Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning

What's new in Domino 6

Performance tuning on z/OS systems

Performance management and capacity planning

Mike Ebbers
Wanda Brewster
Luis Leon Collart
Terry Draper
Chris Henry
Rufus Woody III

ibm.com/redbooks
First Edition (March 2003)

This edition applies to IBM Lotus Domino for z/OS V6.0 (product number 5655-K36).

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.
# Contents

**Notices** .......................................................... ix
**Trademarks** ......................................................... x

**Preface** .......................................................... xi
**The team that wrote this redbook** .................................. xi
**Become a published author** .......................................... xii
**Comments welcome** ................................................ xii

## Chapter 1. What’s new in Domino 6
1.1 The unique benefits of z/OS as a Domino platform ...................... 2
1.2 General recommendations .............................................. 2
1.3 What’s new in Lotus Domino Release 6 for z/OS ......................... 3
   1.3.1 Performance improvements ..................................... 3
   1.3.2 Usability improvements ....................................... 4
1.4 More to come ....................................................... 8
1.5 What’s New in Lotus Domino 6 and Lotus Notes 6 .......................... 8
   1.5.1 What’s new in Lotus Notes 6 for end users ..................... 9
   1.5.2 What’s new in Lotus Domino 6 for Domino administrators ....... 12
   1.5.3 What’s new in Lotus Domino 6 for Domino designers/developers ... 15
   1.5.4 What’s new about Lotus Domino 6 servers .................... 19
   1.5.5 Cross-platform summary ..................................... 22

## Chapter 2. z/OS tuning
2.1 Recommendations ................................................... 23
2.2 Planning the server architecture ...................................... 25
   2.2.1 z/OS address spaces ........................................... 25
   2.2.2 Multiple Domino servers ...................................... 28
   2.2.3 Domino on symmetric multiprocessors ......................... 28
   2.2.4 Setting z/OS parameters ...................................... 28
2.3 Managing the Domino workload ....................................... 30
   2.3.1 Workload Manager goal mode .................................. 30
2.4 XPLink ............................................................. 31
2.5 Storage for Domino servers ......................................... 31
   2.5.1 Virtual storage ................................................ 32
   2.5.2 Real storage .................................................. 32
   2.5.3 64-bit processor storage ..................................... 33
   2.5.4 Page datasets ................................................ 34
   2.5.5 Paging .......................................................... 34
   2.5.6 Place program module in shared storage and LPA ................. 34
2.6 PR/SM™ ........................................................... 36
2.7 Monitoring z/OS .................................................... 38
   2.7.1 RMF .......................................................... 38
   2.7.2 Monitoring z/OS with SMF .................................... 47
   2.7.3 Monitoring z/OS with Domino statistics ....................... 50

## Chapter 3. DASD and file system tuning
3.1 Recommendations ................................................... 55
3.2 DASD subsystem considerations ..................................... 56
   3.2.1 The Enterprise Storage Server ................................ 56
## Chapter 5. Database performance and scalability

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Indexing</td>
<td>106</td>
</tr>
<tr>
<td>5.1.1 Update and Updall tasks</td>
<td>106</td>
</tr>
<tr>
<td>5.1.2 Full text index</td>
<td>107</td>
</tr>
<tr>
<td>5.1.3 Number of indexing tasks</td>
<td>108</td>
</tr>
<tr>
<td>5.1.4 Indexing parameters</td>
<td>108</td>
</tr>
<tr>
<td>5.2 Replication</td>
<td>104</td>
</tr>
<tr>
<td>5.2.1 Replication between servers</td>
<td>104</td>
</tr>
<tr>
<td>5.2.2 Replication between a client and a server</td>
<td>105</td>
</tr>
<tr>
<td>5.3 Hub</td>
<td>106</td>
</tr>
<tr>
<td>5.4 Soft deletes</td>
<td>98</td>
</tr>
<tr>
<td>5.5 Enabling quota enforcement</td>
<td>99</td>
</tr>
<tr>
<td>5.5.1 Full text index</td>
<td>107</td>
</tr>
<tr>
<td>5.5.2 Update and Updall tasks</td>
<td>106</td>
</tr>
<tr>
<td>5.5.3 Full text index</td>
<td>107</td>
</tr>
<tr>
<td>5.5.4 Soft deletes</td>
<td>98</td>
</tr>
<tr>
<td>5.5.5 Enabling quota enforcement</td>
<td>99</td>
</tr>
<tr>
<td>5.5.6 Restoring databases</td>
<td>100</td>
</tr>
<tr>
<td>5.5.7 Test results of transaction logging</td>
<td>101</td>
</tr>
<tr>
<td>5.5.8 Recommendation</td>
<td>101</td>
</tr>
<tr>
<td>5.6 Domino server tasks</td>
<td>102</td>
</tr>
<tr>
<td>5.6.1 Domino tasks started</td>
<td>103</td>
</tr>
<tr>
<td>5.7 Replication</td>
<td>104</td>
</tr>
<tr>
<td>5.7.1 Replication between servers</td>
<td>104</td>
</tr>
<tr>
<td>5.7.2 Replication between a client and a server</td>
<td>105</td>
</tr>
<tr>
<td>5.8 Indexing</td>
<td>106</td>
</tr>
<tr>
<td>5.8.1 Update and Updall tasks</td>
<td>106</td>
</tr>
<tr>
<td>5.8.2 Full text index</td>
<td>107</td>
</tr>
<tr>
<td>5.8.3 Number of indexing tasks</td>
<td>108</td>
</tr>
<tr>
<td>5.8.4 Indexing parameters</td>
<td>108</td>
</tr>
<tr>
<td>5.9 Compact</td>
<td>108</td>
</tr>
<tr>
<td>5.10 Unread marks</td>
<td>109</td>
</tr>
<tr>
<td>5.11 Multiple MAIL.BOX databases</td>
<td>110</td>
</tr>
<tr>
<td>5.12 Web server</td>
<td>110</td>
</tr>
<tr>
<td>5.12.1 Domino Web server</td>
<td>111</td>
</tr>
<tr>
<td>5.12.2 Domino Web server parameters</td>
<td>111</td>
</tr>
</tbody>
</table>

## Chapter 6. Domino configuration advanced options

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Partitioning</td>
<td>114</td>
</tr>
<tr>
<td>6.1.1 Benefits of partitioning</td>
<td>114</td>
</tr>
<tr>
<td>6.1.2 Costs of partitioning</td>
<td>115</td>
</tr>
<tr>
<td>6.1.3 Test results with multiple partitioned Domino servers</td>
<td>115</td>
</tr>
<tr>
<td>6.1.4 Test result with four Domino partitioned servers</td>
<td>116</td>
</tr>
<tr>
<td>6.1.5 Partitioning recommendations</td>
<td>118</td>
</tr>
<tr>
<td>6.2 Clustering</td>
<td>119</td>
</tr>
<tr>
<td>6.2.1 Cluster components</td>
<td>120</td>
</tr>
<tr>
<td>6.2.2 Parameters that affect clustering</td>
<td>120</td>
</tr>
<tr>
<td>6.2.3 Costs of clustering</td>
<td>120</td>
</tr>
<tr>
<td>6.2.4 Monitoring clustering</td>
<td>121</td>
</tr>
<tr>
<td>6.2.5 Effectiveness of Domino clustering</td>
<td>121</td>
</tr>
<tr>
<td>6.2.6 Alternatives to Domino clustering</td>
<td>122</td>
</tr>
<tr>
<td>6.3 Hubs</td>
<td>124</td>
</tr>
</tbody>
</table>

## Chapter 7. Domino monitoring

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 SMF type 108 record</td>
<td>127</td>
</tr>
<tr>
<td>7.1.1 Record mapping</td>
<td>128</td>
</tr>
<tr>
<td>7.1.2 RMF Lotus Domino server report</td>
<td>128</td>
</tr>
<tr>
<td>7.2 Other monitoring tools</td>
<td>142</td>
</tr>
<tr>
<td>7.2.1 The domps command</td>
<td>142</td>
</tr>
<tr>
<td>7.2.2 Notes log</td>
<td>142</td>
</tr>
<tr>
<td>7.2.3 Statistics and events log</td>
<td>143</td>
</tr>
<tr>
<td>7.2.4 Statistics and reporting database</td>
<td>143</td>
</tr>
<tr>
<td>7.2.5 Processing Domino statistics</td>
<td>144</td>
</tr>
</tbody>
</table>

## Chapter 8. Capacity planning and performance management

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Recommendations</td>
<td>146</td>
</tr>
<tr>
<td>8.2 Estimating processor capacity</td>
<td>147</td>
</tr>
<tr>
<td>8.2.1 Factors influencing the processor capacity needed</td>
<td>147</td>
</tr>
</tbody>
</table>

Contents V
8.2.2 Which user measurement? ......................................................... 147
8.2.3 Registered users ................................................................. 148
8.2.4 Connected users ............................................................... 148
8.2.5 Active users ................................................................. 148
8.2.6 Domino transaction rate .................................................... 148
8.2.7 CPU time used per transaction .............................................. 149
8.2.8 Consider the peak times ..................................................... 149
8.2.9 Making an initial estimate .................................................... 150
8.2.10 Capacity planning .......................................................... 150
8.3 Estimating processor storage .................................................. 150
  8.3.1 Estimating processor storage for Notes clients ....................... 151
  8.3.2 Estimating processor storage for POP3 clients ....................... 152
  8.3.3 Estimating storage for IMAP clients .................................... 153
  8.3.4 Estimating storage for iNotes Web Access clients ................. 154
8.4 The use of Domino partitioned servers ...................................... 155
8.5 The use of zSeries LPARs ...................................................... 156
8.6 Estimating DASD required ...................................................... 157
  8.6.1 Estimating DASD space .................................................... 157
  8.6.2 Estimating DASD I/O rates ................................................. 157
8.7 Network considerations ........................................................ 157
8.8 Capacity planning assistance .................................................. 158
  8.8.1 TechXpress ................................................................. 158
  8.8.2 SNAP/SHOT® ............................................................... 159
  8.8.3 IBM Testing Services ...................................................... 159
8.9 Performance management versus capacity planning .................... 159
  8.9.1 What is required for capacity planning .............................. 160
  8.9.2 What is required for performance management ..................... 160

Appendix A. Workload Manager (WLM) example .................................. 165
  Workload and service class descriptions ....................................... 165
  Report classes .................................................................. 166
  Classification rules ............................................................. 167
  Service definition goals for service classes .................................... 168

Appendix B. Using the domps command .............................................. 171
  B.1 Parameters ................................................................ 171
  B.2 Displaying virtual storage .................................................. 171
  B.3 Output examples ................................................................ 172
    B.3.1 Sample output for domps .............................................. 172
    B.3.2 Sample output for domps with -p .................................. 173
    B.3.3 Sample output for domps with -t .................................. 174

Appendix C. Sample notes.ini file ....................................................... 177

Appendix D. Sample output of the show statistics command .................. 181

Appendix E. Domino modify zFS command ........................................... 193

Related publications ................................................................. 199
  IBM Redbooks .................................................................. 199
  Other resources .................................................................. 199
  Referenced Web sites ............................................................ 200
  How to get IBM Redbooks ..................................................... 200
  IBM Redbooks collections ..................................................... 200
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.
Trademarks

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

- Redbooks™
- IBM®
- 1-2-3®
- CICS®
- DB2®
- Domino Designer®
- Domino™
- Enterprise Storage Server™
- ESCON®
- eServer™
- IMS™
- iNotes™
- Lotus Enterprise Integrator™
- Lotus Notes®
- Lotus®
- MVS™
- Notes®
- OfficeVision/MVS™
- OfficeVision/VM™
- OfficeVision®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM™
- RACF®
- Redbooks™
- RMF™
- S/390®
- Sametime®
- StorWatch™
- SNAP/SHOT®
- Tivoli®
- WebSphere®
- Word Pro®
- z/OS™
- zSeries™

The following terms are trademarks of other companies:

- ActionMedia, LANDesk, MMX, Pentium and ProShare 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.

- C-bus is a trademark of Corollary, 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.
Preface

This IBM® Redbook helps z/OS™ system programmers and Domino™ administrators to monitor and tune Domino 6 for z/OS. It identifies factors that affect performance and scalability, and provides recommendations to tune the configuration and parameter settings for optimal performance. Tips are provided for the operating system, the disk I/O subsystem, the network, and the Domino product. In addition, it includes guidelines for capacity planning. A background in z/OS and experience with Domino is assumed.

This version, updated for Domino 6, includes tips on the zFS file system and new monitoring tools and techniques.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Mike Ebbers is a senior IT specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on Lotus products. Mike has been with IBM for 29 years. Before joining the ITSO eight years ago, he worked for 11 years in technical marketing and 10 years in marketing/technical education.

Wanda Brewster is a consulting IT specialist with the Domino for zSeries team in the Washington Systems Center. She has worked at IBM for 25 years. Her areas of expertise include Domino administration, and migration from and coexistence with the legacy office systems OfficeVision®/VM™, OfficeVision/MVS™, and DISOSS.

Luis Leon Collart is a zSeries™ IT specialist located in Brasilia, Brazil. He has six years of experience in Domino, and for four years he has worked with the zSeries platform and supported several large customers who run Domino. His areas of expertise include WebSphere® and Linux on zSeries.

Terry Draper is a zSeries performance specialist in the UK. He has worked at IBM for 18 years, as a processor specialist, storage specialist, and subsystem specialist including CICS®. Prior to joining IBM Terry had 12 years of experience in application and systems programming. His areas of expertise include performance for processors, and z/OS and its subsystems. Recently he has concentrated on Domino performance.

Chris Henry is a senior IT specialist from Milwaukee working in Field Support Services. He has been at IBM for 8 years, joining as part of the Lotus® acquisition. Before joining Field Support Services, he supported the SoftSwitch product line in the field, as well as Lotus Notes®/Domino. He has had extensive experience with most messaging solutions on the market, including Exchange, Groupwise, cc:Mail, OfficeVision, and SMTP messaging. His areas of expertise include Domino installation, messaging, and Domino transaction logging. Chris has presented extensively on Domino transaction logging and backup solutions.

Rufus Woody III is a Consulting IT specialist with the Domino for zSeries team in the Washington Systems Center, located in Dallas, Texas. He has worked for IBM for more than 25 years and has more than 10 years experience with Domino and Notes. He is also the lead IBM representative to the Domino/Notes Project at SHARE (see http://www.share.org for more information). His previous assignments include more than 10 years supporting the MVS office arena, including being IBM's representative to SHARE's Office MVS Project.
Thanks to the following people for their contributions to this project:

Richard Conway  
Robert Haimowitz  
Margaret Hunt  
Mike Schwartz  
Bill White  
International Technical Support Organization, Poughkeepsie Center

Joe Bostian  
Clark Goodrich  
Joe Sweeney  
Lotus Development, Poughkeepsie

Jim Ray  
IBM Server Group

Sreehari Haridevara  
Elsie Ramos  
IBM Global Services, Poughkeepsie

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:  
ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks 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. HYJ Mail Station P-099  
  2455 South Road  
  Poughkeepsie, NY 12601-5400
What’s new in Domino 6

This book provides information to help you tune the performance of Lotus Domino Release 6 for z/OS. It also assists you in planning the capacity for a Domino workload on an IBM @server zSeries (or S/390) platform.

**Terminology note:** In this redbook, we often use the term *Domino 6 for z/OS*. Likewise, when there is no ambiguity about a reference to both zSeries and S/390, we use the inclusive term *zSeries* without adding the qualifier “(or S/390)”. When there is ambiguity or when the reference is only to the newer or older hardware platform, respectively, we qualify the item specifically (“zSeries (only)” or “S/390 (only)”).

Domino 6 for z/OS runs on z/OS V1R2 or later or z/OSe V1R3 or later.

We first review what is new for Domino 6 for z/OS, and features that are different from Lotus Domino 6 on other server platforms. We then discuss the new features which are exclusive to Domino 6 for z/OS, and the significant capacity and performance enhancements in Domino 6 for z/OS. This allows you to see the advantages of upgrading to Domino 6 for z/OS as soon as practical.

Then we provide some general recommendations for performance measurement and tuning that will help get you started in managing your Domino 6 for z/OS environment effectively. This helps maximize your zSeries investment and lowers your Total Cost of Ownership (TCO).

The rest of this chapter lists the new features of Lotus Domino 6 and Lotus Notes 6. There’s something in here for everyone—end users, administrators, and designers/developers—and there are also new things about the Domino 6 server, too. Knowing what’s new will help you understand why you might want to upgrade from previous releases of Domino and Notes.
1.1 The unique benefits of z/OS as a Domino platform

While a Domino server on z/OS appears to Notes users as just another server, the zSeries server platform provides a lot more function than other servers. The zSeries platform, along with its z/OS operating system, has a long history in the commercial marketplace. It has matured into a powerful server platform that meets the needs of users of commercial systems.

Compare that to the history of UNIX in the commercial marketplace, as well as the PC operating systems, which are relatively young. Key functions, such as clustering (for scalability and availability) and systems management tools, which are important in commercial systems, have been available on zSeries and S/390 for many years and are more mature than on many other servers. Users of zSeries and S/390 recognize and value that maturity.

A zSeries server is able to support large workloads because IBM has invested in many capabilities in the processor and software to provide scalability. Among the key features and benefits of zSeries server are the following:

- Very efficient use of symmetric multiprocessors (SMPs), even up to 16-engine machines
- Addressing of terabytes of expanded processor memory and zSeries support of 64 GB of central memory
- Efficient techniques for keeping data in processor memory where it can be accessed quickly
- A sophisticated job scheduler
- The removal of bottlenecks
- Shared access to all disks by all processors
- Separate, dedicated I/O processors
- Efficient automated procedures for backing up terabytes of data
- XPLink reduces compiler CPU usage by 25-30%. This saves 15-20% in CPU usage over Domino R5.

1.2 General recommendations

As you approach the topic of Domino performance, it is useful to remember the following:

Domino was originally written for the PC platform. Therefore, it was developed with different requirements from those for traditional zSeries applications, such as CICS, IMS™, and DB2®. One of our objectives is to explain the characteristics of Domino 6 for z/OS so that you are better able to monitor and tune the system. Another objective is to help you understand which aspects of Domino 6 for z/OS affect the capacity/sizing characteristics of the zSeries servers you will implement on an ongoing basis.

Domino was written to be platform-independent. This means that it does not exploit all z/OS functions directly. Contrast that with CICS, IMS, and DB2, which were all written for z/OS and designed specifically to exploit capabilities of zSeries hardware. Domino was ported to z/OS UNIX System Services, so its features and functions are compatible with other supported platforms.

However, Domino 6 was modified to take advantage of certain z/OS-specific features for increased performance. In addition, numerous changes were made to z/OS specifically to
improve Domino 6 scalability. Extensive testing shows that there is a significant improvement in performance on the zSeries platform (as well as on other platforms).

This redbook guides you in performance considerations for z/OS UNIX System Services needed for managing the performance of Domino 6 for z/OS. This redbook assumes knowledge of Domino 6 for z/OS. This knowledge can be gained by reading *Lotus Domino 6 for z/OS: z/OS Installation Guide for Lotus Domino*, which comes with the product.

**Key performance recommendations**

Our key recommendations for performance tuning, performance monitoring, and capacity planning of Lotus Domino Release 6 for z/OS are:

- Monitor z/OS closely as you introduce Domino. Domino does not run the same way as other zSeries workloads that you may be used to. Make sure that you are not artificially constraining Domino with parameter settings designed for a different type of application.

- Use a team approach. Domino tuning on zSeries is a combination of:
  - z/OS tuning, including z/OS UNIX System Services tuning
  - Domino server tuning
  - DASD tuning
  - Network tuning

Therefore, the Domino administrator, the z/OS and z/OS UNIX System Services performance specialists, the DASD administrator, and the network administrator must work together to optimize Domino performance.

- Consult our list of tuning recommendations at the start of each chapter in this book.

- Use valid measurements for capacity planning.

We discuss some valid capacity planning approaches in Chapter 8, “Capacity planning and performance management” on page 145. Avoid using simple single-function tests as an indicator of the capacity of zSeries as a Domino server. These tests are easy to run, but they can give very misleading results. It is one purpose of this redbook to help you understand what tests generally are valid, what to measure, and how to interpret the measurements. That way you won’t end up coming to invalid conclusions about the true power of Lotus Domino Release 6 for z/OS on the zSeries hardware platform.

