all generated billing information. If you are not using the Billing function, you can remove this entry.
Calendar Connector	Calconn	Processes requests for free-time information from another Domino server. If you do not use calendaring and scheduling or if you use it only on one Domino server, you can remove this entry. Removing this task makes free-time lookups unavailable to other Domino servers.
CA Process	ca	Automates a variety of Domino server-based certificate authority tasks.
Change Manager	Runjava ChangeMan	Manages large-scale changes within the Domino domain.
Cluster Database Directory Manager	Cldbdir	Updates the cluster database directory and manages databases with cluster-specific attributes. If you have not implemented clustering or if this Domino server is not a member of any Domino cluster, you can remove this entry.
Chronos	Chronos	Updates full-text indexes that are marked to be updated hourly, daily, or weekly.
Cluster Replicator	Clrepl	Performs database replication in a cluster. If you have not implemented clustering or if this Domino server is not a member of any Domino cluster, you can remove this entry.
Database Compactor	Compact	Compacts all databases on the Domino server to free disk space.
Database Fixup	Fixup	Locates and fixes corrupted Domino databases.
DIIOP	DIIOP	Allows Domino and a Web browser client to use the Domino Object Request Broker (ORB) server program. If you have written or purchased Domino applications, which contain Java™ applets utilizing the Domino Java classes, you need to enable this. It is not enabled by default. If you do not run the HTTP task, you do not need this entry.
Directory Cataloger	Dircat	Populates directory catalogs and keeps the catalogs up-to-date.
Domain Indexer	Domidx	Creates a central, full-text index for all specified databases and file systems in a domain. Runs only on Domain Catalog servers.
Domino Enterprise Connection Services	DECS	Allows real-time access to back-end relational databases.
Event Monitor	Event	Monitors events on a Domino server.
HTTP Server	HTTP	Enables a Domino server to act as a Web server so Web browser clients can access databases and applications on the Domino server. If you do not want to allow Web browser access to the Domino server, you can remove this entry.

Server task	Command to run task	Description
IMAP Server	IMAP	Enables a Domino server to act as a mail drop for IMAP clients. If you have no IMAP clients, you can remove this entry.
Internet Cluster	ICM	Provides failover and workload balancing for HTTP clients (Web browsers) that access Domino Web servers.
ISpy	Runjava ISpy	Sends server and mail probes and stores the statistics. Is used primarily for troubleshooting.
LDAP Server	LDAP	Enables a Domino server to provide LDAP directory services to LDAP clients.
MTC	MTC	Reads log files produced by the router and writes summary data about message traffic to a database for message tracking purposes.
POP3 Server	POP3	Enables a Domino server to act as a mail drop for POP3 clients. If you have no POP3 clients, you can remove this entry.
Replicator	Replica	Replicates databases with other Domino servers.
Reporter	Report	Reports statistics for a Domino server.
Router	Router	Routes mail to other Domino servers. Do not remove this entry.
Runjava	Runjava	Runs Java server add-in tasks such as Change Manager and ISpy.
Schedule Manager	Sched	Returns meeting times and dates and available invitees. If you are not using the calendaring and scheduling functions, you can remove this entry.
SMTP Listener	SMTP	Listens for incoming SMTP connections, enabling Domino to receive mail from other SMTP hosts.
SNMP QuerySet	QurySet	Allows Domino to respond to Simple Network Management Protocol (SNMP) requests. Prerequisite is the Domino SNMP Agent (LNSNMP).
SNMP Interceptor	Intrcpt	Allows Domino to issue SNMP traps for Domino events. Prerequisite is Domino SNMP Agent (LNSNMP).
Statistics Collector	Collect	Collects statistics for multiple Domino servers.
Stats	Stats	Generates statistics for a remote Domino server on demand. Do not remove this entry.
Update	Update	Keeps views and indexes fresh as the Domino server is running.
Web Retriever	Web	Implements the HTTP protocol to retrieve Web pages and convert them into Lotus Notes documents.

Ask Professor INI is an invaluable resource to demystify what all of the NOTES.INI parameters mean and how best to use them when tuning your Domino server. You can find Ask Professor INI on the Web at:

http://www.lotus.com/ldd/today.nsf/profini?OpenView

Running Web agents concurrently

Many factors can impact agent processing time including the number of documents that the agent must process, how busy the Domino server is when the agent is processing, whether enough AMGR tasks are available to accommodate the agent work, or whether the agent has to wait on any other processes in order to execute. Domino agents process synchronously, meaning one at a time, unless you change the setting for the Run Web Agents Concurrently field in the Domino server configuration document.