### 1.3 What’s new in Lotus Domino Release 6 for z/OS

There are several improvements in Domino 6 for z/OS which are not generic in nature (that is, they are not available on other server hardware platforms). This section discusses enhancements to Domino 6 for z/OS that are specific to that environment.

#### 1.3.1 Performance improvements

Perhaps the most significant enhancement is the performance improvements available in Domino 6 for z/OS. Much of this is due to improvements in XPLink, the rewriting of the Domino 6 HTTP engine, and better I/O performance when using zFS. In performance tests at IBM Poughkeepsie, Lotus software developers have seen Domino 6 for z/OS CPU reductions of 25 to 30 percent for Notes 6 clients and up to 45 percent for iNotes Web Access 6 clients. In addition, they have seen up to 65 percent reductions in I/O when using zFS, compared to Domino R5 for S/390, Notes R5, iNotes Web Access R5, and HFS. These results are a
powerful motivator for upgrading to Notes 6 and Domino 6 for z/OS. However, just upgrading the server to release 6 will show significant CPU savings over R5.

Domino 6.5 for z/OS includes support for HTTP exploiting hardware cryptography (System SSL) for z/OS. See the Domino z/OS Release Notes for instructions.

CPU reduction and z/OS improvements
Domino 6 for z/OS has shown significant CPU reduction for Domino’s mail and calendaring & scheduling functionality in the development lab for both Notes 6 and iNotes Web Access 6 clients. This is mainly due to:

- C/C++ compiler and LE/RTL support of XPLink, which significantly reduces the number of instructions executed between C/C++ function calls
- C/C++ compiler support for new S/390 G5 and G6 machine instructions
- Improved optimization for and reduction in the number of instructions generated by the C/C++ compiler
- Using a new C/C++ compiler option to increase function inlining for C++ object-oriented code in Domino 6
- C/C++ compiler support of the standard template library used by Domino 6

New zFS latch structure
In z/OS V1R3, a new latch structure has been implemented for the zFS file system. This new latch structure significantly increases the number of concurrent accesses that can be made in the same file system or mount point. This is a big advantage when many users are concurrently accessing shared Domino files, such as names.nsf and other application databases. To take advantage of this improved performance, you must migrate to z/OS V1R3 or later.

1.3.2 Usability improvements
Domino 6 also contains new functions and features that make it easier to install and use.

Documentation improvements
The installation documentation has been extensively rewritten, including the addition of flow charts of the installation process and screen captures of the new Domino server setup procedure. It is available in two formats: Adobe Portable Document Format (PDF), available by download, and as a Domino database (in .nsf format) available with the product as well as by download. The PDF document, *Lotus Domino 6 for z/OS: z/OS Installation Guide for Lotus Domino*, is available from Lotus Developer Domain (LDD, formerly known as “Notes Net”). This documentation is also delivered with Domino 6 for z/OS as a database with a default title of *Domino 6 for z/OS Install Guide*. Other than the mildly confusing difference in titles, a graphical cover page, and numbered pages in the PDF format of the installation guide, the contents are identical. To download either of these formats prior to installing Domino 6 for z/OS, see the URL:

http://www.lotus.com/ldd

Next, select the Documentation Library link in the navigator at the left of the window, then select by product, then select Domino for z/OS, and finally, select 6.0 (for the initial release). At this point, you should be able to download either format you desire. Note that the Domino database (.nsf) format requires a Notes 6 client to use.
The release notes for Domino 6 for z/OS, formerly a separate document, have now been merged with core Lotus Notes 6: Release Notes as a chapter titled “Domino for z/OS”.

Lotus Domino 6 for z/OS: z/OS Console Support for Domino has been rewritten. Like the installation documentation, it comes with the product (in .nsf format) and is available for download from LDD in both formats (PDF and .nsf).

**Terminology note:** In this redbook, we use title of the database which is shipped with the product, Domino 6 for z/OS Install Guide, when referring to the installation documentation. Similarly, we use the abbreviated name Release Notes rather than the longer title. Likewise, we use the title of the database which is shipped with the product, z/OS Console Support for Domino, when referring to the new documentation for this feature.

**Domino server setup**
The new Java-based Domino server setup program replaces the formerly-used `httpsetup`, as it does on other platforms. One key enhancement available by using this new program is that server setup now provides appropriate ACL entries that allow administrator access to key system databases, which was not formerly done when using `httpsetup`. This new program uses TCP/IP port 8585 by default, but the port number can be customized at setup time. The previous `httpsetup` used port 8081.

**Java runtime environment now ships with Domino 6 for z/OS**
In Domino 6 for z/OS, the Java runtime environment is now embedded in the Domino executables tree. No additional setup is required to use it. There are no references to the Java runtime environment that use the PATH, CLASSPATH, or LIBPATH environment variables, all references to which will be ignored. No external JDK or SDK is required to use the Java support within Domino 6 for z/OS.

Because the Java version used by Domino 6 for z/OS is Java 1.3.1, you may see minor performance enhancements if you use Java applications on your Domino 6 for z/OS server.

**New BPXPRM verification function**
A new `dom_verify_os` command was introduced in Domino 6 for z/OS. This command is part of the installation procedure and helps you verify that BPXPRMxx values are set correctly.

**New fault recovery and serviceability features**
In the event of a failure, a Domino server (DPAR) can be configured to capture the z/OS state in an SVC dump, to capture the Domino state in an NSD dump, and to automatically restart the Domino server. The NSD dump, which runs as a script called `nsd`, captures significantly more service information with a program called `memcheck`. Server (DPAR) downtime can be significantly reduced and serviceability significantly improved, especially when combined with the improved server restart time when using the enhanced transaction logging and other features engineered into Domino 6.

A new environment variable, `OS390_DOMINO_SVC_DUMP=1`, enables SVC dumps for Domino failures with `ABEND=S0EC6,REASON=07020582` in BPXGXQCT. This new feature makes it possible to capture Domino SVC dumps with no process termination.

Depending on the existing z/OS configuration (if S0EC6 abends are disabled), a SLIP trap may need to be enabled for `ABEND=S0EC6,REASON=07020582` in order to obtain SVC dumps for diagnostic purposes. See the Domino z/OS Installation Guide for more information.
HTTP server task improvements
The HTTP task has been completely re-written in Domino 6 for z/OS. It is more stable and performs significantly better. Because of these changes, Web browser users should use less CPU when compared with the previous release of Domino. In addition, two other HTTP enhancements follow: the WebSphere plug-in and the DSAPI filter.

WebSphere plug-in
The Domino for IBM HTTP Server (Web Connector) feature of Domino R5 for S/390 is no longer supported for use with Domino 6 for z/OS. WebSphere to Domino connectivity in the zSeries environment is now achieved through the use of a general-purpose WebSphere plug-in. This plug-in allows you to configure the IBM HTTP Server to pass selected URL requests (such as URLs that reference Domino databases with an .nsf file extension) to Domino 6 for z/OS (or to any remote Web application server, if appropriately configured). Other requests are still handled by the IBM HTTP Server as usual.

As a result of using this general-purpose plug-in, restrictions which previously applied to the Web Connector no longer apply. In particular, use of iNotes Web Access 6 is now supported when using this new plug-in.

This general-purpose plug-in, which was initially developed by Domino 6 for z/OS development in Poughkeepsie, actually ships with WebSphere. For more details on this subject, see Lotus Domino 6 for z/OS: z/OS Installation Guide for Lotus Domino.

Domino HTTP DSAPI filter for host (RACF) authentication (optional)
The Domino for IBM HTTP Server (Web Connector) feature of Domino R5 for S/390 was able to authenticate browser users using RACF (or equivalent), and map an authenticated user with the corresponding Domino user in response to a challenge issued by the Domino HTTP server task. The Web Connector is no longer supported for use with Domino 6 for z/OS, but this functionality is replaced by a DSAPI filter provided with Domino 6 for z/OS.

When this optional Domino configuration is in effect, any credentials supplied on an authentication challenge (for example, name and password, or X.509 certificate) will be checked by host authentication services (RACF® or equivalent).

This functions exactly like it previously did with Web Connector, but support is now broadened to apply to any Web browser client connecting to Domino’s HTTP server task—not just those connecting through WebSphere.

For more details on this subject, see Lotus Domino 6 for z/OS: z/OS Installation Guide for Lotus Domino.

Server activity and performance monitoring improvements
After server installation, setup, and configuration is complete, the next step on the road to a successful deployment is to monitor the performance of your server and of z/OS. Domino has always been well integrated with the performance monitoring infrastructure of z/OS through SMF type 108 records. Performance-critical Domino information can thus be gathered via the traditional SMF interfaces and analyzed in concert with other RMF data about z/OS and your hardware environment to help provide an integrated picture of Domino server and your hardware platform performance. While this process works well for system programmers who have z/OS performance skills, most Domino administrators find this interface a significant hindrance to their efforts to provide a well-tuned Domino system environment.

In Domino 6 for z/OS, this potential obstacle is overcome by making z/OS performance metrics available to the Domino administrator via new z/OS platform statistics. RMF Monitor II data for CPU and central storage usage is now recorded every minute and can be displayed...
as Domino platform statistics in the same manner as other Domino statistics are maintained. The Domino administrator can display these statistics at the Domino server console or from the appropriate Domino statistics database, both of which are accessible using the Domino Administrator 6 client.

When these enhancements are combined with the (optional, separately-licensed) IBM Tivoli Analyzer for Lotus Domino, it is possible to collect powerful performance assessments, based on current performance plus historical collections of Domino statistics (including z/OS platform statistics).

**Multiple Domino 6 versions in same LPAR**

It is now possible to run multiple instances of varying levels of Domino 6 for z/OS partitions (DPARs) in the same LPAR, plus one release of Domino R5 for S/390 server (DPAR). This is accomplished by making use of Shared Library Support (see next topic).

Some explanation of this feature may be helpful. At the time this redbook is being written, only Domino 6.0.0 for z/OS is available, but maintenance releases are planned on a regular basis. When those maintenance releases become available, it will be possible to run Domino 6.0.0 DPARs, Domino 6.0.1 DPARs, Domino 6.0.x DPARs, and Domino R5.0.11 (for example) DPARs in a single LPAR. This new capability of Domino 6 for z/OS may eliminate the need for multiple test LPARs and/or DPARs, enabling you to save time, effort, and expense in managing your overall Domino infrastructure.

**Note:** This hypothetical example is meant to illustrate the new capability of running multiple release levels of Domino 6 DPARs and a single release level of Domino R5 DPARs in the same LPAR—it is *not* a practical statement about the capacity/performance characteristics of those DPARs in any specific LPAR. This example illustrates the removal of a previously-existing software limitation and is not intended to be a practical statement of the capacity/performance characteristics of any real hardware environment.

**Shared Library Support**

Shared Library Support simplifies the installation process by removing the former requirement of adding Domino DLLs into Dynamic LPA and the requirement to run the PUTINLPA job.

Shared Library Support is one thing that makes possible the use of multiple Domino 6 for z/OS levels in one LPAR.

**Verity KeyView filter support**

Domino 6 for z/OS now uses the Verity KeyView filter for retrieving text from binary attachments, which was previously only available on other hardware platforms. The types of binary attachments which can be indexed include Adobe PDF format, Word, WordPerfect, 1-2-3®, Excel, Freelance, PowerPoint, HTML, and many others (but not Word Pro® attachments).

To enable this support, a database has to be full text indexed with the option to index binary attachments. **(Note:** Indexing attachments as raw text continues as before and does not use the Verity KeyView filter support.)

There is a `notes.ini` setting that allows an administrator to enable or disable KeyView filter support for the entire server. This `notes.ini` setting thus allows administrators to turn this filter option off for all databases which have the full text indexing binary option turned on without having to change each database manually.
z/OS Console Support for Domino

New function for z/OS Console Support for Domino 6 (commonly referred to as “DomCon”) includes the ability to monitor and administer Domino servers at different release levels. This enhancement corresponds with the new ability to run DPARs with multiple versions of Domino 6 for z/OS and DPARs with a single version of Domino R5 for S/390 in the same LPAR. This is accomplished through new executables, a revised setup script, and updated JCL procedures.

Summary

In summary, Domino 6 for z/OS provides many significant enhancements, including:

- Improved scalability with lower CPU utilization
- Significant documentation, installability, and functional enhancements
- Better integration with WebSphere
- Improved capability for Domino administrators to monitor CPU and memory utilization using the new z/OS platform statistics

Each of these can lead to a lower Total Cost of Ownership (TCO) for your organization.

1.4 More to come

There are performance and capacity improvements planned for:

- Lotus Domino
- z/OS and its z/OS UNIX System Services interface
- zSeries File System
- zSeries hardware

Until those improvements occur, we hope that this redbook will assist you to better tune, plan for, and manage your Domino servers on zSeries hardware.

Note: We recommend other redbooks that contain information on tuning. They were written using Domino R5, but many of their suggestions are helpful for Domino 6.

- Lotus Domino for S/390® Release 5: Installation, Customization and Administration, SG24-2083. This redbook tells how to install Domino using recommended parameter values and environment setup conditions.
- Lotus Domino for S/390 Release 5: Running a Large Domino System, SG24-5984. This redbook contains real-world tips from customers on how to plan for, install, and administer large Domino servers in a zSeries environment.
- Lotus Domino for S/390 Release 5: Performance Tuning and Capacity Planning, SG24-5149. If you are running a Domino R5 server, many tips from this redbook are still valid.

1.5 What’s New in Lotus Domino 6 and Lotus Notes 6

In the following sections, we list the new features and functions of Domino 6 that are available to all supported server and client platforms.
Before you can take advantage of all the features in Domino 6 and Notes 6, and other associated clients, you must upgrade to them. Before you upgrade, it would be helpful for you to understand some key features of these new releases. This section will review what is new for end users in Notes 6, improvements for administrators in Domino Administrator 6, enhancements for designers/developers in Domino Designer 6, and what's new about the Domino 6 server for everyone.

1.5.1 What’s new in Lotus Notes 6 for end users

There are numerous new features in Notes 6 for end users, many of which emphasize improved usability in addition to new functionality.

**General usability enhancements**

- New login dialog box that allows you to choose your ID and location at the same time you enter your password
- Enhanced welcome page with a new customization wizard, an option to display the Tip of the Day, increased content types, and dynamic frame switching with options for a preview pane
- Quick Notes, a new function that allows you to create items (such as a brief memo, a new contact entry, a quick reminder, or a new journal entry) directly from the welcome page without actually having to open the database containing that type of item
- Improved bookmark support, including the ability to sort bookmark folders and providing two new folders called Startup and History:
  - The Startup folder allows you to start specified bookmarks, Web pages, databases, and other programs to be automatically opened at Notes startup. (Databases which are bookmarked have a window tab created for them but don’t actually open until the window tab is clicked in order to optimize performance).
  - The History folder contains bookmarks for anything you have opened today and in the last seven days, including documents, views, databases, and Web pages.
- Improved window tab support, including easy access to open windows, improved window management support, and the ability to drag-and-drop window tabs to reorder them
- Resizable dialog boxes for many functions (Select Addresses, Move Document to Folder, Open Database, and so forth) with the resized size being “sticky” (that is, staying the same size the next time you open them)
- Improved editing functions, including editing selected field types directly from views, the ability to edit attachments in-place (meaning without re-attaching the item after editing outside of Notes), easily creating a table from a (subset of a) view via copy-and-paste, and significant enhancements in table creation styles and editing options
- Powerful document locking and unlocking capabilities, which allow you exclusive update control (while still allowing others to view the document), resulting in decreased save and replication conflicts when collaborating with others
- New toolbars that replace SmartIcons and a user-customizable status bar
- Customizable logout screen, which is invoked when the user locks their ID or which is invoked automatically after a user-specified period of inactivity

**Mail improvements**

- Unread mark counts in mail folders
- Tree view of mail folders which is expandable/collapsible (like other file managers)
- Customizable color coding of entries by recipient in inbox and mail folders
Auto-refresh of the user’s inbox after checking for new mail without further user action
New reply and forward options to automatically eliminate attachments at the user’s option
New mail message checkboxes that allow users to select popular options without having to open the Delivery Options dialog box
Drag-and-drop of attachments to the file system
Deleting items moves them to the trash folder immediately without requiring the user to refresh the view
Option to use Internet-style replies and forwards
Updated attachments menu, properties box, and right-mouse menu with commonly-used terms like Save (instead of Detach), and so forth

Document improvements
Distributed document locking to avoid replication or save conflicts (with appropriate manual override controls available only to database managers)
Several tabbed table creation enhancements, including colored tabs and new tabs placement options
Easily create a table with row captions
Copy selected items in a view as a table
Cut and paste an entire table of data
Easily move the contents in a table row up or down using keyboard shortcuts
Place a border around a selected paragraph
Option in spell checking to ignore words containing numbers or all uppercase letters
Rearrange a list by easily moving items up or down
Hide paragraphs from mobile devices
Button enhancements, including colored buttons, improved button edges, and control over button widths
Support for multiple languages within a document and the ability to identify and mark those languages

Address Book improvements
New ability to print contact entries as address labels
Merger of Advanced views into main Address Book view to make them easier to find, but making the default view collapsed so it doesn't “interfere” for users who don't need the advanced views

Calendaring and Scheduling improvements
New calendar “mini-view” that lets you see all entries requiring your action in a single location while viewing your calendar
More calendar tools on one form, including the ability to see the availability of invitees in a collapsible section
Support for scheduling Sametime® online meetings from the calendar meeting form (and automatically re-scheduling the Sametime meeting if there are changes to the meeting which encapsulates the Sametime meeting)
New tabbed format to allow easier switching between daily, weekly, and monthly views
Chapter 1. What's new in Domino 6

- Improved work week support, including support for non-contiguous days in a work week, and support for two-week and one-month work views
- In-view creation and editing of appointments from any calendar view without opening the appointment form
- Improved meeting rescheduling with automatic notification options for the meeting chairperson
- Ability to view and print the status of meeting participant information
- Side-by-side placement of conflicts in calendar grids
- Color coding selected calendar entries by type
- Vastly improved time zone support, including the ability to show two different time zones in selected calendar views (for example, remote time and local time for a meeting that has participants in more than one time zone)
- The ability to schedule meetings that run past midnight
- Significantly improved calendar delegation, including the ability for a principal to assign full calendar management to someone else (regularly, or on an as-needed basis)
- iCal support (Internet-standard calendaring and scheduling via SMTP) and vCal (electronic business card) import
- Enhanced printing options, including the option to print to a Notes document
- New scrollable date picker to change months and years
- New scroll bar at the bottom of the screen to quickly move to other dates in the view you’re using
- Myriad new calendar printing enhancements

**To Do improvements**

- New ability in To Dos to show grid lines of different styles and colors, optionally creating a “ruled paper” look
- In-view editing of To Dos

**Other improvements**

- Ability for a user to customize views by dragging and dropping column headings and have those customizations “stick” the next time the database is opened by that user
- Enhanced replicator page, including new customization options, easy replica (and selective replica) creation via drag-and-drop, replication of databases by folders, sorting replicator page columns, improved progress indicator, reordering of replication so smaller documents are replicated first, and dynamic view rebuilding so documents appear in the view as they are replicated
- Consolidation of user-level security options in the new User Security dialog box
- Increased multi-tasking so you can do other things while, for example, an attachment is being saved to disk, a database is being copied, or the design of a database is being replaced
- Addition of multi-user support, which allows Windows-based Notes 6 client users (only) to share a single workstation (Note: Using this feature requires re-installation of the client code since the sharing of the workstation, Notes 6 data directories, and so forth, requires changes to Windows registry entries to support multiple users. This support is not available for users of Domino Designer 6 or Domino Administrator 6.)
Summary
There are many new and improved features for end users, which makes Lotus Notes 6 the premier messaging client for multi-tasking among mail, calendaring and scheduling, personal information, to-dos, and other applications efficiently. However, keep in mind that while improved usability and customizability were key goals for Notes 6, another primary goal was the ability for both novice and experienced users to intuitively know how to use these improvements without the need for extensive training or retraining. The improvements in the Notes 6 client achieve this goal, which should help you reduce your TCO for Notes clients.

1.5.2 What’s new in Lotus Domino 6 for Domino administrators

The main emphasis for Domino administrators has been threefold: centralization of Notes 6 client functionality, streamlined administration, and more integrated performance management.

Centralization of Notes client functionality

- Powerful support for policy-based management, including the ability to establish and maintain standard settings and configurations for end-user environments, thereby helping automate redundant tasks and eliminate potential help desk calls.

Policy documents can include archive settings/rules, desktop settings/configurations (including user Welcome pages), user registration settings (to allow simpler registration for groups of users with common settings/attributes), first-time setup settings/configurations, and security settings (including passwords and execution control lists (ECLs).

Policy documents can assist with “enforcement” of desktop client configurations since the user’s desktop will have the desktop configuration restored or distributed to the user at the next server authentication, thus reducing the effort to implement new applications by organization or subset of the organization.

The use of Policy documents and Settings documents can help you use Domino to enforce your organization’s standards rather than having your organization’s standards mapped to how Domino works (or, rather, used to work).

Policy documents are another way to automate redundant administration tasks, reducing the amount of repetitive work required by highly-skilled (and usually very busy) administrators.

- Upgrading and deploying Notes 6 client upgrades via IBM Lotus Notes Smart Upgrade allows users to execute incremental installers previously downloaded, thereby simplifying the process by which users request client upgrades, and encouraging users to upgrade to the release recommended by their administrators in a timely fashion. For more information see the Lotus Developer Domain at:

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

Smart Upgrades, in combination with the appropriate policy documents, can help administrators control timing of client upgrades (including specification of a time by which clients must be upgraded or the client upgrade will take place automatically).

Smart Upgrades work with Notes 6.0.0 (and later) clients, so this function is even more motivation to migrate your Notes clients to Notes 6 as soon as possible in order to minimize the workload on your support organization by automating the upgrade process. It can significantly decrease the amount of work required when Maintenance Releases (MRs) or Maintenance Updates (MUs) come out, if you want to implement them, without having your support team lose central control of client configurations.
More streamlined administration

- Powerful (possibly hierarchical) delegation of administration functions/authority to other users/groups on the same servers.
  
  This could be used to allow distributed administration for some (limited) functions but retain tight central control of other functions. For example, departmental administrators could be allowed to update Domino directory information for departmental users but probably would not be allowed to change server configuration documents.

- Significant improvements to the Web Administrator client, which is now Java-based, and which provides most of the same features and administrator user interfaces available on Domino 6 Administrator, including:
  
  - The ability to register and manage users (if the appropriate certificates are made available)
  
  - Group management functions
  
  - The ability to manage mail-in databases
  
  - Policies and Settings creation and management
  
  - The ability to manage certificates
  
  - Files management (for example mail files, databases, templates, and so forth), including database compacting, signing, ACL management, and viewing available disk space
  
  - Almost all server management functions (except Tivoli® Analyzer for Lotus, which is discussed in the next section), including server status, server users, database users, server tasks, HTTP statistics, and operating system statistics (which are new to Domino 6 for z/OS and are discussed in the next section)
  
  - Server analysis, including information about databases, mail routing, replication, logs, and administrative requests
  
  - Schedule management for programs, agents, mail routing, and replication
  
  - The ability to manage enterprise mail, including mail server tasks, mail routing activities and events, mail reports, mail.box management, logfile monitoring, and message tracking
  
  - Replication management capabilities, including replicator tasks, replication schedules, replication events, and replication statistics
  
  - Server configuration, including server documents/configurations/connections, directory functions, Web configuration, server monitoring, cluster management, and more
  
  - The ability to issue server console commands using Quick Console
  
  Administrators will be able to perform all of these functions on supported browsers, including Internet Explorer 5.5 or later on Windows workstations and Netscape Navigator 4.7 or later on Win32 workstations or Linux (RedHat 7.2 or SuSE 7.2) workstations, while using an interface that looks almost exactly like that of the Domino Administrator 6 client. Using the new Java-based Web Administrator client may be a helpful alternative when you decide to distribute some Domino administration functions but don’t wish for these new administrators to have to install Domino Administrator 6.

- Support for index (.ind) files, which are text files containing lists of files to be operated on by Domino 6 server tasks (such as fixup, compact, updall) to allow administrators to use a single console command (for instance fixup daily.ind, where daily.ind contains lists of certain users’ mail files) rather than multiple commands (such as one fixup command per mail file) or one lengthy command with multiple arguments (for example the list of those certain users’ mail files on the command line), which is not likely to get entered flawlessly if there are more than a few mail files.
Improved license tracking to allow you to manage the numbers and types of client licenses in a Domino domain.

Migration of person and group entries from an LDAP-compliant external directory to a Domino directory via the LDAP Domino Upgrade Service.

Enhanced support for Internet Site documents to control configurations for protocols such as HTTP, IMAP, POP3, SMTP, LDAP, IIOP, and also for global Web settings; this support can be especially useful in the service provider (xSP) environment where multiple Internet sites may be hosted on a single physical server.

An enhanced Internet Site view in the Domino directory (which replaces the Domino R5 Web Configurations view), a single view which contains configuration for all Internet server tasks.

Enhanced filtering of mail messages based on specified criteria and defined actions.

Ability to specify enhanced anti-relay restrictions so that your Domino server can’t be used as a “spam relay host.”

Allow specification of relay access for authenticated external hosts (for instance POP3 clients connecting through an ISP).

Allows administrators to specify that SMTP connections should be checked against the list of hosts on DNS Blacklist sites in order to cut down on unwanted mail.

Improved mail file quota specification/management.

New ability to specify user preferences for Web users, including time zone, date/time format, and number/currency format.

More integrated performance management

Enhancements to Domino Administrator 6 to optionally support the IBM Tivoli Analyzer for Lotus Domino. This is a separately licensed/priced/installed product, which does not require the use of any other Tivoli products, and which integrates with Domino Administrator 6. Key feature include:

- Server Health Monitoring, which allows real-time assessment of server performance and offers recommendations to improve server performance based on a combination of intelligent rules and past history.

- Activity Trends, which provides data collection, several alternative types of data analysis, and recommendations for resource balancing.

- The combination of these two tools allows administrators to make better use of (often limited) server resources. When used in conjunction with the AdminP process, automating the occasional relocation of users from one server to another is simplified with respect to effort expended and elapsed time.

- Support for the new Java-based Domino Console, which works with the new Java-based Domino Controller; this is discussed later in the chapter.

Summary

As you can see from the features enumerated, the goal of these enhancements is to leverage the skills and power of Domino administrators and the improvements in Domino Administrator 6 and the new Java-based Web Administrator to lower the total cost of ownership for the Domino 6 environment.
1.5.3 What’s new in Lotus Domino 6 for Domino designers/developers

There are many exciting new options for design/development of Domino applications.

**Application design**
- Allow document locking (by end users) for database documents as part of a design
- Document locking of design elements to avoid replication/save conflicts during a team design effort (with the ability for database managers to override a lock if appropriate)
- Printing source code from the Programmer’s pane
- Using hide-when features to streamline a mobile application
- Customizable “twisties”
- Ability to create borders for embedded elements
- Sharing non-NSF (that is, platform file system) files
- Creating layers in pages to enhance presentation
- Creating style sheets as shared resources
- The ability to add tools, such as third-party tools, that have not previously been part of Domino Designer®, so that you don’t have to leave Designer to work on another aspect of your design or development effort, using Eclipse technology. For more information, see: [http://www.eclipse.org](http://www.eclipse.org)
- Support for WebDAV (DAV stands for “distributed authoring and versioning”)
- New field types: rich text lite, color, time zone
- Enhancements to Shared Resources, which now include Files, Style Sheets, and Data Connections
- View design enhancements, including programmable column colors, custom icons and multiple icons in a view column, enhanced formatting and display options for numbers and dates in a view column, displaying a background image for standard views and calendar views, allowing users to customize views, allowing in-view document creation and/or editing, and many others
- Enhancements to shared actions
- Viewing and exporting XML for design elements
- New and enhanced URL commands
- Significant enhancements to context-sensitive Help for applications, including the ability to use a common set of JavaScript commands in both Notes and Web browsers

**Programming in general**
- Auto-completion (similar to type-ahead when addressing mail recipients) for @functions, @commands, and LotusScript statements; plus parameter prompting to show the programmer how to complete the programming statement quickly and efficiently
- Remote debugging for agents and script libraries running on the server
- Breakpoints persist across debugging sessions (unless explicitly cleared)
- Enhancements in the event model to make programming for Notes clients and for Web browsers more consistent. This feature also deprecates some previously common events in order to discourage their use and encourage use of the enhanced event model options (meaning they will continue to be supported but their use is discouraged); this means you can use JavaScript for programming functions for both Notes and Web browser client events.
Support for creating program libraries using JavaScript in addition to LotusScript and Java

The addition of several new types of database resources, including Files, Style Sheets, and Data Connections

The ability to organize applications by creating nested folders which contain arbitrary (meaning user-chosen) sets of databases, design elements, and even other object types that haven’t previously been part of the traditional set of Domino database design elements

The ability to drag-and-drop elements into folders in the bookmarks pane

The addition of Shared Code objects and enhancements to Shared Resources, ensuring ease-of-maintenance for commonly used items

The ability to select multiple elements for action, such as setting the properties of several elements at one time, thereby ensuring self-consistency in a large application

Pop-up displays for full path names for elements, which allows you to be sure you are changing the proper element if you choose to have separate versions of this element in two different parts of a complex application or, for example, the same element (say, a form) in a template and a database

The ability to bookmark a design element or code object in another database in order to centrally manage that item (for example, a block of JavaScript code or a commonly-used subform)

The ability to work with resources from other databases, including design elements such as views or forms

New context-sensitive Insert Resource dialog box, which makes it easy to select and reuse objects

Reorganization of agents, outlines, subforms, actions, shared fields, and script libraries into Shared Code in the Domino Designer 6 integrated development environment (IDE)

**Programming in XML**

- Enhanced support for the XML representation of Domino data (also known as Domino XML, or DXL) in applications
- Inclusion of the XML4J parser and the LotusXSL processor in Domino Designer 6 to allow parsing and transforming of XML
- Enhanced Java classes for the properties and methods needed to support the XML4J parser and the LotusXSL processor
- Significant enhancements to LotusScript to support processing XML, including support for the Simple API for XML (SAX) parser and import/export processing to transform Domino objects to/from the Document Object Model (DOM) format

**Programming with @formula language and @commands**

- Completely re-written @formula/@command engine, resulting in better performance
- Improved formula rules such as nested assignment statements, using braces to delimit text constants and remarks, subscripted lists, and reassignment of temporary variables
- Over forty (40) new @formulas and over fifteen (15) enhanced @functions, including enhancements using structured programming features such as looping, list manipulation, and subscripting
- Over fifteen (15) new @commands that duplicate existing @commands but run synchronously rather than after @functions
- Four (4) new @commands and over ten (10) enhanced @commands
Programming with LotusScript
- Ability to recompile all LotusScript code in a database with a single action
- Over fifty (50) new or enhanced LotusScript classes (some of which were previously downloadable, all of which are fully integrated now), including support for XML, DXL (DOMino XML), XSLT (XML Style Sheet Transformations), SAX parsing, and the document object model (DOM) also used by JavaScript
- New data types plus converters to support these types
- Support for accessing Java classes (programs) from LotusScript via a new LotusScript Extension (LSX)
- Five (5) new string functions that correspond to functions previously available in the @formula language
- A new option on the Open statement to allow you to specify the language to use for file I/O
- The addition of fourteen (14) properties and methods (previously only available in COM, which is basically unchanged in Release 6) to LotusScript
- Enhancements to the Domino Designer Help, including aliases for Microsoft Visual Basic functions/elements

Programming with Java
- Addition of two (2) new classes (MIMEHeader and Stream) and enhancements to two (2) existing classes
- Creation of over fifty (50) new methods, update of two (2) methods, and one (1) method obsoleted (which was replaced by other methods in multiple Domino programming languages)
- Addition of fifteen (15) new properties and one (1) property updated
- Removal of the requirement to include NCSOW.jar in the CLASSPATH to execute remote (IIOP) calls in the WebSphere environment, while continuing to support programs that use this jar file

Programming with Lotus Connectors
- Integration of Lotus Connectors, providing native access to a wide variety of data sources, including DBMSs, ODBC, platform file systems, transaction systems, and enterprise resource applications
- Extension of Lotus Connector support through an LSX (LotusScript Extension)
- Integration of the Lotus Connectors documentation with Domino Designer 6 Help

Agent improvements
- New agent builder interface
- Improvements in modifying, saving, and signing agents on the server
- Allowing server-based agents to access databases on remote servers
- The ability to specify on whose behalf the agent will run
- Ability to change a personal agent to shared and vice versa
- Allowing users with Editor access to enable or disable LotusScript and Java agents without requiring them to have Domino Designer 6 or Domino Administrator 6 installed
- The ability to enable scheduled agents using a Web browser
- Improved programmability (@formula, LotusScript, and Java) features to support enhancements to agent technology in both the Notes 6 client and the Domino 6 server
- A new remote debugger for server-based agents and other agent troubleshooting tools
- Multi-threading of background agents
- Performance improvements in Domino server startup and in agent loading/reloading
- Security enhancements that match the increased granularity of distributed administration levels possible with Domino Administrator 6
- Enhanced Domino console commands to support improved agent functionality, including a help index for the agent manager

**Web technology design/development improvements**
- The ability to use the same JavaScript code (called Common JavaScript in the IDE) for Notes clients and Web browsers
- Support for Domino applications as Web services, including creating WSDL (Web Services Description Language) files as page elements with XML content
- Integrated Style Sheet creation and editing
- Built-in WYSIWYG HTML creation (with type-ahead features like other programming languages), conversion, color coding, and syntax checking
- Support for layers on pages or forms

**Programming with JSP tag libraries**
- Provides JSP tag libraries that support quickly and easily incorporating Java functionality into JavaServer Pages.

  **Note:** The JSP has to execute on a J2EE-based Web Application Server such as (but not limited to) WebSphere Application Server (referred to hereafter as WebSphere), not on Domino. The JSP tag creation is done in the Domino Designer 6 environment. The JSP can be invoked from the Domino environment but has to execute in the J2EE environment.

- Provides access to Domino objects from a JSP using the Domtags tag library
- Provides access to commonly-used J2EE container tasks using the Domutil tag library
- Provides support for using Domino tags with WebSphere either locally (same server) or remotely (different server)
- Provides debugging and error handling support when using Domino JSP tag libraries

**Lotus Enterprise Integrator™ (LEI) enhancements**
- Easier-to-use, more intuitive task-oriented end user interface with significantly enhanced help
- Support for server-side browsing, allowing you to install the external connectivity software only on the Domino server where LEI and Domino Administrator 6 are installed, thereby increasing your flexibility while saving time and expense
- The ability to enable timestamp-based polling
- Improved coexistence with Domino Enterprise Connection Services (DECS)
- Virtual Field activity, which provides synchronous access to external data sources from Domino applications (similar to DECS’ RealTime activity)
- Virtual Document activity, which removes the need to store key documents in the Domino database, thereby leaving a smaller footprint in the Domino database while storing the data required to ensure integrity between the application and the external data in one place
- The ability to create Virtual Agents to run external system stored procedures
The addition of Integrated Credentials to allow you to correlate Notes user IDs with security credentials on external systems

The ability to create Virtual Attachments in the Notes client, which appear to be part of a document but are actually stored as a binary object on the external system

Inclusion of a multi-platform installer

Easier upgrade and migration from previous releases

**Summary**

Domino Designer 6 is now more powerful and easier to use. It delivers features that allow designers and developers to be more productive while managing more complex applications, reduce time wasted by switching from Designer to other tools (and back, not to mention importing the non-Designer work that was done), has easier-to-use/easier-to-manage presentation features, and powerful new programming features, plus enhancements to the existing tools previously available. Domino Designer 6 is truly a state-of-the-art integrated development environment.

1.5.4 What’s new about Lotus Domino 6 servers

In addition to the enhancements listed previously, the Domino 6 server has also been enhanced significantly on all supported platforms.

**Installation, setup, configuration, and operation**

- Enhancement to Domino server setup using a Java-based install process (without HTTP), provides consistency in functionality and appearance for Domino 6 server setup and initial configuration across all platforms.

- Support for remote server setup (without using HTTP) eliminates the need for an installer to be present at the remote system once TCP/IP communications are established with that system.

- Enhanced server setup profile support, including recording/playback of setup for various types of servers (for example, mail servers versus application servers).

- Support for multi-version UNIX installation (including z/OS UNIX System Services), meaning that multiple levels of Domino 6 can be installed in one hardware instance (for example, an LPAR) concurrently along with one level of Domino R5. This support would allow Domino 6.0.0 and Domino 6.0.1 (in the future), as well as one level of Domino R5.0.x to be installed in a single hardware instance concurrently. While production Domino partitions might more logically be maintained at the same code level, this support could potentially reduce your need to maintain several test systems just to support multiple levels of Domino code.

  **Note:** The number of concurrent servers you can support on a given set of hardware is dependent on the capacity/performance characteristics of your hardware, as well as how you intend to use this multi-level environment (for instance, test versus production).

- New UNIX installation options (including z/OS UNIX System Services), including making template installation optional, the optional creation of the /opt/lotus soft link during installation, support for setting up a service provider (xSP) after installing a Domino Enterprise Server, and adding data directories (only) to an existing Domino installation.

- Implementation of the Java-based Server Controller, which runs on a Domino server and controls the server.

- Addition of the Java-based Domino Console, which communicates with the Server Controller to allow you to manage your servers as if you were sitting at a local Domino server console. The Domino Console can run on any platform except Macintosh, can
control multiple Domino servers, and includes multi-release support for those servers which support the Server Controller. Domino Console does not replace the functions of Domino Administrator 6; it only provides console support, not administrator functionality.

- The algorithm that computes the Server Availability Index in Domino 6 is different from Domino R5. IBM technote 1164405 describes the details.
- Addition of console command line help for the `load` or `tell` commands for most server tasks using the `/?` or `-?` option after the command.

**Better performance**

- Enhanced database (NSF) design to allow for better performance
- Rewritten and optimized HTTP stack for significantly better performance
- New memory management techniques to make memory allocation faster and to improve overall memory utilization
- Minimizing and/or eliminating semaphore contention in the Domino 6 server task
- New lock manager for transaction logging to improve performance of transaction logging by locking at the record level rather than the database level when possible
- Addition of view logging to transaction logging, which can greatly decrease restart time for a Domino 6 server
- Support for a centralized directory to maximize server resources on servers which use only a Configuration Directory
- Support for streaming replication to reduce the need for client-to-server connect time and to make maximum use of network bandwidth
- Enhancements to iNotes Web Access 6, especially in the calendaring and scheduling area
- Optimizing the `@formula` engine code
- Internet protocols (such as POP3 and IMAP) support the threadpool model for better scalability

**Improved functionality**

- Support for access by agents running on other servers (within appropriate security limits)
- Enhanced AdminP functionality, including support for service provider (xSP) requests, system monitoring requests, and LDAP-related requests
- Support for index (.ind) files, which were described previously under enhancements to Domino Administrator 6
- Support for enhanced server mail rules, which were discussed previously under enhancements to Domino Administrator 6
- Journaling of mail messages acted on by server mail rules in order to allow later retrieval of those mail messages
- Improved server activity logging, which captures more data and allows more control over data collection than Domino billing, including server activity logging for xSPs

**Directory and LDAP enhancements**

- A new (optional) central directory architecture, which allows you to maintain a full Domino directory (`names.nsf`) on selected servers and a Configuration Directory (a replica subset containing only server configuration documents) on other servers, with Person and Group lookups to take place on (remote) designated directory servers, a reduced workload on servers with Configuration Directories since they no longer have to replicate all
information, and tighter central control over full directories since they would occur on far fewer servers

- Extended ACLs (xACLs), which allow you to refine user access to specific parts of the Domino directory (names.nsf) or Extended Directory Catalog (for example, xdc.nsf); also used to enforce database security for Notes client name lookups and for anonymous LDAP search access

- Activity logging for LDAP to allow you to see who is performing read and write activity

- Failover support for remote LDAP directories by specifying more than one value in the Hostname field in an LDAP Directory Assistance document

- New statistics for directory assistance (DA): DAReloadCount, DAFailoverCount, and DARefreshServerInfoCount

- Added custom LDAP filters for use with directory assistance to a remote LDAP directory to allow you to override the default search filters (for example, for mail address lookups, client authentication credential lookups, or group authorization lookups)

**Networking**

- Optional native network compression between Domino 6 servers and Notes 6 clients, which could be especially useful for clients which are not LAN-attached

- AutoDialer for coordinating dial-up ISP connections between servers

- Enhanced anti-relay restrictions

- Support for DNS Blacklist filters

- Optional support for LZ1 compression for attachments, which is more efficient than Huffman compression

**Security enhancements**

- Allowing revocation status of a certificate to be verified before trusting it via Certificate Revocation Lists for the Internet certificate authority (CA)

- Synchronizing Internet passwords with Notes ID passwords

- SSL session resumption, which allows authentication over SSL sessions to occur only on the first connection, with subsequent connections reusing the existing SSL keyring information, thus eliminating the heavy-duty cryptographic calculations necessary for SSL authentication on most session requests

- New HTTP protocol security options to limit the type and size of requests and content

**Web server enhancements**

- Complete re-write of the Domino 6 HTTP server with increased functionality (HTTP 1.1 plus extensions) and improved performance

- Support for persistent connections with Web clients, including iNotes Web Access 6

**Recovery and problem determination**

- Improved automatic fault recovery for Win32 and UNIX server platforms (includes z/OS platform) to allow a Domino server to automatically be shut down and restarted without operator intervention. In z/OS, an SVC dump is taken, nsd is run, and the Domino server is automatically restarted.