In the Domino server document, you click Internet Protocols →Domino Web Engine and page down to the Web Agents section. Then you change the value for the Run Web Agents Concurrently field from Disabled to *Enabled*. The value for this variable is set to Disabled by default.

Prior to Domino 6, this variable was adjusted by adding the variable DominoAsynchronizeAgents to the NOTES.INI file of the Domino server to allow asynchronous processing of agents. By setting this variable equal to the value of 1, agents c run concurrently.

Attention: When using *LSXODBC-based agents via HTTP and setting the Run Web Agents Concurrently field to Enabled, this scenario can cause threads for the HTTP job to stop responding. Eventually this causes the HTTP job to hang and stop responding. An alternative is for the HTTP job to fail with a SQL7903 error, causing the Domino server to crash.

Due to the process behind running Web agents concurrently, and the documented OS/400 restriction regarding multithreaded connectivity sharing handles, this specific scenario is unsupported at this time. The only supported configuration is with the Run Web Agents Concurrently field set to Disabled.

For more details about this limitation, see TechNote 1153455, *Domino for iSeries*: *Use of LSXODBC Agents via HTTP with Concurrent Web Agents Enabled Causes Hang or Crash* on the Web at:

http://www.ibm.com/support/docview.wss?uid=swg21153455

Number of documents in mail files

The size of mail files is something with which many companies struggle. More and more companies are implementing archiving policies to help keep the size of mail files to a more reasonable level. This is a good practice, but does the size of a mail file matter? The answer is “it depends”. The key deciding factor is the number of documents stored in that mail file.

The IBM Rochester Domino development team performed a set of measurements using different mail file sizes and different numbers of documents stored in those mail files. The results were quite surprising.

Table 2 shows the impact of the number of documents in a mail file. Two different sizes of mail files were measured: a 700 MB mail file and a 100 MB mail file. The number of documents in the 100 MB mail file was increased from 1,024 to 4,096. There was no measurable difference in performance, even though the document count was increased four times.

However, as shown in Table 2, similar results were also produced for a 700 MB mail file. In this case, the document count started at 7,196 and increased four times to 28,672. In this

scenario, the amount of CPU consumed increased from 8% to 16%, *doubling* CPU demands on the server.

Table 2 Correlation of mail file size and document count to CPU utilization

Mail file size	Document count	CPU utilization
700 MB	28,672	16%
700 MB	7,196	8%
100 MB	4,096	6%
100 MB	1,024	6%

To learn more about this set of measurements, refer to the Redpaper *Sizing Large-Scale Domino Workloads on iSeries*, REDP-3802.

Run-time priority tuning

The iSeries operating system code has some advanced performance management algorithms built into it. Part of the work that these algorithms provide is to appropriately manage the memory and CPU resources that are available to various types of jobs running on the same iSeries server.

In scenarios where multiple workloads run on the same iSeries server, it may be necessary to adjust run-time priorities to obtain optimal performance. Some examples of when this may be required include:

- Domino is running on the same iSeries server that is hosting 5250-based interactive sessions. The interactive sessions run at priority 20, which is the same run-time priority used by Domino jobs. In this case, the Domino workload may need to be scaled back to a lower run-time priority, for example priority 30. This allows the interactive sessions to receive priority for CPU cycles to continue sub-second end user response times for the 5250-based interactive users.
- Different types of Domino servers are running that have different processing requirements. An example is an iSeries server that has four Domino servers:
  - Server 1 runs Domino e-mail.
  - Server 2 hosts a Domino Web order-entry application.
  - Server 3 is a test server.
  - Server 4 is a development server.

  In this scenario, Server 2 takes top priority because it provides the interface for customers to order products from the company. A close second to these processing requirements is Server 1, which provides Domino e-mail services—a critical element in any corporate environment today. Servers 3 and 4 clearly take a back seat to both Server 1 and Server 2.
- Domino is running on the same system that is also hosting WebSphere®. In this scenario, Domino jobs run at priority 20 by default and WebSphere jobs run at priority 25 by default. You need to be aware of these default run priorities when deciding which workload takes precedence in a specific environment.

OS/400 algorithms decide which jobs should receive CPU resources based on their run-time priority. As mentioned earlier, the iSeries OS/400 code has advanced performance management algorithms built in. A critical component to how these algorithms work is based on the run-time priority of individual jobs or tasks in the system. Different bands control how OS/400 storage management dispatches tasks.

These bands are:

- Band 0: 0-9 (system tasks)
- Band 1: 10-16
- Band 2: 17-22
- Band 3: 23-35
- Band 4: 36-46
- Band 5: 47-51
- Band 6: 52-89
- Band 7: 90
- Band 8: 91
- Band 9: 92
- Band 10: 93
- Band 11: 94
- Band 12: 95
- Band 13: 96
- Band 14: 97
- Band 15: 98
- Band 16: 99

The lower the priority number is, the higher the priority is associated with a particular job. The default run-time priority of Domino jobs is 20. A job or task with a run-time priority of 20 gets more CPU resources than a job or task running at a run-time priority of 50.

A dynamic priority scheduler on the iSeries is enabled by default. The priority scheduler looks at all of the tasks that need CPU resources on the system. Then based on their run-time priority, it decides which tasks to dispatch to the CPU unit. If it sees that a specific job is being "starved" for CPU, it internally increases the run priority of an individual job. The rate at which this internal priority is adjusted depends on the band in which the job is located.

All Domino jobs or tasks have a run-time priority of 20 by default. This is the same run-time priority of interactive jobs and can therefore cause performance issues for interactive jobs on the system.

To reduce the CPU consumption of specific tasks, you can assign a higher run-time priority number to those specific Domino tasks that consume the most CPU. The tasks that most frequently consume large amounts of CPU are:

- **Agent manager task (AMGR)**: This task is responsible for the execution of agents on the Domino server. Agents perform a multitude of functions and are crucial elements to many Domino applications.
- **HTTP task**: This task performs Web serving functions on the Domino server.
- **UPDATE**: This is a critical task that is present on the Domino server. It is responsible for updating any indexes over Domino databases when documents are inserted, updated, or removed from the Domino database. UPDATE normally runs continuously on the Domino server and keeps all indexes up-to-date with changes that are happening dynamically in Domino databases.

To give you an idea of how you can adjust the run-time priority of Domino jobs appropriately, on a busy system, some common examples of run-time priority adjustments include setting the following tasks as such:

- UPDATE = 25
- HTTP = 25
- AMGR = 25

- Router = 36
- Clrepl = 36
- Replica = 36

Important: When adjusting the run-time priority of a given job or task, the higher the run-time priority number is, the lower the actual run-time priority is. For example, a job with a run-time priority of 50 receives fewer CPU cycles than a job with a run-time priority of 20.

You can alter how much CPU a given Domino task can consume by altering its run-time priority. By assigning a Domino task a higher run-time priority, it consumes fewer CPU resources. A two-step process is involved in permanently changing the run-time priority of specific Domino tasks or jobs on the system:

1. Create a class with the new run-time priority that you want to assign to specific Domino tasks.

   Each job or task that executes on the iSeries has an associated class that provides information about the characteristics that job should adapt when running on the system. You can provide any name that is meaningful to you for the class name. However, it is imperative that you create the new class in the QUSRNOTES library. This library hosts Domino specific elements. The run-time priority parameter of the Create Class (CRTCLS) CL command provides the run-time priority that you want a specific Domino task or set of tasks to use.

2. Edit the domino_classes stream file in the /QIBM/UserData/LOTUS/Notes directory.

   The first time you edit this file, you find that the file is empty, which is normal. To change the run-time priority of specific Domino tasks, you need to add three different variables to this file:

   - SERVER
   - CLASS
   - TASKS

   Consider this example:

   ```
   SERVER=Mail01
   CLASS=mailclass1
   TASKS=calconn,router
   CLASS=mailclass2
   TASKS=sched
   SERVER=App01
   CLASS=appclass
   TASKS=amgr
   ```

   As mentioned earlier, the domino_classes file is most likely to be empty the first time you edit it. You need to add the three keyword variables to this file:

   - The SERVER variable indicates the Domino server for which you want to change the run-time priority of specific tasks. The value for this parameter needs to be the unqualified name of the Domino server. If you provide the fully qualified name of the Domino server, for example, Mail01/kgci, the Domino server name is not recognized. In our example, the name of the first Domino server we are working with is Mail01.
   - The CLASS variable specifies the class you created in step 1. The class contains the new run-time priority that you want to assign to a specific Domino task or set of tasks. In this example, the mailclass1 class is applied.
   - The TASKS variable lists the specific Domino tasks to which you want to apply a different run-time priority. In this example, the calconn and router tasks have the run-time priority specified in the mailclass1 class.