- New notes.ini variable to execute a script upon server failure as an alternative to automatic fault recovery

- Improved first failure data capture (FFDC) via enhanced system diagnostics (NSD)
Improved transaction logging

View logging for selected views (for example, the ($ServerAccess) and ($Users) views) in order to avoid rebuilding view indexes unnecessarily upon server restart

Ability to start/restart Domino servers from the Domino Console, even if the server is down

Color coding of console messages by severity level, if desired, on the Domino console

Other

Roaming user support has been deferred until Domino 6.0.1, depending on early trial program results.

Single copy template (SCT) has been deferred until at least Domino 6.0.1 or later, depending on test results.

Summary

As you can see, many features have been included to make Domino 6 easier to install, set up, customize, and operate. Emphasis on improved functionality, expanded support for standards and Internet protocols, added features for clients (Notes, administrator, and designer), increased reliability, tighter security, and better server and network performance mean that Domino 6 is the premier platform for Web applications, messaging, and e-business.

1.5.5 Cross-platform summary

The many improvements identified for Notes 6, Domino Administrator 6, and Domino Designer 6 and other client variations, plus significant improvements in the Domino 6 server across all server platforms, are powerful motivators to move to the Domino 6 and Notes 6/iNotes™ Web Access 6 environment as quickly as possible, especially when considering the possibility of lowering your total cost of ownership. The additional enhancements which are specific to the IBM zSeries (discussed in section 1.3 on page 3) make Domino 6 for z/OS an excellent platform choice.
Chapter 2. z/OS tuning

In this chapter we describe how Domino interfaces to z/OS, using the Domino address space and task structure. We discuss the tuning considerations for z/OS when you are running one or more Domino servers. We cover this topic in detail, after first presenting a summary of our recommendations.
2.1 Recommendations

These are the key recommendations for tuning z/OS to run Domino:

► Make sure you have enough central memory for Domino.

Domino creates multiple address spaces, some of which have large working sets. z/OS UNIX System Services also makes extensive use of storage for performance in the kernel address space, DFSMS dataspaces, and zFS colony address spaces. You need enough real storage on your processor for these or you will page to DASD, which will severely impact Domino response times.

► In the BPXPRMxx member in SYS1.PARMLIB, check your z/OS UNIX System Services settings and make them large enough.

There are several z/OS UNIX System Services parameters in BPXPRMxx which, if too low, will cause problems in the Domino server. The server may not be able to create new tasks or get the virtual storage it requires. See Domino 6 for z/OS Install Guide for the recommended values and follow them. This documentation ships with the product and is also available at:

http://www.lotus.com/idd

From the navigator window, select Documentation Library → by product → Domino for z/OS → 6.0. You can then select to download the install guide from there.

Note: Use the new Domino 6 for z/OS command, dom_verify_os, to verify your BPX parameter values. See 5.2, “Verifying BPX parameters after Domino installation” on page 89 for more details.

► Make the Domino server a high-priority z/OS workload.

The Domino server is an online system. It needs the same level of service that you would give to your production CICS or IMS systems. If the priority within z/OS is not set high enough, then response times to clients will suffer and client requests will time out.

► We recommend that z/OS APAR OA07605 be applied, since Domino 6 for z/OS makes heavy use of asynchronous I/O.

APAR OA07605 - SRB SCHEDULING ALGORITHM, ASYNC I/O REQUESTS, PERFORMANCE PTFs:
– R707 UA11607 - z/OS 1.4
– R708 UA11608 - z/OS 1.5
– R709 UA11609 - z/OS 1.6

► Tune and monitor the entire z/OS system.

If you add a Domino server to a badly-tuned z/OS system, it will suffer. Do all the things you would normally do for performance. Get IBM to help set up the system before you install the Domino server, if you need the additional expertise. Then, as you start to use Domino, monitor the system closely to see how Domino performs.

► Check out the z/OS UNIX System Services performance tuning tips at:


You can also read Debugging UNIX System Services, Lotus Domino, Novell Network Services, and other Applications on z/OS, SG24-5613.
2.2 Planning the server architecture

Now we get into more detail about the z/OS environment and some tuning techniques. This section deals with how Domino uses address spaces in z/OS, how to run multiple Domino instances (called Domino partitions, or DPARs, in Domino terminology) in one system or LPAR, and what z/OS parameters to set for optimum performance.

2.2.1 z/OS address spaces

Domino 6 for z/OS uses multiple z/OS address spaces, as shown in Figure 2-1 on page 25.

![Figure 2-1 Domino tasks in z/OS address spaces for a single Domino instance (DPAR)](image)

At initialization, the main server address space (UNIX process) is always started. Additional server address spaces are created at startup, depending on the functions requested in the initialization control file, called notes.ini (even though this file refers to the Domino server and not to the Notes client in this case). These additional server address spaces perform background tasks or functions requested by threads in the main server address space in order to service client requests. Server address spaces can also be started at the request of the Domino administrator while the Domino server is running. Domino also dynamically starts and stops server address spaces to perform some long-running background tasks.

The Domino address spaces are created and managed by Workload Manager (WLM). After a Domino process has completed, Workload Manager keeps the address space available for 30 minutes to allow reuse of the address space by another Domino process. Address spaces are retained in Workload Manager goal mode. Domino address spaces retained by the Workload Manager keep their Domino job name but have a step name of BPXAS.

**Address spaces for user connections**

Thread pooling was introduced with Domino R5 for S/390 for Notes servers to reduce the resource requirements of the server. In this design, each server thread supports many connections. Figure 2-2 on page 26 shows the thread pool model and how the server process uses it.
When a client connects to the server, a client session is created with a unique socket descriptor and a virtual thread ID is assigned. When a client makes a request to the server, a physical thread is associated with the virtual thread and does the required work on behalf of the client's session. After the work is completed, the physical thread is returned to the pool of physical threads. The physical thread is no longer associated with the virtual thread and is available to service another request. With this reusable thread model, the server needs far fewer physical threads than the maximum number of concurrent user connections. In Domino R5, only Notes clients used thread pools, but in Domino 6 they are also used by clients that utilize Internet protocols such as IMAP and POP3.

By utilizing a single main server address space and a small fixed number of physical threads to service client requests, Domino's resource requirement on the system is much lower. This means less contention for common system resources, and less real storage consumed per concurrent user request. The end result is a much more linear growth curve.

**Address space naming**

The Domino server creates all its address spaces with a job name of the setting of Notes.OS390_JOBNAME_PREFIX, followed by a character. This is shown in Figure 2-3, where the system is running a Domino 6 for z/OS server. The display shown is produced using the z/OS command `D A,L`. This server has more functions started than are normally seen on a Domino server.

The job name prefix is specified by including an export statement in the .profile file of the server's user ID's home directory. For example:

```
export Notes.OS390_JOBNAME_PREFIX=MAIL1
```

where MAIL1 is the job name prefix. The main server address space will be MAIL1SER and other address spaces will be of the form MAIL1xxn, where xx is two characters which show the Domino function and n is a one-character suffix of 1 to 9 or A to Z.
For example, `MAIL1RO1` would be a router address space and `MAIL1RE1` would be a replicator address space.

If you do not specify a job name prefix in `Notes_OS390_JOBNAME_PREFIX`, the job names are the user ID under which the server is started, followed by a number. With this form of job name, you cannot predict what the job name will be for a particular function, which makes it more challenging to manage your Domino environment, especially if you implement more than one Domino instance (DPAR).

We strongly recommend that you specify your own job name prefix to help simplify systems and performance management.

**Address space and program name**

If you do not specify your own job name prefix and wish to obtain the function names of address spaces on a running Domino server, issue the z/OS command `DOMVS,U=userid`.

This user ID is the one you used when you started the Domino server.

The function being performed by this Domino address space can be determined from the program name, which is the last qualifier in the CMD field.

**Note:** If the path name is too long, the CMD field may not be large enough to display the full program name.

A list of server task names is provided in 5.6, “Domino server tasks” on page 102, as well as in the Domino Administrator 6 Help database. For a full list, refer to the *Domino 6 for z/OS Install Guide*. Note that the program names may appear in upper case or lower case, but they are the same program.
2.2.2 Multiple Domino servers

Multiple Domino server instances can run in a single z/OS image. An instance of a Domino server is called a Domino partition (DPAR). Each DPAR has the set of address spaces just described. Each DPAR is independent of other DPARs, with its own address spaces and files.

Two DPARs use TCP/IP to communicate and transfer data between one another. This means they do not have to be in the same image or processor.

This gives you a high degree of flexibility on where to run a Domino server. It can easily be moved from one z/OS image to another. The file system (or systems) containing system and user files for the Domino partition can be unmounted from one z/OS image and mounted on a second z/OS image. The Domino server can then be started on the second z/OS image.

As the number of users increases, it may be necessary to create more DPARs. The recommendations for the number of users in each DPAR are in 6.1.5, “Partitioning recommendations” on page 118.

You can run multiple DPARs in the same z/OS image, or in different Processor Resource/Systems Manager (PR/SM) Logical Partitions (LPARs) on a single processor, or spread across more than one processor. You can configure the Domino servers to meet the capacity required to support your Notes clients. We give recommendations on the number of DPARs per LPAR in 6.1.5, “Partitioning recommendations” on page 118.

2.2.3 Domino on symmetric multiprocessors

Domino makes effective use of multiple processors by using multiple threads and processes to execute work concurrently.

The Domino main server address space has a pool of physical threads, each of which is a separate task, and more than one of these tasks can execute concurrently on a multiprocessor.

Domino has other address spaces to perform functions on behalf of the clients or as background functions. Each of these may be able to run concurrently with the server tasks.

You can also specify that more than one copy of a server function is started in the notes.ini initialization parameters, if required. The following is an example:

```
ServerTasks=Replica,Replica,Router,Update,Stats,...
```

In this example there are two address spaces performing the Replication function and one each for Router, Update, and Statistics.

2.2.4 Setting z/OS parameters

It is important to set up z/OS and z/OS UNIX System Services to allow for the number of address spaces and UNIX processes created by a Domino server. Since Domino also makes extensive use of storage, it is important that the z/OS UNIX System Services storage parameters in SYS1.PARMLIB(BPXPRMxx) are set to the recommended values.

These values are listed in Domino 6 for z/OS Install Guide. They are discussed in detail in Lotus Domino for S/390 Release 5: Installation, Customization and Administration, SG24-2083. Table 2-1 on page 29 lists the parameters that you should check.

There is a program supplied with Domino 6 for z/OS to check the BPXPRMxx settings. After setting the PATH environment variable, execute the program `dom_verify_os`.
Table 2-1  z/OS parameters

<table>
<thead>
<tr>
<th>Parameter type</th>
<th>SYSLIB.PARMLIB member</th>
<th>Parameters</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>IEASYSxx</td>
<td>MAXUSER</td>
<td>Add at least 10 to your current value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECSA</td>
<td>65M to 100MB²</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ESQA</td>
<td>65M to 100MB²</td>
</tr>
<tr>
<td>z/OS UNIX System Services</td>
<td>BPXPRMxx</td>
<td>AUTOCVT</td>
<td>NO (default)¹</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FORKCOPY</td>
<td>COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCSEMNIODS</td>
<td>20,000 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCSEMNOPS</td>
<td>32,767 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCSHMPAGES</td>
<td>25,600 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCSHMNIDS</td>
<td>500 (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCSHMNSEG</td>
<td>1,000 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCSHMSPAGES</td>
<td>2,621,440 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXPROCUSER</td>
<td>At least 50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXASSIZE</td>
<td>2,147,483,647 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXCPUTIME</td>
<td>2,147,483,647 (max)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXFILEPROC</td>
<td>65,535 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXMEMAREA</td>
<td>33,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXQUEUESIGS</td>
<td>100,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXSHAREPAGE</td>
<td>32,768,000 (maximum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXTHREADS</td>
<td>1,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXTHREADTASKS</td>
<td>1,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETWORK DOMAINNAME (AF_UNIX) MAXSOCKETS</td>
<td>10,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETWORK DOMAINNAME (AF_INET) MAXSOCKETS</td>
<td>35,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SHRLIBRGNSIZE</td>
<td>50,000,000 for each program directory¹,³</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SHRLIBMAXPAGES</td>
<td>4096¹</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSPLEX</td>
<td>NO¹,⁴</td>
</tr>
<tr>
<td>z/OS CSM</td>
<td>IVTPRMxxx</td>
<td>FIXED MAX</td>
<td>120M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECSA MAX</td>
<td>30M</td>
</tr>
</tbody>
</table>

¹ New parameters in Domino 6 for z/OS.
² Add to the ECSA and ESQA size the requirements for other application which run alongside Domino. Also add 10 MB ESQA for each Domino 6 for z/OS server at a different service level.
³ Add 50,000,000 for each additional DPAR running at a different service level. You should have a single program directory for a Domino 6 service level and it should be shared between all DPARs running at that level.
⁴ Production Domino servers should not share active files in a Sysplex.
Modules should also be placed in shared storage and libraries in the link list. Also, dynamic LPA should be set up for XPLink. Details about the modules and libraries can be found in 2.5.6, “Place program module in shared storage and LPA” on page 34.

2.3 Managing the Domino workload

Your Domino servers should receive the same consideration that you would give a production CICS or IMS system. They should have the resources required to give good response times to the user.

We recommend that you use Workload Manager in goal mode to manage your z/OS system when running a Domino server. Workload Manager will attempt to give your Domino server the resources it needs.

It is strongly recommended that you not single out individual Domino address spaces for specific treatment. In other words, do not attempt to specify higher or lower priority for individual Domino address spaces. This also includes related address spaces such as virus checkers. Domino uses extensive locking between tasks, so all tasks in a Domino server should run at the same priority. Delays in dispatching of one address space may cause delays elsewhere, which will increase user response times and also increase CPU consumption.

If you plan to run the Domino server and other high-priority workloads in the same z/OS image, ensure that you have enough processor capacity (CPU, storage, and I/O connectivity) to meet the needs of all workloads without impacting the responsiveness of each.

It is important to maintain the response time of the Domino server to the Notes client. If the Domino server does not respond within a specified period, as set on the Notes client, then the client treats this as though the server is not available. This TCP/IP session timeout is specified in the Notes User Preferences settings in the Ports dialog box on the client by using the TCPIP Options button to change the “Connection attempt time-out duration” (see Notes Help). The default value is 5 seconds. If this value is exceeded, the user will then have to decide whether to retry the request, switch to another server, or work locally on the client. This situation should be avoided.

2.3.1 Workload Manager goal mode

z/OS Workload Manager goal mode is the supported way to manage Domino servers. Even if you are only running a single Domino server on z/OS, goal mode simplifies the setup of your system to meet its performance objectives. It avoids potential errors that can result from setting parameters in compatibility mode. Most people find that goal mode can be set up fairly quickly on a system, even if they have not used it before.

In IBM, we use Workload Manager (WLM) to control our Domino servers for a large user population. An example of the Workload Manager definitions for a set of production Domino servers can be found in Appendix A, “Workload Manager (WLM) example” on page 165.

A Domino server should be placed in a service class with a high velocity and not have multiple performance periods. We recommend a value of at least 60 or 70 for the velocity. The Domino server should also have a high importance relative to other workloads.

You may wish to manage each DPAR separately by defining a separate service class for each DPAR. When running Domino DPARs across several z/OS images within a sysplex the WLM policy applies to all images within the sysplex. You will need to define a unique service class for each DPAR within the sysplex. The workload classification can use the job name prefix to select the appropriate service class.
One advantage of using a separate service class for each DPAR is that a problem in one DPAR will not take CPU resource from other DPARs in the LPAR. Another advantage is that DPARs within an LPAR may be set to different importance and velocity goals to give priority to some users or applications over others.

When setting up Workload Manager for a Domino server, you need to:

- Define a workload for z/OS UNIX System Services work.
- Define service classes for z/OS UNIX System Services work:
  - z/OS UNIX System Services startup processes which run under the user ID OMVSKERN, in the OMVS subsystem.
  - Forked address spaces, such as Domino server address spaces, which run under the user ID used for the Domino server, in the OMVS subsystem.
- Define classification rules for the OMVS subsystem:
  - Startup processes under user ID OMVSKERN.
  - Forked address spaces, such as the Domino servers.

Consider turning I/O priority management on. You can do this from the ISPF WLM administrative application, option Service Coefficients/Options. If you have an IBM TotalStorage Enterprise Storage Server (ESS, also commonly known as “Shark”) installed, turn on the Dynamic Alias Management option as well. Dynamic alias management requires I/O priority management to be active. Dynamic alias management, together with the Parallel Access Volumes (PAV) feature of ESS, can greatly improve performance of your DASD and hence your Domino installation.

### 2.4 XPLink

XPLink was introduced in OS/390 Version 2 Release 10. It was designed to improve the performance of call-intensive C and C++ applications. Domino is an ideal candidate for the benefits of XPLink, and with Domino 6 for z/OS significant CPU savings have been achieved by the implementation of XPLink.

XPLink is a new linkage convention for OS/390 and z/OS. It achieves the performance improvement by reducing the number of instructions for calling a module by changing register conventions and the layout of the stack. Applications can have a mixture of code compiled with XPLink and code with the previous z/OS linkage conventions.

Domino makes many nested calls during its execution and thus the CPU savings of Domino 6 for z/OS over Domino R5 for S/390 are primarily the result of the implementation of XPLink.

More information on XPLink can be found in XPLink: OS/390® Extra Performance Linkage, SG24-5991. This redbook documents a benchmark of a Domino Mail with Calendaring and Scheduling workload, showing significant CPU savings.

There is a possibility that XPLink might be implemented in the future for other call-intensive features often added to Domino, such as virus-checking.

### 2.5 Storage for Domino servers

Here we discuss how Domino servers use the various types of storage available on z/OS.
2.5.1 Virtual storage

As described in 2.2.1, “z/OS address spaces” on page 25, a Domino server creates a large number of address spaces. Some of these address spaces, such as the main server address space, use a large amount of virtual storage, particularly in extended private (above 16 MB addressing) storage.

For Domino 6 for z/OS Notes, each connection is represented by a virtual thread which exists in the main server address space. So the storage to support the virtual threads which service the user connections is allocated from a single address space.

You should plan to provide a minimum address space size of 1.6 GB. To achieve this you should not excessively over-size ECSA and ESQA. RMF™ provides a virtual storage report which shows the size and use of ECSA and ESQA. In addition, an enhancement is planned in Domino 6.0.1 to allow the domps -p command to show virtual storage usage for 31 bit processing. See Appendix B, “Using the domps command” on page 171 for a description of the current command and examples of its output.

2.5.2 Real storage

The storage requirements of a Domino server grow with the user load they are required to support. As a result, large server deployments will require correspondingly large amounts of real storage. You should not add Domino servers to a z/OS image without sufficient processor storage to support the combined workloads. This would cause paging to auxiliary page datasets, affecting the Domino servers and the other workloads on the z/OS image.

A 31 bit z/OS image can be configured with more than 2 GB of real storage, and the storage can be split between central and expanded. It is recommended that you define as much storage as possible as central storage, up to the 2 GB limit; any extra storage would be defined as expanded storage.