If you want to change the run-time priority of additional tasks associated with the same Domino server, you can simply repeat the CLASS variable that lists the class, which contains the appropriate run-time priority. Then specify the TASKS variable that lists the Domino tasks, which should adopt yet another run-time priority. In this example, the Domino scheduling task, sched, adopts the run-time priority specified in class mailclass2.

If you have more than one Domino server running on the same iSeries server, simply repeat the SERVER variable with the name of the Domino server. Then specify the CLASS and TASKS variables to indicate which class, with which specific Domino tasks are associated, the Domino server should adopt. In our example, the App01 Domino server adopts the run-time priority specified in the appclass class for the agent manager (AMGR) task.

When you are finished making changes to the domino_classes file, press F3 twice to save your changes. You need to stop and restart the Domino server for the new run-time priorities to be adopted by the Domino tasks listed in this file.

When adjusting the run-time priority of specific Domino tasks, you cannot impact two specific jobs, the QNNINSTS and SERVER tasks. The QNNINSTS job handles the task of monitoring the Domino server and restarting it if the Domino server goes into a panic. This auto restart capability has been built into the Domino server on the iSeries server since it first became available natively on the iSeries. The SERVER task is a critical task that controls a lot of work happening on the Domino server. Because of this criticality in Domino server performance and functionality, the run-time priority of this task cannot be changed.

The ability to change the run-time priority of individual Domino tasks or jobs first became available with Version 4 Release 3 of OS/400 and Domino Release 5.0 for the iSeries. All later versions of OS/400 and Domino continue to support this capability.

Network performance tuning

Performance of the network can have a dramatic impact on end-user response times and CPU processing on the server. This section addresses some communication performance tuning items that often affect communication performance related to Domino environments.

UDP checksum

Communication performance is critical to any Domino environment because of the client/server nature of Lotus Notes client-based applications and Web browser-based Domino applications. One of the parameters associated with the TCP/IP attributes of a communications line description is UDP checksum. The default setting for the TCP/IP attributes is to enable UDP checksum validation. This is the correct setting if the communication is happening to another iSeries server or to other non-Microsoft Windows®-based servers. The performance issue comes into effect when the communication is between the iSeries server and a Windows-based client or server.

The issue revolves around the fact that the iSeries server uses the industry standard protocol for UDP checksum. Windows uses its own proprietary checksum algorithms. Because of this, there can be conflicts that need to be resolved when communicating between an iSeries server and a Windows-based system.

A *checksum* is a count of the number of bits transferred in a packet sent across the communication line. The checksum is sent with a packet to let the receiving end know how many bits were sent on the transmission. If the number of bits received does not match the checksum, then the packet needs to be retransmitted.

Note: Route settings and Domain Name System (DNS) configurations also play a part in this performance equation. Intelligent switches dynamically generate a route table versus using a static route table. If you have more than one NIC card on the iSeries server, you run the risk of generating an ARP storm if route tables are dynamically generated.

DNS settings also affect performance. You can obtain optimal performance with local DNS settings. However, this may not be possible with your required network configuration.

For more details about these two items, refer to the redbook *Domino 6 for iSeries Best Practices Guide*, SG24-6937.

Because iSeries and Windows use different checksum algorithms, sometimes the checksum information sent with the transmission is translated to a different number than it should be, thereby causing unnecessary retransmissions to occur. If you see a lot of retransmissions happening for a specific communication line used to access a Domino server, then we recommend that you turn off UDP checksum for the communication line involved. You can find the retransmission rates for a communication line by performing the following steps:

1. From an OS/400 command line, type the following Network Status (NETSTAT) CL command:

   ```
   netstat *cnn
   ```

2. Examine the number of connections associated with the communication line used by the Domino server. You do this by typing an option `5` (Display) next to these connections.
3. On the Display TCP Connection Status display, page down until you see the Retransmission information section.
4. Look at the Total retransmissions and Current retransmissions values. If there are more than a small number of retransmissions, we recommend that you disable the UDP checksum capability for the communication line.

   To make UDP checksum unavailable, use the Change TCP/IP Attributes (CHGTCPA) CL command and change the UDP checksum (UDPCKS) parameter to *NO, for example:

   ```
   CHGTCPA UDPCKS(*NO)
   ```

   Note: Keep in mind that all other services on the iSeries server using this communication line are impacted by this change. Some applications, such as independent auxiliary storage pools (IASP) require UDP checksum functionality. In this case, do not make UDP checksum unavailable for the communication line.

Connectivity between a Lotus Notes client and the Domino server on the iSeries server is the primary situation where UDP checksums occur. The Windows 2003 code uses the industry standard UDP checksum protocol. Therefore, this problem should become much less of an issue as desktops are upgraded to this version of Windows or later.

Maximum transmission unit size

The maximum transmission unit (MTU) size parameter affects the actual size of the line flows. By increasing the value of this parameter, you can reduce the overall number of transmissions and increase the potential capacity of the CPU and the IOP (input/output processor). Similar parameters also exist on the Web browser. The negotiated value is the minimum of the server and browser (and perhaps any bridges or routers), so increase them all. Be sure to set the MTU the same on the browser, client, router, and iSeries line description.

The recommended setting varies depending on the communications protocol that is used:

- 4Mb Token Ring = 4060
- 16 Mb Token Ring = 16388
- Ethernet 802.3 = 1492
- Ethernet version 2 = 1500

To change the MTU size, perform the following steps:

1. Enter the Configure TCP/IP (CFGTCP) command on an OS/400 command line and press Enter.
2. On the Configure TCP/IP display, select option 2 (Work with TCP/IP Routes) and press Enter.
3. On the Work with TCP/IP Routes display, in the Opt column next to the IP address used by your Domino server, type option 2 and press Enter.
4. On the Change TCP/IP Route display, enter the recommendation from the previous list in the MTU parameter and press Enter.

TCP/IP buffer size

You can increase Web serving performance by increasing the buffer size that is used by TCP/IP, especially when sending large amounts of data. Change the buffer size from the default (8000) to 64000. To change the buffer size, perform the following steps:

1. Enter the Configure TCP/IP (CFGTCP) command on an OS/400 command line and press Enter.
2. On the Configure TCP/IP display, select option 3 (Change TCP/IP Attributes) and press Enter.
3. On the Change TCP/IP Attributes display, locate the TCP receive buffer size (TCPRCVBUF) parameter and enter the new value of 64000. Locate the send buffer size (TCPSNDBUF) parameter and enter the new value of 64000. Press Enter.

Note: Use care when setting the TCP/IP buffer size on older networks. You also want to monitor the number of retransmissions taking place on a given line before you change this setting to the 64000 send and receive buffer size. If there are several retransmissions on a given line, we recommend that you do *not* set the size of the TCP/IP buffer size to 64000.

For details about optimally setting your TCP/IP environment for Domino in a multiple OS/400 logical partitions (LPAR) environment, see the redbook *Coexistence of Multiple Lotus Domino Releases in an LPAR Environment*, SG24-6593.

Packet filtering

Packet filtering happens on every communications line. Therefore, make sure that you have the proper hardware configuration to enable this support. In the past, when filtering was not configured correctly on a system, the result was a communications performance problem.

The primary concern has been with the 2838 Ethernet card. If you have this card, you need to check the part number and ensure that filtering is optimally configured.

- If the part number on the 2838 Ethernet card is 21H5458, it has built-in filtering. You should not have an external filter since double filtering can cause problems.
- If the part number is anything else on the 2838 Ethernet card, then an external filter is required. We recommend the external filter cable with part number 97H7385.

Duplex settings

The duplex settings are HALF and FULL. The best performance is with FULL duplex. However, the duplex setting on the line description must match the setting on the port on the switch, if the line is hooked up to a switch.

- If the line is connected directly to a 8271 Ethernet switch, the switch's port is FULL duplex. Therefore the duplex setting on the Ethernet line description should be set to FULL.
- If the line is connected to a *stackable* hub, then the duplex setting on the line should be set to HALF to match the hub setting.