The processor storage required to support a number of Domino users is given in 8.3, “Estimating processor storage” on page 150. The processor storage required is influenced by the number of address spaces and the working set of these address spaces. The working set size is the number of 4 KB pages that the program references. These pages include program instructions and data. This is smaller than the virtual storage size, because there are parts of programs and data that are not used during normal running of the server. These include parts used only at Domino startup, error routines, and infrequently-used functions.

Unreferenced pages in the working set will be stolen (paged out) by z/OS so they can be used by other address spaces. If the z/OS image has far more storage than is required to run its workloads, then these pages are not stolen and the working set remains larger than is actually required.

Table 2-2 shows an example of processor storage use for a system that has two Domino server partitions, using the Mail with Calendar and Scheduling workload. There is some additional support-type work running. We collected this data from an RMF report.

<table>
<thead>
<tr>
<th>Component</th>
<th>Processor storage used (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total processor storage used</td>
<td>2624</td>
</tr>
<tr>
<td>- Central storage:</td>
<td>2046</td>
</tr>
<tr>
<td>- Expanded storage:</td>
<td>578</td>
</tr>
<tr>
<td>Shared storage</td>
<td>1035</td>
</tr>
</tbody>
</table>
Table 2-3 shows how much storage is used for each Domino server task. We collected this data from SMF 30 records; therefore it doesn't include the shared storage.

<table>
<thead>
<tr>
<th>Component</th>
<th>Processor storage used (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMVS</td>
<td>337</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>31</td>
</tr>
<tr>
<td>ESQA</td>
<td>32</td>
</tr>
<tr>
<td>Domino server 1</td>
<td>316</td>
</tr>
<tr>
<td>- 1849 registered users</td>
<td></td>
</tr>
<tr>
<td>- 681 connected users</td>
<td></td>
</tr>
<tr>
<td>Domino server 2</td>
<td>177</td>
</tr>
<tr>
<td>- 489 registered users</td>
<td></td>
</tr>
<tr>
<td>- 144 connected users</td>
<td></td>
</tr>
</tbody>
</table>

Table 2-3   Storage used for Domino tasks

<table>
<thead>
<tr>
<th>Domino task</th>
<th>Server 1 (MB)</th>
<th>Server 2 (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>21</td>
<td>16</td>
</tr>
<tr>
<td>sched</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>amgr</td>
<td>90</td>
<td>11.9</td>
</tr>
<tr>
<td>event</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>adminp</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>collect</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>update</td>
<td>77</td>
<td>27</td>
</tr>
<tr>
<td>callconn</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>router</td>
<td>25</td>
<td>17</td>
</tr>
<tr>
<td>replica</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>stats</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

### 2.5.3 64-bit processor storage

The number of Domino partitions that can run on a given LPAR depends primarily on two factors: the amount of CPU, and the amount of storage available. From a CPU standpoint, LPARs can generally be made as large as necessary, limited only by the amount of CPU that the system architect is willing to give to the LPAR. The story is different for storage though. S/390 (and zSeries machines in 31 bit mode) allow for 2 GB of central storage and 2 GB more of expanded storage, giving a total of 4 GB of real storage that can be used by an LPAR. This is most often the factor that limits the number of Domino partitions that can run on one LPAR.

Our past experience has shown that two Domino partitions heavily loaded with Notes client users will allocate roughly 2 GB or virtual storage, plus some additional overhead for other processes associated with the server. However, the working set of this virtual storage that actually needs to be backed by central storage at any instant in time is substantially smaller than this. One LPAR can effectively manage the virtual storage needs of two Domino partitions, given 2 GB of central and 2 GB of expanded real storage.
With 64 bit real storage on zSeries, there is no expanded storage, so all real processor storage is considered to be central. The 4 GB real storage limit has also been raised to 64 GB, effectively eliminating the need to stop at two Domino partitions in an LPAR. Just as Domino R5 lifted the scale limitations of the release 4 server, and allowed for Domino server consolidation, 64 bit real addressability of zSeries allows the number of LPARs to be significantly consolidated.

Tests in the lab have indicated that it is possible to run at least 12 Domino partitions on one LPAR, given the necessary amounts of CPU and real processor storage. This was under a benchmark workload, and is not generally a recommended configuration for a production environment. It does serve as a useful indication that more than two Domino partitions can run in a single LPAR. In general, we believe that you can run between four and six DPARs per LPAR in production.

Note that all other non-Notes client workloads follow these same general rules of thumb. iNotes Web Access or POP3 users will cause Domino to use CPU and storage very differently, and Domino itself has different internal scale limitations for these different types of users. However, z/OS in 64 bit mode is much more capable of satisfying the resource requirements for multiple DPARs with a single LPAR. Monitor your server resource usage, and as long as the resources are available, you can consolidate Domino partitions onto an LPAR.

2.5.4 Page datasets

When supporting large Domino server workloads, there may be a significant increase in page space needed. Therefore, you should evaluate the size and number of page datasets on the system. The page datasets must be able to hold the virtual storage requirements of the address spaces in the z/OS image. The size of the page datasets and the number used can be found in the PAGESP report in RMF.

Always allocate far more page dataset space than you require for the normal running of your z/OS system. Then, if you experience an increased demand for processor storage, such as when z/OS takes a system dump of a large address space or part of the processor storage goes offline, you will have enough page dataset space to handle these situations.

2.5.5 Paging

A Domino server does not like paging to auxiliary page datasets. The configuration should have enough real storage to satisfy the working set requirements for the number of DPARs running in the LPAR.

If the DPAR does any paging from auxiliary, then user response times will increase and the CPU used by the DPAR will also increase. The increase in CPU is caused by some processes spinning waiting for another process to complete, which in turn is delayed by the page faults.

Paging from expanded storage in 31 bit mode is not a concern as long as the page movement rate from expanded storage is not too high.

2.5.6 Place program module in shared storage and LPA

When many address spaces use the same program, it is not efficient for each address space to have its own copy. Each copy consumes real storage. It is better to have just one copy in real storage that all the address spaces can use.
For Domino R5 for S/390, the Domino programs were placed in the Extended Link Pack Area (ELPA). In Domino 6 for z/OS, many of the Domino programs are placed in shared library (system and user) space. In both cases there is only a single copy of the program in real storage.

There are also programs from z/OS UNIX System Services and the C/C++ Runtime Library that should be placed in the ELPA.

**Domino R5 for S/390 program modules in LPA**

Domino Release 5 and earlier releases recommend that certain modules be placed in the ELPA. If you are running one of these releases, you will still have their modules in the ELPA. This process is described in *Lotus Domino for S/390 Release 5: Installation, Customization and Administration*, SG24-2083.

**Domino 6 for z/OS program module in shared library space**

Domino Release 6 does not put its main modules in the LPA. Domino Release 6 defines the program libnotes as a Shared Library Program. This means that there is still a single copy of the module in storage for all Domino address spaces at the same maintenance level. This process is described in the *Domino 6 Installation Guide*.

You can have more than one level of Domino 6 for z/OS in the same z/OS image, and also a single level of Domino R5 for S/390 in the same z/OS image.

You must make sure that there is enough space in the system shared library for the number of concurrent levels of Domino 6 for z/OS that you wish to run, plus other applications and system requirements.

Make sure that you use only one program directory for a level of Domino 6 for z/OS within an z/OS. If you use several program directories for the same Domino 6 for z/OS level, then you will have multiple copies of the program code, which will take up extra virtual and real storage.

You also need to set the SYS1.PARMLIB member BPXPRMxx parameter SHRLIBRGNSIZE to 50,000,000 times the number of concurrent Domino 6 for z/OS levels you wish to run in the LPAR.

To specify that the Domino Release 6 program is placed in Shared Library space, issue the command:

```
  extattr +l /notesdatapath/libnotes
```

where notesdatapath is the path to the modules of your level of Domino 6 for z/OS.

The user ID under which the server will run needs to be authorized to the BPX.FILEATTR.SHARELIB RACF Facility Class.

To check that libnotes is defined as a Shared Program Library in the executable directory enter the following command:

```
  ls -E libnotes
```

The result should be of the form:

```
-rwxr-xr-x --sl 1 IBMUSER NOTES 50593792 Sep 18 23:20 libnotes
```

For “--sl”, “s” indicates that the program is enabled to run in a shared address space and “l” indicates that the program is loaded from a shared library region.
Additional ESQA space
You will need more space in ESQA also. To handle the additional control blocks, plan for an additional 10 MB in ESQA for each Domino 6 for z/OS level.

C/C++ Runtime Library program modules
We recommend that you place certain C/C++ Runtime Library programs in the Link Pack Area (LPA). (Nearly all the modules go above 16 MB in the Extended Link Pack Area (ELPA).)

For a list of the C/C++ Runtime Library program modules to place in LPA and how to do it, see *OS/390 UNIX System Services Planning*, SC28-1890 (edition 07), Chapter 16, “Tuning performance.”

The C/C++ Runtime Library programs that you do not put into LPA should be cached in the Virtual Lookaside Facility (VLF).

2.6 PR/SM™

Processor Resource/Systems Manager (PR/SM) provides the capability to run multiple copies of z/OS and OS/390 operating systems on a single processor. Each instance of the operating system is called a z/OS or OS/390 image and runs in a Logical Partition (LPAR).

To support a large number of Domino users, you run multiple DPARs in an LPAR and have multiple LPARs. An LPAR has dedicated or shared processor engines (CPs), dedicated storage, and can have dedicated or shared channels for connection to I/O control units. It is important to set the LPAR definitions correctly for CP control and to give the LPAR sufficient storage so that it does not page to auxiliary page datasets.

You can specify that an LPAR can have dedicated CPs, but usually this is not done because it does not give the flexibility to use the total processor capacity to meet each LPAR's peaks and troughs. If an LPAR with dedicated CPs is not using all its capacity, then its spare capacity cannot be used by another LPAR that is short of capacity.

With shared CPs, you define how many CPs an LPAR can use (these are called logical CPs). You also specify a weighting for each LPAR with shared CPs. This weighting describes to PR/SM how you wish the CPs on the processor (these are called physical CPs) to be apportioned between the LPARs.

Table 2-4 is an example of a processor with six physical CPs and four Domino LPARs:

<table>
<thead>
<tr>
<th>LPAR NAME</th>
<th>Logical CPs</th>
<th>Weight</th>
<th>Maximum % of Processor</th>
<th>Relative Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOM1</td>
<td>5</td>
<td>400</td>
<td>83%</td>
<td>40%</td>
</tr>
<tr>
<td>DOM2</td>
<td>2</td>
<td>400</td>
<td>33%</td>
<td>40%</td>
</tr>
<tr>
<td>DOM3</td>
<td>3</td>
<td>150</td>
<td>50%</td>
<td>15%</td>
</tr>
<tr>
<td>DOM4</td>
<td>3</td>
<td>50</td>
<td>50%</td>
<td>5%</td>
</tr>
</tbody>
</table>

DOM1, DOM2, and DOM3 are production LPARs; DOM4 is a test LPAR.

The number of logical CPs and the weights are set on the processor Hardware Management Console (HMC). To change the number of logical CPs, the LPAR needs to be restarted. The LPAR weight can be changed dynamically while the Domino workload is running. The LPAR
weight is just a number and an LPAR’s share of the processor (the “Relative Weight” column) is its weight divided by the sum of the weights of all the active LPARs. Making the sum of the weights 1,000 gives good granularity and it is easy to calculate the relative weight.

The “Maximum % of Processor” column shows how much of the total capacity of this processor the LPAR can consume. The “Relative weight” column shows how the capacity is shared when all the LPARs require at least their share, that is, the processor is running at 100% utilization. If an LPAR at any time does not require its share then it is available for other LPARs.

The example in Table 2-4 shows a configuration that would not deliver the desired results and has some performance issues.

DOM1’s planned share of the processor is 40%; to deliver this would require 2.4 logical CPs. Allowing for peaks, 3 or 4 logical CPs would seem to be better than the specified 5 logical CPs.

Specifying more than the required number of logical CPs adds to the multiprogramming effect of the LPAR (consuming more CPU) and also to the CPU used for PR/SM management of these CPs. If you wish to allow for an unusual capacity demand, then leave the number of logical CPs at 5 but VARY OFF (the best time to do this is at Initial Program Load (IPL)) one or two CPs for the LPAR. The effect of VARYing OFF a CP is the same as if the LPAR was defined without that CP. When the unusual peak occurs, then you can VARY ON the CPs. This technique allows you to be able to react to the unusual peaks without stopping the Domino server.

On a zSeries processor running z/OS 1.2 or higher, a new feature called Intelligent Resource Director is available. This will dynamically vary the weight of an LPAR and also change the number of logical CPs for the LPAR under control of the WLM. The use of this feature will make management of LPARs much easier.

The CPU cost of PR/SM managing the LPARs can be gauged by the ratio of the sum of the number of logical CPs for all active LPARs divided by the number of physical CPs on the processor. This assumes there are no LPARs with dedicated CPs. This ratio should be 2 or less. In the example the ratio is $13 / 6 = 2.17$, which is greater than the recommended ratio, but with our proposed change to the number of logical CPs for this LPAR it will then be less than 2.

DOM2 has a desired share of 40% of the processor. However it has only two logical CPs, which restricts the LPAR to executing on only two physical CPs at any point in time. Thus it cannot use more than 33% of the capacity of the processor. In order to satisfy the desired share, DOM2 should have the same number of logical CPs as DOM1.

DOM3 and DOM4 both have 3 logical CPs but a low relative weight. DOM3 could be reduced to 2 logical CPs and DOM4 to 1 logical CP. For a production Domino LPAR you should have a minimum of 2 logical CPs, but for a test Domino LPAR 1 logical CP may be sufficient.

It is recommended that you monitor the normal CPU requirement of each LPAR and set the weights to match this. During migration the weights may need to be adjusted many times to match the changing requirement of the LPARs.

You need to make sure that LPAR definitions do not artificially constrain your Domino servers.
2.7 Monitoring z/OS

An essential part of running any application is to be able to monitor the resources consumed. This provides input into both performance tuning and capacity planning. In this section we discuss the major sources of information about a Domino server provided by the z/OS system on which it runs. DASD I/O monitoring is covered in Chapter 3, “DASD and file system tuning” on page 55. We discuss Domino monitoring techniques in Chapter 7, “Domino monitoring” on page 127.

Traditional z/OS information sources provide a lot of information for capacity planning, resource accounting, and performance tuning. These have been enhanced for applications such as Domino that use the z/OS UNIX System Services interfaces. First we discuss the Resource Measurement Facility (RMF), which provides system-level information that is useful for capacity planning and tuning. Then we look at the data available in System Management Facilities (SMF), which is useful for monitoring Domino resource usage, both CPU and I/O.

Additionally, Domino 6 for z/OS captures information in an SMF record. This includes Domino transaction, user, database, and monitoring information. See 7.1, “SMF type 108 record” on page 128 for more information on the Domino SMF record contents.

2.7.1 RMF

Resource Measurement Facility (RMF) is used to monitor the activity of the z/OS system. RMF gives information about CPU, storage, workloads, DASD controllers, and DASD volumes.

There is a lot of detailed information available with RMF. Here we include a discussion of the z/OS UNIX System Services OMVS report, some key system indicators, and useful data about a Domino server workload. Details on producing RMF reports and interpreting them are found in:

- z/OS RMF User's Guide, SC33-7990
- z/OS RMF Report Analysis, SC33-7991
- z/OS RM Performance Management Guide, SC33-7992

RMF has two online monitors, which you can access from TSO:

- The RMFMON (Monitor II) command shows data about a processor’s CPU, storage, address spaces, and DASD volumes.
- The RMFWDM (Monitor III) shows if an address space is being delayed and what is causing the delay. RMFWDM keeps information in storage or on DASD so you can look at delays after the event.

Some RMF Monitor II data is now recorded in Domino's platform statistics. See 2.7.3, “Monitoring z/OS with Domino statistics” on page 50 for more details.

Table 2-5 on page 39 shows you which report option to specify and which SMF records to collect when generating various RMF reports.
Table 2-5  Interval and duration reports

<table>
<thead>
<tr>
<th>Report name</th>
<th>Report option</th>
<th>Gathered by</th>
<th>SMF Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>SUMMARY(INT)</td>
<td>Monitor I</td>
<td>70</td>
</tr>
<tr>
<td>OMVS Kernel Activity</td>
<td>REPORTS(OMVS)</td>
<td>Monitor III</td>
<td>74</td>
</tr>
<tr>
<td>CPU Activity and Partition Data</td>
<td>REPORTS(CPU)</td>
<td>Monitor I</td>
<td>70</td>
</tr>
<tr>
<td>Paging Activity</td>
<td>REPORTS(PAGING)</td>
<td>Monitor I</td>
<td>71</td>
</tr>
<tr>
<td>Workload Activity (Goal mode)</td>
<td>SYSRPTS(WLMGL)</td>
<td>Monitor I</td>
<td>72</td>
</tr>
<tr>
<td>Device Activity</td>
<td>REPORTS(DEVICE)</td>
<td>Monitor I</td>
<td>74</td>
</tr>
</tbody>
</table>

Figure 2-4 is an example of the job control language (JCL) and control statements to produce an RMF report.

```plaintext
//XXXXXXX JOB '00','WTD',MSGLEVEL=(1,1),MSGCLASS=T,
// NOTIFY=XXXXXXX,USER=XXXXXXX,CLASS=A
//RMFPP EXEC PGM=ERBRMFPP,REGION=0M
//MFPINPUT DD DISP=SHR,DSN=INPUT.DATA.SET
//MFPMSGDS DD SYSOUT=* 
//SYSIN DD *
RTOD(1400,1600)
STOD(1400,1600)
DATE(02195,02196)
DINTV(0030)
REPORTS(CPU,PAGING,OMVS,DEVICE(DASD))
SYSRPTS(WLMGL(POLICY,SCLASS,RCLASS))
SYSOUT(T)
/*
//
```

Figure 2-4  Sample RMF JCL and control statements

**RMF summary report**

You should monitor the same things for Domino servers as for other workloads that require good response times. Areas to consider are CPU, DASD response times, paging, and storage.

The RMF summary report provides a useful overview of the system. An example is shown in Figure 2-5 on page 40.
You should produce a summary report for the whole day and check the following:

- CPU BUSY is not too high.
  
  If the utilization is high (over 80% during peak periods), make sure that the Domino server is set to a high enough priority so that it is not impacted. Also check that the weight for this LPAR has been set high enough.

  If the utilization reaches 100%, you are running more work than the z/OS image can support and workloads are suffering. You may need to define more logical CPs for this LPAR. To ensure that running at 100% does not impact the Domino server, make it a high priority workload.

- DASD average response time is not high, preferably less than ten milliseconds.

- DEMAND PAGING is zero.
  
  If there is a small amount of paging, make sure that the UNIX System Services and Domino address spaces, and the common storage areas, are not impacted.

**RMF OMVS kernel activity report**

An OMVS kernel activity report is included for z/OS UNIX System Services. RMF Monitor III must be active to collect the information for this OMVS report.

This report is useful to see if your values in the z/OS UNIX System Services BPXPRMxx member, in SYS1.PARMLIB, have been set too low. It shows the maximum values you have set, and also minimum, average, and maximum values observed during the RMF reporting interval. It also shows the number of overruns, that is, the number of times processes could not be created or storage was not available.

Figure 2-6 on page 41 is an example of an OMVS kernel activity report. Note that this report was run on our test system and therefore shows very little activity.
You should check periodically that the maximum used values (CURRENT (TOT) in the MAXIMUM column) are well below (MAXIMUM (TOT)) in all the process and interprocess communication and memory map sections. The MAXIMUM (TOT) fields are the values from the active BPXPRMxx in SYS1.PARMLIB. You should not allow any overruns (OVERRUNS (N/S) in the MAXIMUM column) when running a Domino server, because Domino would be unable to start processes or obtain storage.

The RMF reporting interval is typically set at 15 minutes. You are only interested in whether you are exceeding the values in BPXPRMxx and not in how the values change during the day. We suggest that you run an RMF report just for the OMVS kernel activity, specifying a complete day, or more, in the DINTV parameter. The DINTV has a maximum value of 100 hours.

**RMF CPU activity report**

The CPU activity report shows how busy the z/OS image is and also how the various address spaces are being managed.

Figure 2-7 on page 43 is an example of an RMF CPU Activity report.
On this report, pay particular attention to the following values:

- **LPAR BUSY TIME PERC** should be below 90%.
  
  You will have a planned normal utilization for the z/OS image. This should be low enough to handle peaks.
  
  The report shows the individual CPs and a TOTAL/AVERAGE line. You only need to be concerned with the TOTAL/AVERAGE line.