Be aware that setting the duplex setting to AUTO may not have the results that you are expecting. For example, some CISCO switches have a default setting of AUTO. Using a FULL duplex setting in the line description when the switch is set to AUTO has resulted in the line sometimes being FULL duplex and sometimes being HALF duplex.

For more details, see the iSeries Service TechNote 6654625, *OS/400 Ethernet - 100 Mbps (FC2838 / FC2849), 1 Gigabit (FC2760 / FC5701 STP or UTP), or 1 Gigabit (FC2843 or FC5700 Fiber), and Full Duplex (CRTLINETH, CHGLINETH)*, on the Web at:

http://www.ibm.com/support/docview.wss?uid=nas1990a2cc52f1fb76186256665004727bf

Integrated file system attributes

All Domino databases are stored in the IFS of the iSeries server. A tuning parameter was added to stream files in OS/400 V5R2 that allows you to influence how Domino files are paged into memory.

A stream file, which is part of the iSeries IFS, may contain from 0 GB to 256 GB of data. Each stream file consists of one to many 16 MB logical segments, depending on the allocated size of the stream file. For example, a file with an allocated size of 32 MB requires two 16 MB segments. Each of these 16 MB segments must be individually created when a stream file is created or expanded beyond a 16 MB boundary. With OS/400 V5R2, when a stream file increases in size by more than one segment, the IFS does a more efficient job of creating these segments in parallel. This allows for overlap of I/O and CPU processing.

When a database that Domino needs to access is not already in memory, a page fault occurs. The data in the Domino database that is requested is paged into memory by default in 16 Kb block sizes. OS/400 V5R2 allows you to change the blocking factor used when paging Domino databases into memory.

This new capability is only available to *TYPE2 stream files. Most Domino servers are already using *TYPE2 stream files so this enhancement applies to most environments. You can use the Display Link (DSPLNK) CL command against a Domino server's data directory to see if the file format is *TYPE2, for example:

```
DSPLNK OBJ('/DOMINO/DOMSVR1/*') OUTPUT(*PRINT) OBJTYPE(*STMF) DETAIL(*EXTENDED)
DSPOPT(*ALL)
```

Here /DOMINO/DOMSVR1 is the data directory path of the Domino server.

The DSPLNK command generates a spooled file that you can examine. Look for the value specified for the File format parameter. If the file format is *TYPE2, you can take advantage of the new blocking size tuning parameter. If the file format is *TYPE1, use the new OS/400 V5R2 CL command, Convert Directory (CVTDIR), to convert the directory to a *TYPE2 file format.

The new blocking factor enhancement offers three different types of page fault blocking sizes:

- ***Normal**: This is the default and is best for low memory contention situations. Memory is paged in 16 Kb blocks.
- ***Minimize**: Use in high memory contention situations. Memory is paged in 8 Kb blocks.
- ***Dynamic**: Use when memory contention situations change throughout the day. If memory is plentiful, the block size used is 12 Kb. If memory is constrained, an 8 Kb block size is used.

Note: To fully take advantage of the *Dynamic paging option, the memory pool where Domino is running needs the paging option set to *CALC. If the memory is using the *FIXED paging option, then 12 Kb block sizes are used regardless of memory availability.

You can use these different blocking factors in combination with the previous tuning parameters, such as NSF buffer pool size and DB cache, to ensure that memory is used most optimally on the iSeries server.

To take advantage of this new tuning option, use the Change Attribute (CHGATR) CL command and specify the desired page block size transfer size by setting the value for the *MAINSTGOPT attribute, for example:

```
CHGATR OBJ('/LOTUS/DOMINO/DOMDEV01/MAIL/*') ATR(*MAINSTGOPT) VALUE(*DYNAMIC)
```

For more information, see the white paper *V5R2 Stream File Performance Enhancements for iSeries*, which you can find on the Web at:

http://www.ibm.com/servers/eserver/iseries/perfmgmt/pdf/ifsperf.pdf

Collection Services

Version 5 Release 2 (V5R2) of OS/400 in combination with Domino 6 offers the ability to collect Domino statistic data synchronously with OS/400 performance data. When you configure a new Domino 6.x server on the iSeries server or upgrade existing Domino servers to Domino 6.x, you see a new task listed in the ServerTasks= variable in the NOTES.INI file. This new variable is the Collection Services (COLSRV400) task.

Note: If you are running Domino 6.x on OS/400 Version 5 Release 1, the COLSRV400 task is not present. You can also find Domino statistics in the Statistics and Reporting (statrep.nsf) database.

This new task automatically collects Domino performance data when OS/400 performance data is collected. With this new performance data collection feature, all of the performance statistics are stored in one place, allowing you to not only analyze performance of OS/400, but to also analyze performance of the Domino servers in conjunction with the OS/400 data.

The performance data file that contains these Domino statistics is QAPMDOMINO. Not all of the Domino statistics are included in this file. Only the primary ones are included. For details about the Domino statistics reported in this file, see the iSeries Information Center at:

http://as400bks.rochester.ibm.com/iseries/v5r2/ic2924/index.htm?info/rzahx/rzahxqapmdomino.htm

The types of Domino statistics that are collected are a subset of the same types of statistics that you can collect through Domino statistics and events data collection or by issuing the

show stat command on the Domino server console. The limitations are found in the details of the QAPMDOMINO file contents linked to earlier. For details about the statistics collected by the COLSRV400 task, see the redbook *Domino 6 for iSeries Best Practices Guide,* SG24-6937.

The main benefit you gain from the COLSRV400 task is that the data is automatically collected with OS/400 performance data statistics. This makes it easier to determine what is happening on the whole iSeries server in relation to the Domino work that is also executing on the system.

Summary

If you are looking for additional information related to Domino for iSeries tuning tips, see the redbook *Domino for iSeries Sizing and Performance Tuning*, SG24-5162. This redbook contains additional information related to many of the performance tuning recommendations discussed in this Redpaper. It discusses several additional iSeries Domino performance tuning tips. It also contains information about how to perform iSeries Domino capacity planning.

Additional Domino performance resources include the Domino for iSeries performance Web page that contains links to capacity planning information, application design guidelines, and iSeries and Domino tuning tips. You can find this Web page at:

http://www.ibm.com/servers/enable/site/domino/perform.html

The Lotus performance Web page contains really good Domino cross-platform performance information. You can find this page on the Web at:

http://www.lotus.com/ldd/performance

The Lotus DeveloperWorks contains several articles that detail a variety of Domino performance topics. These topics range from how to debug semaphore lock problems that are causing performance issues to how to fine-tune HTTP Domino performance. Look for the Lotus DeveloperWorks on the Web at:

http://www.lotus.com/ldd

If you need to perform capacity planning for your Domino environment, you can use the IBM @server Workload Estimator tool. This tool allows you to do the appropriate capacity planning information for various environments on your iSeries server including Lotus Domino workloads. You can find this free sizing and capacity planning tool on the Web at:

http://www-912.ibm.com/wle/EstimatorServlet

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Rochester Center.

Kim Greene is president of Kim Greene Consulting, Inc. She has over five years of experience with Domino and 13 years of experience with the AS/400® and iSeries platform. She specializes in iSeries Domino performance analysis, system and application tuning, enterprise integration with back-end applications and data, and custom Domino development. Kim frequently writes for technology magazines and speaks at conferences worldwide. Previously she worked for IBM in Rochester, MN, in PartnerWorld® for Developers (PWD) helping IBM Business Partners incorporate Domino into their existing applications. She also worked on areas of OS/400 performance in the AS/400 development laboratory.

Dave Johnson is an advisory software engineer at IBM in Rochester, Minnesota. He works on iSeries System Performance team focusing on Domino and Lotus collaboration products. His 19-year career with IBM has been spent mostly in performance-related work. He also worked in OS/400 development writing code for the IFS.

Joe Peterson is a Senior Software Engineer on the iSeries development team in Rochester, MN. His focus is on the performance of Lotus products.

Thanks to the following people for their contributions to this project:

Debra Landon
ITSO, Rochester Center

Jennifer Ginder
Dan Hill
Gerri Passe
IBM Rochester Development team

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on June 30, 2004.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an Internet note to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. JLU Building 107-2
  3605 Highway 52N
  Rochester, Minnesota 55901-7829 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	IBM®	QuickPlace®
ibm.com®	Lotus Notes®	Redbooks(logo) ™
iNotes™	Lotus®	Redbooks™
iSeries™	Notes®	Sametime®
AS/400®	OS/400®	WebSphere®
Domino®	PartnerWorld®	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.