- **MVS BUSY TIME PERC** should be very close to the LPAR BUSY TIME PERC.
  
  If the MVS busy percent is not within one or two percent of the LPAR busy percent, the LPAR is being delayed by PR/SM. The difference between the percentages is the time z/OS believes a task is running but PR/SM has not dispatched the logical CP.
  
  You need to see if this is happening for all your LPARs. If it is, then you need more capacity on the processor. If it is not, then you have the LPAR weight set too low. By increasing the LPAR's weight, this LPAR will receive more CPU at the expense of the other LPARs.

- **NUMBER OF ASIDS IN READY AVG** shows the average number of address spaces that can run concurrently.
  
  This is an indication of the number of logical CPs this LPAR requires. If the LPAR weight has been set too low and work is being delayed, then this value may be artificially low. First give the LPAR the resource it needs and then assess the number of logical CPs needed.

- **NUMBER OF ASIDS OMVS AVG** gives the average number of z/OS UNIX System Services address spaces that were in the system during the interval.
  
  The Domino server address spaces are included in this number. They run as OMVS address spaces.
Figure 2-7  RMF CPU activity report

When running in LPAR mode, make sure the processor is not at high utilization. Also make sure that the LPAR has a sufficient share of the processor so that it is not being constrained by other LPARs. You can see this on the partition data report, shown in Figure 2-8 on page 44.

The recommendations for LPAR settings are discussed in 2.6, “PR/SM™” on page 36.

---

**RMF CPU partition data report**

When running in LPAR mode, make sure the processor is not at high utilization. Also make sure that the LPAR has a sufficient share of the processor so that it is not being constrained by other LPARs. You can see this on the partition data report, shown in Figure 2-8 on page 44.

The recommendations for LPAR settings are discussed in 2.6, “PR/SM™” on page 36.
Create the report at times of peak loading for the processor and check the following:

- **MVS PARTITION NAME**
  This is the name of the LPAR in which the z/OS you are looking at is running.

- **NUMBER OF PHYSICAL PROCESSORS CP**
  This is the number of CPs on the processor on which LPARs running operating systems such as OS/390 and z/OS are dispatched.
  The ICF number can only support Parallel Sysplex® Coupling Facilities.

- **WAIT COMPLETION**
  This should be set to NO.

- **DISPATCH INTERVAL**
  This should be set to DYNAMIC.

- **PHYSICAL PROCESSORS TOTAL (TOTAL line)**
  This shows how busy the processor is. If it is at or near 100 percent, the LPARs will receive resources according to their weights. If the z/OS image is in an LPAR with dedicated engines, you should consider only the LPAR utilization.

- **WGT**
  The weights are used to allocate CPU resources between LPARs that are sharing engines. These weights come into effect when the CPs assigned to shared LPARs are 100 percent busy. An LPAR receives its weight divided by the sum of the weights of the active LPARs. In Figure 2-8, the UNLA LPAR would receive 45/(45+45+10), which is 45 percent of the shared CPs.

- **PROCESSOR NUM**
  This is the number of logical CPs defined for the LPAR. Make sure that you are not constraining an LPAR by giving it too few logical CPs. Figure 2-8 shows we have twelve
CPs on the processor, but the UNLE LPAR has only four CPs specified. So it can never use more than 33% of the processor, even if the other LPARs are idle.

- **CAP DEF**

  Capping is restricting the LPAR CPU to its relative weight, even if there is spare capacity on the processor. We do not recommend capping an LPAR running a Domino server. You could affect the server response time when CPU resource is available by restricting the LPAR to its specified weight.

You can find more information about LPARs in *S/390 PR/SM Planning Guide*, GA22-7236.

**RMF paging activity report**

The key indicator of central storage is the Unreferenced Interval Count (UIC). For 31 bit z/OS images, expanded storage can be monitored using the Migration Age and Expanded Storage Available Frames.

Figure 2-9 shows the RMF paging activity report for a 64 bit z/OS image. Thus it does not have any expanded storage information.

<table>
<thead>
<tr>
<th>CENTRAL STORAGE</th>
<th>LOCAL PAGE DATA SET SLOT COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABLE</td>
<td>AVAILABLE SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>1,079,999</td>
</tr>
<tr>
<td>MAX</td>
<td>1,079,999</td>
</tr>
<tr>
<td>AVG</td>
<td>1,079,999</td>
</tr>
<tr>
<td>CQA</td>
<td>VIO SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>LPA</td>
<td>NON-VIO SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>1</td>
</tr>
<tr>
<td>MAX</td>
<td>1</td>
</tr>
<tr>
<td>AVG</td>
<td>1</td>
</tr>
<tr>
<td>CSA</td>
<td>BAD SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>LSQA</td>
<td>TOTAL SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>1,080,000</td>
</tr>
<tr>
<td>MAX</td>
<td>1,080,000</td>
</tr>
<tr>
<td>AVG</td>
<td>1,080,000</td>
</tr>
<tr>
<td>TOTAL FRAMES</td>
<td>SHARED FRAMES AND SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>CENTRAL STORAGE</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>NUCLEUS</td>
<td>FIXED TOTAL</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>SQA</td>
<td>FIXED BELOW 16 M</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>LPA</td>
<td>AUXILIARY SLOTS</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>CSA</td>
<td>TOTAL</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0</td>
</tr>
<tr>
<td>LSQA</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td></td>
</tr>
<tr>
<td>TOTAL FRAMES</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2-9  RMF paging activity report
You should regularly monitor the following:

- **HIGH UIC AVG**
  
The UIC is a measure of how central storage pages are referenced. This value should be at or near 2540. The higher the number, the better. If it is low, then the z/OS image requires more central storage.

- **MIGR AGE AVG**
  
  This applies to 31 bit z/OS only.
  
The migration age is the length of time in seconds that a page of storage remains in expanded storage before migrating to a page data set. This number should be in the thousands of seconds. A low number means expanded storage is overcommitted, and a very high number means it is underutilized.

  In Workload Manager goal mode, the storage used by address spaces is managed to achieve the velocity or response time goals requested.

- **EXPANDED STORAGE AVAILABLE**
  
  This applies to 31 bit z/OS only.
  
  This shows the minimum, average, and maximum number of expanded storage frames (a frame is 4 KB) which are unused. This is an indication of the real storage you could use to improve the Domino system—by increasing the zFS user cache, for example. Alternatively, it could be kept back for known future growth.

**RMF Workload report**

It is recommended that you define your workloads to manage each Domino partition separately, by assigning each a unique Service Class. Also define a Report Class for each function within each DPAR. This allows you to monitor the resources used by each Domino partition.

Figure 2-10 is an example of the Workload Manager goal mode definitions for a Service Class.

In Workload Manager goal mode, the SYSRPTS(WLMGL(SCLASS,RCLASS)) RMF report gives the CPU used by service classes and report service classes. Figure 2-10 shows an example of a report class.

---

**Figure 2-10 RMF Workload Activity report**

---
Under the Service Rates column, you see the TCB and SRB (which are the CPU seconds used by this workload during the interval), as well as the APPL% (which is the percentage of the interval when the workload was processing). For a Domino server, you may see the APPL% above 100 percent, because the server has multiple threads that can be executing on more than one CP concurrently. This report can be used to obtain the CPU time used by a Domino server and its separate functions.

You can also obtain the central storage (and expanded storage in 31 bit mode) used by the address spaces in this workload group from the same report. However, these are weighted numbers based on the time during the interval that the address spaces were resident (not swapped out). The actual storage required by the address spaces may be higher when all the address spaces are resident.

### 2.7.2 Monitoring z/OS with SMF

System Management Facilities (SMF) is an important source of information on Domino resource use. SMF collects data about CPU, storage, and I/O consumption for address spaces in z/OS.

Details of the SMF records and how to manage them are found in *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

The RMF monitor also writes its records to the SMF data set as record types 70 through 79. The RMF record types are discussed in 2.7.1, “RMF” on page 38.

RMF reports can be created using the RMF postprocessor, but no utility is provided to report on the other SMF records. Many installations have products that can report on SMF data. You can also write a program to produce the desired reports, although the format of SMF records is complicated.

Table 2-6 shows which sections of this redbook present details about some useful SMF record types.

<table>
<thead>
<tr>
<th>SMF record type</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record type 30</td>
<td>“Common address space work (SMF record type 30)” on page 48</td>
</tr>
<tr>
<td>Record type 42</td>
<td>“DFSMS statistics and configuration (SMF record type 42)” on page 49 (HFS only)</td>
</tr>
<tr>
<td>Record type 92</td>
<td>“Open/MVS file system activity (SMF record type 92)” on page 50</td>
</tr>
<tr>
<td>Record type 108</td>
<td>7.1, “SMF type 108 record” on page 128</td>
</tr>
</tbody>
</table>

### Interval recording of SMF record

You have the option to turn on interval recording for address spaces. Without interval recording, a record containing information about the address space is only written when the Domino server is shut down.

With interval recording a record is written every specified number of minutes, giving information about the current interval. An appropriate interval is 15 minutes. With interval recording you can see the resource consumption for an address space over time.

You specify interval recording in the SMFPRMxx member of SYS1.PARMLIB, where xx is the suffix of the currently active member. You can have interval accounting for all workload types,
or you can select specific workload types. You will need interval recording enabled for the OMVS workload type.

Figure 2-11 shows an example of the SMFPRMxx parameters. You should use this as a guide only, since you may require your own values for other parameters. The important parameters are INTVAL, SYNCVAL, and INTERVAL. In the example, the interval is set to 15 minutes, but you can choose the frequency of reporting you require. If a job step starts at 10:50, interval records are written at 11:00, 11:15, 11:30 and so on until the step ends, when a final interval record is written. Each interval record shows the resources consumed since the previous interval record was created.

```
ACTIVE /* ACTIVE SMF RECORDING */
  DSNAME(SYS1.&SYSNAME..MAN1, /* SMF DATA SET NAMES */
  SYS1.&SYSNAME..MAN2, /* SMF DATA SET NAMES */
  SYS1.&SYSNAME..MAN3) /* SMF DATA SET NAMES */
  NOPROMPT /* DON’T PROMPT THE OPERATOR */
  REC(PERM) /* TYPE 17 PERM RECORDS ONLY */
  INTVAL(15) /* RECORDING EVERY 15 MIN */
  INTERVAL(SMF,SYNC) /* TURN ON INTERVAL RECORDING */
  SYNCVAL(00) /* SYNCHRONIZE TO THE HOUR */
  MAXDORM(3000) /* WRITE AN IDLE BUFFER AFTER 30 MIN */
  STATUS(010000) /* WRITE SMF STATS AFTER 1 HOUR */
  JWT(0800) /* 522 AFTER 2 HOURS */
  SID(&SYSNAME(1:4)) /* SYSTEM ID IS &SYSNAME */
  LISTDSN /* LIST DATA SET STATUS AT IPL */
  LASTDS(MSG) /* DEFAULT TO MESSAGE */
  NOBUFFS(MSG) /* DEFAULT TO MESSAGE */
  SYS(TYPE(0:255),EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,IEFUJW,IEFUSI,
  IEFUJP,IEFU50,IEFWJ,IEFUTL,IEFU29),
  NOTYPE(99),INTERVAL(SMF,SYNC),DETAIL)
```

Figure 2-11  SMFPRMxx example for SMF interval recording

With interval synchronization, you can directly compare SMF data with RMF data. We recommend the following options:

- **SMF option:**
  - INTVAL(nn): SMF global recording interval in minutes
  - SYNCVAL(mm): Synchronization with the specified minute of the hour
  - INTERVAL(SMF,SYNC): Write interval records using the INTVAL interval length and synchronized to the SYNCVAL time

- **RMF monitor I option:** SYNC(SMF)
  - SYNC(SMF): Synchronization with SMF. This means that records will be written as specified with INTVAL and SYNCVAL options.

For more information, see the following publications:

- z/OS RMF User’s Guide, SC33-7990
- z/OS MVS System Management Facilities (SMF), SA22-7630

**Common address space work (SMF record type 30)**
The type 30 record contains a lot of information about CPU and storage use by an address space. This record type can be produced at specified intervals during an address space
execution, which is very useful for the long-running Domino address spaces. We recommend that you enable interval recording to allow SMF to write frequent type 30 records.

The type 30 records can be quite large. Therefore, if you are not already collecting them, you should plan on your SMF data sets filling more rapidly and allow for this in your operational procedures.

There are many fields with similar types of data (for example, CPU data) in the type 30 record. The fields that we found useful are:

- **SMF30JBN**: JOB name.
- **SMF30EXN**: Program name of the z/OS UNIX System Services process. *Do not use SMF30PGM, which does not contain the z/OS UNIX System Services program name.* The program name in z/OS UNIX System Services can be up to 16 characters, so you must use SMF30EXN. The accounting chapter in *OS/390 MVS System Management Facilities (SMF)*, GC28-1783 describes the format of the program name.
- **SMF30CPT**: CPU time under a TCB.
- **SMF30CPS**: CPU time under an SRB.
- **SMF30OFR** and **SMF30OFW**: Any z/OS UNIX System Services physical files system (such as zFS or DFSMS) reads and writes to regular files. There are also counts of reads and writes to pipes and special files in the fields SMF30OPR, SMF30OPW, SMF30OSR, and SMF30OSW.
- **SMF30AIS**: Start Subchannel (SSCH). All I/O to the UNIX file system is reported in the OMVS OS/390 UNIX kernel address space. The I/O is performed in cross-memory mode under the TCB of the Domino address space. This field is zero for the z/OS UNIX System Services Domino address spaces. The OMVS address space record contains the number of I/Os performed for all z/OS UNIX System Services users.

### DFSMS statistics and configuration (SMF record type 42)

The data set statistics in the subtype 6 record provide useful information about HFS file systems. The DFSMS file system is stored in a z/OS extended partitioned data set (PDSE). The type 42 record shows the activity for all of the UNIX files stored in that data set.

By default, the records are written at dataset close time. This means from the time the HFS dataset was mounted until it is unmounted. This is not very useful for performance analysis. To get significant statistics for the Domino address spaces, you should have interval recording active. Then the type 42 records are written at the same time as the SMF type 30 records. These records contain useful information about I/O performance, including:

- Number of I/Os
- Average connect, disconnect, pending, and control unit queue time, as well as response time
- Cache performance, including number of cache candidates and cache hit rates

You will find these records useful when you have an I/O performance problem and need greater detail than the DASD volume information provided by RMF.

If you have only one HFS data set on a volume, and there are no other data sets on the volume, then the RMF DASD device and cache reports are sufficient.
Open/MVS file system activity (SMF record type 92)
The type 92 record does not provide interval records, as types 30 and 42 do. Information is reported for the duration of the mount of the file system (zFS or HFS) and for the duration of the open for UNIX files.

Record subtype 5 contains information about a specific UNIX file system. It is written when the file system is unmounted, which should not be a frequent occurrence with a Domino server. Thus there may be no useful information from this record, because the activity it reflects covers a long period of time.

The UNIX file close record, subtype 11, provides information on the requests made to a UNIX file by the z/OS UNIX System Services Domino address spaces. It contains:

- UNIX file name
- Read and write calls
- Directory I/O blocks
- Blocks read and written
- Bytes read and written

Collecting record type 92
Collecting type 92 records, particularly subtypes 10 and 11, generates a very large amount of of SMF records. They can account for more than 90% of your SMF data if collected.

We recommend that you only collect type 92 records when you have a need for this detailed level of information. Only collect them for a short period of time.

You should note that to create these records:
- Subtype 5 must be collecting at the time the UNIX file system is mounted and unmounted.
- Subtype 11 must be collecting at the time the UNIX file is opened and closed.

2.7.3 Monitoring z/OS with Domino statistics
Platform statistics are new for Domino 6, and they provide another way to gain insight into the combined behavior of Domino and z/OS. Platform stats can be thought of as the mirror image to the SMF 108 record: performance information is gathered from z/OS, and stored as Domino statistics that can be collected and processed just like any other Domino stat.

All of the information that serves as the foundation for platform stats comes from RMF Monitor II, and is gathered by default on a 1 minute interval. Statistic names are made up of several components, with a common high-level component of “Platform.” This high-level component is followed by one or more components to make up the full statistic name. Some statistics have a suffix qualifier to indicate some additional attribute of the value, such as peak average, minimum, or maximum.

Table 2-7 is a complete list of the available platform statistics for z/OS.

<table>
<thead>
<tr>
<th>Statistic Name</th>
<th>Meaning</th>
<th>Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform.ActiveNumOfDominoPartitions</td>
<td>Number of Domino partitions on this system image.</td>
<td>none</td>
</tr>
<tr>
<td>Platform.System.PctCombined.CpuUtil</td>
<td>MVS view of CPU utilization. This is how much of the CPU the operating system thinks it is using, and is what would be considered total CPU utilization for other platforms.</td>
<td>average, peak</td>
</tr>
</tbody>
</table>
Note that some statistics contain components of `<instancename>` and `<instancenum>`. These are components that correspond to the name of an add-in server process, and the instance of that add-in. For example, if a Domino server has two instances of the Agent Manager running, the platform statistics for it look like this:

```
Platform.Process.amgr.1.PctCpuUtil = 0.1
Platform.Process.amgr.2.CurrentLocation = 0
Platform.Process.amgr.2.PctCpuUtil = 0
```
Wherever possible, platform stat names are consistent with those of other platforms. There are some statistics which have meaning only in a z/OS environment, and in these cases z/OS shows through. For example, Unreferenced Interval Count is a z/OS-unique value, and this is externalized through the Platform.Memory.PageHighUIC statistic.

**How to interpret the CPU numbers**

Although CPU utilization is often summarized as a single percentage value, the complete picture of CPU utilization is more complex than that. As a result, it is very important to understand exactly what these numbers are saying.

There are three sets of CPU statistics listed in Table 2-7 on page 50 that are worth noting:

- **Platform.System.*.CpuUtil**
  Instantaneous snapshots of CPU utilization at the moment that RMF was queried. These come in two views of the CPU (MVS and LPAR), and are the kinds of measurement that people normally think of when talking about CPU usage.

- **Platform.Process.ActiveDomino.TotalCpuUtil**
  The percentage of all CPU milliseconds used in the last polling interval that Domino consumed. This value tells you how much of the LPAR this Domino server is using, relative to all of the other processes running in the system.

  The percentage of all CPU milliseconds used in the last polling interval that were consumed by this process. The sum of these per-process numbers is the Platform.Process.ActiveDomino.TotalCpuUtil.

The key here is that the Platform.Process.* values do not reflect absolute CPU usage. They are a relative measure, indicating the portion of total CPU cycles used by Domino in any given interval. Also note that these differ fundamentally from the Platform.System.* values, because the system values are instantaneous measurements of the CPU used by the LPAR at the moment the information was collected from RMF. The process values are calculated over the total measurement interval.

The process values are useful for determining if Domino is using the portion of CPU cycles that it should, relative to other work on the LPAR, or if a given Domino process is using more or less CPU than it should, relative to other Domino processes. The system values are useful when judging the amount of CPU capacity available for the LPAR.

**Ways to gather and use platform statistics**

Platform stats can be gathered with the `show stat` command from the Domino server console, or displayed using the Domino administrator client. In particular, the charting functions of the administrator client under the Server → Performance tabs can be very useful tools to watch z/OS performance metrics over time. Domino events and monitors are also very helpful when you want to know when resources become scarce on a given LPAR. See the Domino Administrator 6 help database (help\help6_admin.nsf) for more information.

Additionally, there is a very useful set of health monitoring functions available with a separate license from the IBM Tivoli Analyzer for Lotus Domino product. These functions include an automated health metric that takes into account all of the platform information available, and useful thresholds for these values. The automated assessment is based on heuristics that have been fashioned from experience in the lab and in production environments specific to z/OS. It also contains some sophisticated trend analysis tools that can shed light on long-term patterns of resource usage. For more information, see:

http://www.tivoli.com/products/index/analyzer-lotus-domino
What’s next
For Domino 6.0, platform statistics currently include data on CPU and storage (memory) usage. In release 6.0.2 or later, these may be expanded to include disk and network traffic.
DASD and file system tuning

In this chapter we discuss the performance of the disk Direct Access Storage Device (DASD) subsystem, and how you can monitor and tune it. I/O response time can be a significant part of application response time. Therefore, good DASD response time is an important part of Domino performance.

We also discuss the use of the zSeries File System and its advantages over the DFSMS.

Finally, we explain how to set up the DASD volume structure to support a production Domino server.

First we give some recommendations, then we discuss the subjects in detail.
3.1 Recommendations

These are the key I/O tuning recommendations:

- We recommend the use of zFS instead of DFSMS in your critical file systems used by Domino.
- We recommend the use of z/OS V1R3 or above to get the latest enhancements in zFS.
- Monitor the system: Look for high channel utilization and watch for DASD devices with high activity that are not performing well.
- Use zFS where possible. If not using zFS, carefully follow the recommendations in this chapter. Use D GRS,C to check for latch contention. Monitor HFS datasets for a need to reorganize the data.
- Use high performance DASD subsystems such as the IBM Enterprise Storage Server™ (ESS). This has the additional benefit of allowing you to use Parallel Access Volumes to improve the DASD performance.
- Consider the performance implications of allowing users to index Domino databases.

3.2 DASD subsystem considerations

The zSeries platform supports a broad range of DASD subsystems, from the older 3380 models to the latest Enterprise Storage Server product range. Newer DASD products generally provide better performance characteristics; you should consider these for online workloads such as Domino.

A key element of the DASD subsystem performance is cache storage. Volumes that contain UNIX file systems should be placed on high-performing cached DASD with support for both read and write caching.

3.2.1 The Enterprise Storage Server

The IBM Enterprise Storage Server offers a high-performance DASD subsystem. There are several features that significantly enhance the I/O performance. These include parallel access volumes (PAV), multiple allegiance, I/O priority queueing, and new CCWs.

PAV provides the ability to issue multiple I/O requests within a single z/OS image to the same volume concurrently, potentially eliminating IOS Queue (IOSQ) time. This allows I/Os to more than one database, and multiple I/Os to the same database, at the same time. This is a great advantage when multiple users have their mail databases on the same volume. This also applies when applications have databases on the same volume, and also for busy large databases such as a Name and Address Book (Domino Directory). PAV can have a major benefit to Domino servers because the multi-tasking of a Domino server can produce multiple I/Os to databases concurrently.

PAV can be established within the ESS with a specified number of PAVs to each real volume; this is called static PAV. The preferred method is to allow WLM to manage the distribution of PAVs to real volumes dynamically; this is called dynamic PAV.

Multiple allegiance allows more than one z/OS image to have concurrent access to data as long as there is no data writing conflict. This is only of benefit for volumes where two Domino servers have their data on the same volume, such as file systems from two servers on that volume. All of a Domino server’s databases are dedicated to that server and are not shared.
I/O priority queuing use the WLM goals for the application to be carried through to the I/O subsystem. They will be priority-queued within z/OS, in the processor I/O subsystem and within the ESS.

The ESS stripes a logical volume across the multiple disks in an array, which means that many cache misses to a single volume may be able to be processed in parallel. With PAV and its management by WLM, the need to manually split data across multiple volumes is greatly reduced.

ESS can help Domino in all major areas of performance. We have typically seen improvements of 30 to 50 percent in elapsed time for write-intensive tasks, such as database compaction, replication, and view rebuilding, compared to traditional DASD architectures. Also, end-user response time can be improved by 20 to 30 percent if you use ESS for your databases.

This DASD subsystem can help you integrate Domino with other enterprise applications that use DB2, IMS, and CICS. However, there are certain limits in Domino internal design, as well as in environments outside Domino, that cannot be solved by a DASD subsystem. You can find a more detailed description of potential ESS benefits for Domino at this URL:


Recommendations

- Use ESS in a production environment where possible. Also use WLM-managed PAV to enhance the performance.
- Use as many channel paths as possible.
- Monitor and evaluate collected data from ESS using IBM TotalStorage Expert (formerly known as StorWatch™ Enterprise Storage Server Expert). This tool, together with RMF reports, can help you better understand what is going on behind the scenes.
- Follow the recommendation in “Recommended file system structure for Domino” on page 72 to design a DASD layout for your Domino servers.

3.3 Monitoring DASD I/O activity

You should monitor the DASD I/O activity for your Domino file systems. You can do this at the z/OS level with RMF and SMF, and at the USS level with online statistics.

3.3.1 RMF

Use RMF to monitor the DASD response time for volumes that contain Domino file systems. The RMF summary report that lists the average DASD response time and DASD I/O rate is discussed in “RMF summary report” on page 39.

Other reports provide you with information on:

- Channel path activity
- I/O queueing activity
- Cache subsystem activity
- DASD activity
These reports are useful for identifying performance issues such as:

- Poor response time from a DASD volume. Aim for at most six to ten milliseconds, although it will depend on the activity rate and the DASD device type.
- Too much activity on a single DASD volume.
- Channels that are too busy. That is, more than 40 percent busy for ESCON® channels with four paths to a device.
- Insufficient paths. We recommend at least four paths to every control unit. Define the maximum allowed by the DASD subsystem to a z/OS image.

**Monitoring channel paths**

The RMF Channel Path Activity report shows the channel utilization of all the channels. It gives the utilization from this LPAR in the PART column and the utilization from all LPARs in the processor in the TOTAL column. It is important to look at the TOTAL utilization from all LPARs when assessing if a channel is too busy.

You can also see the utilization of the OSA channels, as can be seen in the sample report shown in Figure 3-1.

---

**Figure 3-1  RMF channel path activity report**

---

**Monitoring I/O queueing**

The RMF I/O Queueing Activity report shows the I/O queueing information to control units. It shows channel paths to the control unit in the CHAN PATHS column; the CONTENTION RATE column gives a measure of the contention for channel paths. The higher the contention rate value the more contention there is for paths. The DELAY Q LGNTH shows the average queue length for access to the channel path. Figure 3-2 on page 59 shows an example of an RMF I/O Queueing Activity report.
Monitoring DASD cache

The RMF Cache Subsystem Activity report shows the use of the cache in the DASD controller and also, by volume, the effectiveness of the caching for each volume. A sample of this report is shown in Figure 3-3 on page 60.

In the Cache Subsystem Overview, for Read I/O Requests, the hit rate (H/R) for normal reads and for sequential reads is shown. The nearer to 1.000 it is, the better. For example, a normal read hit rate of 0.923 indicates that 92.3% of normal reads found the data in the cache, and 7.7% had to wait for the data to be retrieved from the disks.

At the volume level, you can see the read hit rate (READ H/R) and write hit rate (WRITE H/R). You can also see percentage of the I/O requests that were reads (% READ).

If you are not getting good read hit rates, then you may need more cache. Also you may have fragmentation within the file system, so a reorganization may improve performance. To reorganize a file system, allocate a new file system on a new volume, copy the old to the new using a utility such as Copytree, then unmount the old file system and mount the new file system in its place. This tip applies to both zFS and HFS.

It may be that the nature of your data access will not allow good read cache hit rates. The write cache hit rate should always be 100% on an ESS subsystem.

The I/O rates that you see on this report may differ slightly from the corresponding I/O rates on the RMF Direct Access Device Activity report. This is because the DASD subsystem has a different way of counting I/Os from that used by z/OS. This should not be a concern.

The RMF cache subsystem activity report and the IDCAMS LISTDATA STATUS command are useful tools to monitor cache use and effectiveness.
**Figure 3-3** RMF cache subsystem activity report

### CACHE SUBSYSTEM STATUS

#### SUBSYSTEM STORAGE

<table>
<thead>
<tr>
<th>Status</th>
<th>Configured</th>
<th>Available</th>
<th>Pinned</th>
<th>Offline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Volatile Storage</td>
<td>256.0M</td>
<td>256.0M</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Volatile Storage</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

#### CACHE FAST WRITE

- Configured: - ACTIVE
- Available: - ACTIVE

#### IML DEVICE AVAILABLE
- YES

### CACHE SUBSYSTEM OVERVIEW

#### TOTAL I/O

- TOTAL I/O: 56652
- CACHE I/O: 56651
- CACHE OFFLINE: 0

#### READ I/O REQUESTS

- NORMAL: 55082 (15.3%)
- SEQUENTIAL: 1566 (0.4%)
- CFW DATA: 0 (0.0%)

#### WRITE I/O REQUESTS

- NORMAL: 2 (0.0%)
- SEQUENTIAL: 52 (0.0%)
- CFW DATA: 0 (0.0%)

#### CACHE MISSES

- DFW BYPASS: 0 (0.0%)
- NORMAL: 4247 (1.2%)
- SEQUENTIAL: 52 (0.0%)
- CFW DATA: 0 (0.0%)

### CACHE SUBSYSTEM DEVICE OVERVIEW

#### VOLUME

<table>
<thead>
<tr>
<th>Volume</th>
<th>Dev</th>
<th>RRID</th>
<th>% I/O</th>
<th>I/O Rate</th>
<th>---CACHE HIT RATE---</th>
<th>---DASD I/O RATE---</th>
<th>ASYNC</th>
<th>TOTAL</th>
<th>READ</th>
<th>WRITE</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERIAl</td>
<td>NUM</td>
<td>I/O Rate</td>
<td>READ</td>
<td>DFW</td>
<td>CFw</td>
<td>STAGE</td>
<td>DWBP</td>
<td>ICL</td>
<td>BYP</td>
<td>OTHER</td>
<td>RATE</td>
</tr>
<tr>
<td>*ALL</td>
<td>100.0</td>
<td>15.7</td>
<td>14.6</td>
<td>0.0</td>
<td>0.0</td>
<td>1.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>*CACHE-OFF</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>*CACHE</td>
<td>100.0</td>
<td>15.7</td>
<td>14.6</td>
<td>0.0</td>
<td>0.0</td>
<td>1.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XCL003</td>
<td>2F85</td>
<td>N/A</td>
<td>99.2</td>
<td>15.6</td>
<td>14.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XAV94</td>
<td>2F94</td>
<td>N/A</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XAV95</td>
<td>2F95</td>
<td>N/A</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XAV96</td>
<td>2F96</td>
<td>N/A</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XAV97</td>
<td>2F97</td>
<td>N/A</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XAV98</td>
<td>2F98</td>
<td>N/A</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XAV99</td>
<td>2F99</td>
<td>N/A</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Monitoring DASD devices

The RMF Direct Access Device Activity Report shows the activity rate and response times for DASD devices. A sample is shown in Figure 3-4.

The response time, in milliseconds, is split up into its separate components. The AVG IOSQ TIME column shows the time spent waiting for access to the DASD volume in z/OS. If this value is a significant proportion of the response time, then PAV would provide a major benefit.

The disconnect time (AVG DISC TIME column) includes the time waiting for data to be moved from the disk drives to the cache in a cache miss. As we discuss in “File system reorganizations” on page 66, an increase in the disconnect time can indicate that the file system needs to be reorganized to place UNIX files into contiguous areas on the disk. Disconnect time also includes the time waiting for a free channel to reconnect to when sending data back to the processor, so high channel utilization can increase the disconnect time.

A high % DEV UTIL indicates that the DASD volume is very busy. This could be because there is a high activity rate to the volume (DEVICE ACTIVITY RATE) or there is a long average response time (AVG RESP TIME), or a combination of both. You need to improve the response time or move some of the data to a less busy volume to improve performance.

---

**Figure 3-4  RMF direct access device activity report**

### 3.3.2 SMF record types

SMF provides you with more detailed information on I/O activity. See 2.7.2, “Monitoring z/OS with SMF” on page 47 for details about the SMF records.
Record type 42
SMF record type 42, subtype 6 provides information about HFS dataset performance. I/O statistics are provided for each HFS dataset. These are useful when you have more than one HFS dataset on a volume. If you have only one HFS dataset on a volume, then the RMF cache report and the RMF DASD device report should give you sufficient information.

Record type 30
SMF record type 30 reports activity on a job and job step basis. The z/OS UNIX System Services process section also provides information on file system lookups.

Record type 92
SMF record type 92, subtype 5 collects information at the time the file system is mounted and unmounted. SMF record type 92, subtype 11 collects information at the time the file system is opened and closed.

Record type 92s create a large amount of SMF data for a Domino server. We recommend that you do not collect these unless you have a problem and need the level of information they provide. If needed, you should turn on their collection for a short period of time and then turn it off again.

3.3.3 UNIX statistics

You also can obtain online statistics at the UNIX level by issuing the command:

/usr/sbin/kerninfo stats

The results are shown in Figure 3-5 on page 63.
3.4 UNIX file system

z/OS provides the option of two file systems for applications that use the UNIX interfaces: zFS and DFSMS HFS. We concentrate on the newer zFS and provide setup recommendations for the DFSMS.

3.5 DFSMS recommendations

Although the recommendation is to use the zFS as the UNIX file system for Domino, there are still users of DFSMS. Lotus Domino for S/390 Release 5: Performance Tuning and Capacity Planning, SG24-5149 contains details on the management of DFSMS. Since that redbook was published, we have gained access to more customer experiences with the different tuning options applicable to Domino. In this section we summarize our recommendations based on those experiences.

3.5.1 Buffer pool sizes

There are two parameters controlling the DFSMS buffer pool. These are the virtual storage size (VIRTUAL) and the fixed storage size (FIXED). They are set in BPXPRMxx and can be modified using the CONFIHFS OMVS shell command.

Experience has shown that letting the VIRTUAL parameter default to half the central storage in the z/OS image is a good recommendation. In practice DFSMS uses about 400 to 500 MB for this buffer pool in a 31-bit environment with two large Domino mail DPARs.

If you constrain the size of the buffer pool too much, then metadata gets flushed out of the buffers and there is an increase in the exclusive locks taken on HFS datasets to manage the buffer shortage. All these slow down the Domino server and increase its CPU use.

The recommendation is to set the FIXED value to the default of zero. This means that there are no long-term fixed buffers and the buffers need to be fixed before performing I/O. There is a small cost in fixing the pages but this should not be significant. By setting to zero, central storage is not retained by DFSMS when it is not needed and it can be used by other address spaces.

These are the only tuning options that directly affect DFSMS buffering of data.

3.5.2 Sync daemon interval

This should be left at the default value of 60 seconds. Domino performs many fsync requests to guarantee data has been hardened to disk. This means that in practice most data is written more often than the sync daemon interval.

3.5.3 File system space monitoring

It is very important that you monitor the free space in your file systems.

In DFSMS, the application cannot write to a full file system. If your file system is filled, it will become unusable.

It has also been found that an HFS dataset performs better when it is kept below 80 to 85 percent full. This is because the HFS dataset does not become so fragmented and there is not the extra effort required searching for free space. When the HFS dataset becomes 85 percent full, you should move some files to another HFS dataset that has more free space.

A function was provided with APAR OW44631 to monitor how full an HFS dataset is. The parameter:

\[ FSFULL(threshold, increment) \]

is specified in BPXPRMxx and can be set for all HFS file systems. It can also be specified on the MOUNT statement for a specific HFS dataset. The threshold is the percentage used when...
the message appears and the increment is the percentage increment for issuing a new message. The message will be removed if any of these three events occurs:

- The threshold is increased above the current used
- Files are deleted to bring the space used below the threshold
- The operator deletes the message

The message issued is:

IGW023A HFS dsname EXCEEDS percentage% FULL

3.5.4 Shared HFS

Shared HFS allows sharing of HFS files across a Parallel Sysplex. This feature can be used to give backup servers access to production files. You should, of course, not allow multiple Domino servers to actively share the same files in a production mode.

3.5.5 DFSMS locking

DFSMS uses latches to synchronize updates to its HFS files. It frequently takes an exclusive latch on a whole HFS dataset while it updates essential data within the HFS dataset. During this exclusive lock, other requests cannot access any files (even for read) in that HFS dataset.

The time spent waiting for a latch is not measured in any RMF reports. In fact, you may have very good DASD response times but Domino may be waiting for a long time for its I/O requests to be satisfied.

The only way to monitor latch contention is to issue the D GRS,C operator command. This command should be issued frequently, every 5 or 10 minutes, by your automation procedures. You can then refer to the console log if you suspect you have latch contention. The message issued in response to the command is ISG343I.

An example of the ISG343I message for contention on a resource is shown in Figure 3-6.

If you are having latch contention, then you need to:

- Improve the response time to the DASD volume containing the HFS file.
- Move some high activity files to another HFS dataset.
- Use smaller HFS datasets, particularly if you have implemented multivolume support.
- Migrate the HFS dataset to zFS which allows more concurrent access.

```
ISG343I 12.22.45 GRS STATUS 660
NO ENQ RESOURCE CONTENTION EXISTS
LATCH SET NAME: SYS.BPX.A000.FSLIT.FILESYS.LSN.01
CREATOR JOBNAME: OMVS      CREATOR ASID: 000E
LATCH NUMBER: 6887
REQUESTOR ASID EXC/SHR OWN/WAIT
MAIL1SER   009A  EXCLUSIVE  OWN
MAIL1SER   009A  EXCLUSIVE  WAIT
```

Figure 3-6  D GRS,C response
3.5.6 File system reorganizations

UNIX file system data is not necessarily contiguous within its underlying store. An updated block is not written back in the same place but is written in a free block, and the pointers within the metadata are changed to reflect the new placement.

Over time, data in a large file becomes widely distributed across the file system. In fact, with multivolume file systems it would be spread across all the volumes. This spreading of data can lead to inefficient DASD caching with low read cache hit rates. More tracks have to be transferred from disk to satisfy the I/O read request.

The need to reorganize the file system (for example, by `copytree` to a new file system) can be detected by an increase in the DASD volume's disconnect time portion of the response time in the RMF Direct Access Device Activity report, or reduction in the cache read hit rate in the RMF Cache Subsystem Activity report, as described previously.

The `copytree` tool can be found by clicking the `copytree` link at:


3.6 Multivolume file system

Both DFSMS and zFS support multivolume file systems. The most important benefit is the capacity to allocate more space per file system. But care must be taken, or performance could degrade.

This section discusses how to improve performance on multivolume file system datasets, including recommendations on tuning the appropriate parameters.

3.6.1 Recommendations

Here are some recommendations to get better performance with a multivolume file system:

- Use the IBM Enterprise Storage Server (ESS) for a significant improvement.
  You should implement Parallel Access Volumes (PAV) to improve DASD response times.
- Implement zFS to improve locking and also attain better response times through buffering.
- Avoid using a large number of volumes with multivolume file system datasets. It is preferable to define 20 or 25 GB per file system. Several file systems with 8 or 9 volumes are better for performance than one file system with 25 or 30 volumes.

With DFSMS, plan for up to ten 3390 model 3 equivalents for low activity Domino mail HFS datasets and up to four 3390 model 3 equivalents for high activity Domino mail or applications HFS datasets.

- If you need more space for the Notes data directory, you have two options for optimizing performance:
  - Allocate a single file system on an IBM 3390 model 9 volume. For example, you can allocate a file system with this mount point:
    /notesdata/mail1;/notesdata/mail2;/notesdata/mail3;/notesdata/mail4
  - Make a multivolume file system on an IBM 3390 model 3 with a few volumes, under the Notes data directory, and place your database in this file system.
  - Make your transaction log file system as small as possible. We do not recommend using a multivolume file system for this log.
Do not use a multivolume file system unless it is necessary. It is more efficient to use several single file system files on IBM 3390 model 9 volumes (8.4 GB). Or, use the ESS and you will have the best performance possible.

Domino for z/OS is a UNIX application running on z/OS UNIX System Services, so it is important to organize your UNIX file system directory structure for performance. You only need a multivolume file system if your database increases in size constantly.

With multivolume capability, which allows you to allocate one super file system with 59 volumes, it is possible to think, “I just need one or two large files for all my data.” However, this is not the best way to handle your file structure. A small file system is faster and easier to manage, while a large file structure is difficult to maintain and makes it difficult to back up data. If you lose just one volume, you have lost everything.

### 3.7 zSeries File System (zFS)

The zSeries File System (zFS) is a z/OS UNIX System Services file system that can be used in addition to the DFSMS File System.

zFS provides significant performance gains in accessing large files that are frequently accessed and updated. zFS also provides reduced exposure to loss of updates by writing data blocks asynchronously and not waiting for a sync interval. In addition, the granular zFS locking scheme allows for large performance improvements. More information about zFS can be found in [z/OS Distributed File Service zSeries File System Implementation](https://www.ibm.com/docs/en/zos/2.5.0?topic=files-zseries-fs), SG24-6580.

#### 3.7.1 zFS performance

Most of the zFS performance improvement comes from its data caching capability. DFSMS only caches files up to 1 MB in size. This means that nearly all the Domino files cannot make effective use of the caching. zFS not only caches small files, it is also able to cache large files such as Domino databases. The effectiveness of the cache will be determined by the space that you are able to define for its cache size. By migrating from DFSMS to zFS, you can have improved performance by higher cache hits that avoid I/Os to DASD.

zFS also has a different locking mechanism from DFSMS, allowing more concurrent access to the data. DFSMS took frequent exclusive locks on the whole HFS dataset while performing updates. This prevented other read and write accesses to any file in the same HFS dataset. This produced delays for HFS users waiting for the exclusive latch.

zFS does not use this latch mechanism and does not take frequent exclusive locks on the whole HFS dataset. This allows a lot more concurrency of access to files within the HFS dataset.

A laboratory test of 1,000 simulated Webmail users using Domino R5 for S/390 showed a major reduction in DASD I/Os. Figure 3-7 on page 68 shows the average I/O savings for 24 mail volumes. This reduction is because of the better data buffering in zFS.

A typical saving of 65% of the I/Os has been achieved by converting from DFSMS to zFS.
zFS has shown a very large performance increase for sequential processing. zFS has a sequential detection mechanism which enables read-ahead in sequential mode. This is of significant benefit to Domino sequential processes, such as compact and creating full text indexes (FTI) and backups.

Table 3-1 displays the results of measurements performed on the IBM US Name and Address Book, which was 1.2 GB in size with 191,000+ users, 1,850+ groups and 1,900+ connection documents. You can see that there are considerable savings in the time to perform these functions.

Table 3-1 Sample utility timings

<table>
<thead>
<tr>
<th>Function</th>
<th>HFS Elapsed Time (sec)</th>
<th>zFS Elapsed Time (sec)</th>
<th>Difference</th>
<th>HFS : zFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>compact -c</td>
<td>2,631</td>
<td>1,065</td>
<td>-60%</td>
<td>2.47 : 1</td>
</tr>
<tr>
<td>compact -b</td>
<td>405</td>
<td>166</td>
<td>-59%</td>
<td>2.44 : 1</td>
</tr>
<tr>
<td>compact -B</td>
<td>3,179</td>
<td>1,548</td>
<td>-51%</td>
<td>2.05 : 1</td>
</tr>
<tr>
<td>updall -RC</td>
<td>17,553</td>
<td>5,490</td>
<td>-69%</td>
<td>3.20 : 1</td>
</tr>
<tr>
<td>create FTI</td>
<td>4,463</td>
<td>3,300</td>
<td>-26%</td>
<td>1.35 : 1</td>
</tr>
</tbody>
</table>

3.7.2 zFS tuning

In this section we look at the different caches used within zFS and the best way to tune them. These recommendations are based on the large Domino database sizes, and would be different for other applications which have small HFS files.

User file cache

The user file cache is used to cache all user files that are 7 KB or larger. Files smaller than 7 KB are held within the metadata and are cached in the metadata cache.

Because of the large size of Domino files, the zFS user cache is the most important cache for Domino performance. Its size is defined by the user_cache_size parameter in the IOFSPRM file. The default is 256 MB. This cache is allocated in data spaces and to avoid paging you will
need sufficient real storage to support your user cache size. If you start to page to auxiliary
page datasets, then you may see worse performance than with DFSMS. If you are
constrained for real storage, then you will have to compromise between good zFS
performance and real storage use.

To set an efficient user cache size, monitor your cache hit ratio. You should look to achieve a
hit ratio of over 90%; the higher the better. It is very difficult to maintain a very high hit rate in
the zFS user cache that handles Domino mail files, because they usually are very large and
you may need to give an unacceptably large amount of real storage to achieve a very high hit
rate. With zFS there is the potential to improve the processing of the Domino data directory
and the transactional log files.

You may wish to start with the default 256 MB of user file cache and change as your
monitoring indicates.

To tune the user file cache hit ratio, you can use the modify command with the query
parameter and specify the VM report option. This command can be issued on your operator
console. For more details about this command, see z/OS V1R4.0 Distributed File Service
zSeries File System Administration, SC24-5989. The format of the command is:

MODIFY ZFS,QUERY,VM

Figure 3-8 is an example of the output from a modify command.

<table>
<thead>
<tr>
<th>User File (VM) Caching System Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>External Requests:</td>
</tr>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Reads 0 Fsyncs 0 Opens 0</td>
</tr>
<tr>
<td>Writes 0 Setattrs 0 Unmaps 0</td>
</tr>
<tr>
<td>Asy Reads 0 Getattrs 0 Schedules 0</td>
</tr>
<tr>
<td>Flushes 0</td>
</tr>
<tr>
<td>File System Reads:</td>
</tr>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Reads Faulted 0 (Fault Ratio 0.06%)</td>
</tr>
<tr>
<td>Writes Faulted 0 (Fault Ratio 0.00%)</td>
</tr>
<tr>
<td>Read Waits 0 (Wait Ratio 0.00%)</td>
</tr>
<tr>
<td>Total Reads 0</td>
</tr>
<tr>
<td>File System Writes:</td>
</tr>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Scheduled Writes 0 Sync Waits 0</td>
</tr>
<tr>
<td>Error Writes 0 Error Waits 0</td>
</tr>
<tr>
<td>Scheduled deletes 0</td>
</tr>
<tr>
<td>Page Reclaim Writes 0 Reclaim Waits 0</td>
</tr>
<tr>
<td>Write Waits 0 (Wait Ratio 0.00%)</td>
</tr>
<tr>
<td>File Management: (File struct size=160)</td>
</tr>
</tbody>
</table>

Figure 3-8  Output from the MODIFY ZFS,QUERY, VM command

This command gives you a lot of information about the user file cache, including the fault ratio.
To get the hit ratio, subtract the fault ratio from 100%. That is:

hit ratio = (100% - fault ratio)
If your hit ratio is low, you should consider increasing the user file cache size to accommodate more data. To do this, you must alter the IOEFSRM file and then restart the zFS. As was mentioned previously, if you define too large a user file cache, it may use too much real memory, and your system will start to page.

The `user_cache_size` parameter can be dynamically changed in z/OS 1.4 with the command `zfsadm config`. For more information about this command, see `z/OS V1R4.0 Distributed File Service zSeries File System Administration`, SC24-5989.

**Metadata cache**

The metadata cache is used to contain all file system metadata. This includes all directory contents, file status information, file system structures, and also caching of data for files smaller than 7 KB. Metadata is referred to and updated frequently for most zFS file operations, hence achieving a good hit ratio is often essential to good performance for most workloads. A good hit ratio may be considered to be 90% or more depending on your workload.

The metadata cache is stored in the zFS primary address space and its default size is 32 MB, but you can change it by changing the `meta_cache_size` parameter. Since the metadata cache only contains metadata and data for small files, it does not need to be nearly as large as the user file cache. The operator `MODIFY ZFS,QUERY,LFS` command output gives statistics for the metadata cache, including the cache hit ratio. The output from this command is quite large, so you will have to search for the Metadata Caching Statistics section. You can see in Figure 3-9 what it looks like.

```
<table>
<thead>
<tr>
<th>Buffers (K bytes)</th>
<th>Requests</th>
<th>Hits</th>
<th>Ratio</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096</td>
<td>32768</td>
<td>57</td>
<td>22</td>
<td>5</td>
</tr>
</tbody>
</table>
```

*Figure 3-9  Metadata cache part of the output from MODIFY ZFS,QUERY,LFS*

To tune this parameter, you should follow the same methodology as for the user file cache.

For Domino, the metadata cache should be sufficiently large with its default setting of 32 MB. You should not have to tune this cache as much as the zFS user cache.

**Log file cache**

Every zFS aggregate contains a log file used to record transactions describing changes to the file system structure. An aggregate is equivalent to an HFS dataset. zFS uses a linear dataset to store its data and thus the linear dataset is an aggregate.

The log file cache is a pool of 8 K buffers used to contain log file updates. Log file buffers are always written asynchronously to disk, and normally only need to be waited upon when the log is becoming full, or if a file is being fsync’ed.

The log file cache is stored in a data space and its default is 64 MB. The log file cache can grow dynamically by adding one 8 K buffer for each attached aggregate. This ensures each aggregate always has one 8 K buffer to use to record its most recent changes to file system metadata. Since log files are written asynchronously, the cache essentially allows write-behind of log files; and since the cache is shared among all aggregates, aggregates that have a higher write rate use more buffers in the cache using a least-recently-used (LRU) algorithm.
The log file cache is a write-only cache, so a read hit ratio is not relevant. However the operator `MODIFY ZFS,QUERY,ALL` command does show log file I/O rates and I/O waits. It is desirable to make the log file cache large enough that log file I/O waits do not occur too frequently compared to the log file I/O rates. To do that, you must change the `log_cache_size` parameter in the IOEFSPRM file.

However, every workload is different. Domino for example, usually issues a lot of `fsync` operations, forcing zFS to sync the log file more frequently. So, you will probably not need a very large log file cache.

**Vnode cache**

Every object in the zFS file system is represented by a data structure, called a vnode, in memory. zFS keeps a cache of these and recycles these vnodes in an LRU fashion.

Every operation in zFS requires a vnode, and z/OS UNIX System Services keeps pointers to zFS vnodes. Since z/OS UNIX System Services keeps references to zFS vnodes, zFS may be forced to dynamically increase the size of this cache to meet the demands of z/OS UNIX System Services.

To create a zFS vnode for a newly referenced file, or a newly created file for a user, requires the pathlength to initialize the structure and obtain its status information from the metadata cache. If the file status is not in the metadata cache, then disk I/O may also be required.

The vnode cache is stored in the zFS primary address space and the default number of vnodes is 16384. As with any cache, a good hit ratio is desirable; the operator `MODIFY ZFS,QUERY,LFS` command shows the vnode cache hit ratio in the LFS Vnode Cache Statistics section. Figure 3-10 shows an example of the Vnode Cache Statistics section.

<table>
<thead>
<tr>
<th>LFS Vnode Cache Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vnodes Requests Hits Ratio Allocates Deletes</td>
</tr>
<tr>
<td>16384 13 12 92.3% 0 0</td>
</tr>
</tbody>
</table>

LFS Vnode structure size: 296 bytes
SAF Access Cache Requests: 0 hits: 0 (hit ratio 0.0%)

*Figure 3-10  Vnode cache part of the output from MODIFY ZFS,QUERY,LFS*

Since the vnode cache is essentially backed by the metadata cache, if the vnode hit ratio is low but the metadata cache hit ratio is high, your performance may not suffer too much, since a vnode cache miss requires only a short pathlength to initialize the vnode structures.

### 3.7.3 zFS locking improvements

zFS provided availability improvements in z/OS 1.6. In most cases, internal zFS error situations will only mark a single file system un-writable until un-mount or mount, as opposed to all zFS files systems being marked un-writeable, as in pre-z/OS 1.6 environments.

zFS locking improvements were made in the user interface code on z/OS V1R3. This improved the throughput rate for measurements with a high number of concurrent users reading or writing the same file. Compared with z/OS V1R2, the reads per second and writes per second improved 95% in some cases, such as if two threads were to write to the same file at the same time.
Because of Domino’s characteristics, this kind of improvement is very important, especially when working with databases such as names.nsf that are utilized by almost all functions of the server. This kind of improvement would avoid database contention and cause a better response time for the server.

3.7.4 Migrating data from HFS to zFS

You can migrate from DFSMS to zFS one HFS dataset at a time. The current restriction is that the root directory must remain as a DFSMS file system.

Using the OMVS pax command

You can copy data from an DFSMS file system to a zFS file system by using the OMVS pax command, which uses an intermediate archive file. Refer to z/OS: UNIX System Services Command Reference, SA22-7802, for more information on the pax command. When the data is being copied, the file systems being accessed must be mounted and the Domino server must be down.

Using an intermediate archive file

Use the pax command to copy the source (DFSMS) file system into an intermediate archive file and then use the pax command to copy from the archive file into the target (zFS) file system. This archive file can be a z/OS UNIX System Services file or it can be a z/OS dataset. In this example we migrate only the /notesdata directory, but this same process can be done for all Domino file systems. Issue the following commands from OMVS:

1. Position to the source (DFSMS) file system mounted at /notesdata
   ```bash
cd /notesdata
   
   2. Create a z/OS UNIX System Services archive file called /tmp/zfs1.pax that contains the DFSMS mounted at /notesdata
   ```
   ```bash
   pax -wvf /tmp/zfs1.pax
   
   3. Unmount the old (DFSMS) from the /notesdata mount point and mount the new (zFS) at /notesdata
   ```
   ```bash
   
   4. Position to the (zFS) file system mounted at /notesdata
   ```
   ```bash
cd /notesdata
   
   5. Read the archive file into the zFS file system mounted at /notesdata
   ```
   ```bash
   pax -rvf /tmp/zfs1.pax
   ```

3.8 Recommended file system structure for Domino

Before you install Domino 6 for z/OS, you should allocate three file systems and mount them in these directories:

- /notesdata - control files, databases and templates
- /notesdata/mail - Notes mail
- /usr/lpp/lotus - Product code modules

This file structure is intended for initial installation and not for production Domino servers. To create a production Domino server, you should change this structure, to allow for files to grow and deliver good performance to your Domino databases and system files.
Figure 3-11 on page 73 shows an example of a file system structure, designed to maximize Domino performance and space allocation. This is an example of the structure needed to support a large Domino mail DPAR.

If your are constrained by the availability of DASD volumes then you should, as a minimum, place each DPAR's /notesdata, /names (Domino directory) and /log (Notes log file) on their own volumes. You will need to monitor the performance of the other Domino files to see if they merit moving to separate volumes.

With zFS, especially considering its locking improvements, it may be possible to have more Domino files on the same volume.

We recommend that you create a separate directory called /notesdata/names and a new file system that you will mount in this new directory. Put your names.nsf (Domino directory) file in this new file system. The size of your new file system will depend on the size of your names.nsf file. Make sure to create a file system big enough to handle the file's natural growth. To redirect the access from your old file location to the new one, you can choose between using a database link or a UNIX link. To create a UNIX link for this example, issue this command:

```
ln /notesdata/names/names.nsf /notesdata/names.nsf
```

To reduce the activity on your /notesdata file system, you can repeat this same process with your mail box files and the log.nsf file. In Figure 3-11, they were placed, respectively, in the /notesdata/mailbox directory and /notesdata/log directory. If you have more than one mail.box file, it may not be necessary to create one file system for each one of them. They can be placed in the same file system.
The transaction log also must be placed in a different file system. In Figure 3-11, it was placed in the /notesdata/trlog directory.

The /notesdata/noteslog directory in Figure 3-11 is, by default, the directory used by the z/OS Console Support for Domino (DomCon) to place its log files and some control files. These files should be placed in a separated file system. If it shares the same file system with the /notesdata directory, the natural growth of the DomCon logs can use all the space available in the file system and cause a server crash.

If your Domino server crashes for some reason, it will probably create a CEEDUMP file and some NSD logs. These files are useful to determine the reason for the crash. By default, the NSD logs and the CEEDUMP are created in the /notesdata directory. To avoid space problems with this file system that can result in a server crash, you should redirect the creation of the files to another directory and file system. In our example, we created the /notesdata/dump directory, and mounted a new file system on it. To redirect the creation of those files, use these two variables:

```
NSD_LOGDIR
_CEE_DUMPTARG
```

Both should be exported by the Domino user, or by the DomCon user if you are using z/OS Console Support for Domino.

We also changed the path used for the creation of the temporary Domino files by using the notes.ini variable `TMP`. This variable must be a relative path of the directory, based on the server /notesdata directory. In our example, it was set like this:

```
TMP=dump
```

By default, Domino also creates a mail directory under the notesdata directory. We recommend that you create more file systems in new mail directories, based on your number of users and mail database sizes. The same may be done when working with applications.

We also recommend that you restrict the number of volumes in a multivolume file system. If the number of volumes in a file system is too large, it may become difficult to manage and your response time can increase. This was discussed in 3.6, “Multivolume file system” on page 66.

Remember to specify Debug_outfile if you need to set debug notes.ini parameters. But keep in mind that the output size is unpredictable.

As an example, a customer is using the DEBUG_OUTFILE=debug.out parameter in NOTES.INI to capture debugging information. The LotusScript agent is running information (in conjunction with LOG_AGENTMANAGER=1 and DEBUG_AMGR=*), but the debug.out file grows at the rate of more than 200 MB per day and puts a strain on DASD free space.

We recommend the output file be separate from the /notesdata and system file systems.
Network tuning

All client types are connected to a Domino server on zSeries through a TCP/IP network. In this chapter we discuss the performance of the TCP/IP network and how you can monitor and tune it.

Note: When deploying a Domino server on z/OS, TCP/IP is the only network protocol supported.

A client-server application such as Lotus Notes/Domino is designed so that parts of the application run on the client and parts run on the server. These parts must communicate with each other, and this is done through a set of network resources (hardware and software). The challenge is to ensure good performance along the entire path, as well as consistent response time to the user. Network performance is an important part of Domino performance.
4.1 Recommendations

Here are some key network tuning recommendations:

- Consider the whole network.
  The network consists of many hardware boxes and links. Monitor all the components in
  the network and address bottlenecks.

- Define REGION=0M for the TCP/IP region size.
  TCP/IP development recommends setting the region size at 0. TCP/IP buffers are
dynamically allocated in the private area. REGION=0M allows TCP/IP to allocate the
maximum virtual storage above and below the 16 MB line. If you have implemented an
IEFUSI exit, check that it will allow the specification of 0M for the region size.

  If you use the sample TCP/IP start procedure available in the SEZAINST sample library,
you will find that releases prior to V2R8 include a sample procedure with region size of
7500 KB. Ignore this setting and use 0 instead. Remember to consult the INFOAPARS
II11710, II11711, and II11712 for this and further tuning information.

4.2 TCP/IP

Lotus Domino for z/OS uses the TCP/IP network protocol to communicate with any kind of
client. This section discusses TCP/IP considerations on z/OS.

4.2.1 Monitoring TCP/IP

TCP/IP provides several commands that are useful for monitoring the system and highlighting
problems. In particular, we discuss:

- PING
- TRACERTE
- NETSTAT
- TCPIPSTATISTICS

You can also use Spool Display and Search Facility (SDSF) and Resource Measurement
Facility (RMF) to monitor TCP/IP at the z/OS level. Monitor CPU usage by the TCP/IP
address space, paging, and I/O rates for communication devices, for example. See 2.7.1,
“RMF” on page 38.

In addition, you should use a network monitor product for more detailed monitoring, tuning,
and troubleshooting.

PING

The PING command sends an echo to an IP host to determine whether the host is
accessible. The response shows how long the round trip message took. This is the first tool to
use to identify connectivity problems, and it can also show performance problems.

The PING command can be issued from any system that supports TCP/IP, including a Notes
workstation client or a Web browser, where you would issue it from a command prompt. PING
can also be issued from z/OS TSO, as shown in Figure 4-1 on page 77.
In this example, the PING command is sending 481 bytes of data five times. The time taken for the data to be sent to the remote host with IP address 9.12.2.17 and then returned to this system was 0.0184 seconds. High values for the response time can show performance problems en route between the issuing system and the target system.

TRACERTE
The TRACERTE command is similar to the PING command except that it reports the round trip time by intermediate hosts. It is helpful to check how each connection between the issuing system and the target system is performing and to determine bottlenecks along the route. You must be defined in the OBEY statement of the TCP/IP profile data set to issue this command. Figure 4-2 shows an example of this command.

For more information on the TRACERTE command, see OS/390 SecureWay Communications Server IP User’s Guide, GC31-8514.

D NET command
To monitor TCP/IP buffer usage, you can use the VTAM command D NET:

- D NET,CSM (see Figure 4-3 on page 78)
- D NET,BFRUSE (see Figure 4-4 on page 79)

To interpret the output of these commands, refer to VTAM Network Implementation Guide, SC31-8370.
### Figure 4-3  Output from D NET,CSM

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>INUSE</th>
<th>FREE</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K ECSA</td>
<td>204K</td>
<td>180K</td>
<td>384K</td>
</tr>
<tr>
<td>16K ECSA</td>
<td>48K</td>
<td>464K</td>
<td>512K</td>
</tr>
<tr>
<td>32K ECSA</td>
<td>320K</td>
<td>192K</td>
<td>512K</td>
</tr>
<tr>
<td>60K ECSA</td>
<td>0M</td>
<td>0M</td>
<td>0M</td>
</tr>
<tr>
<td>180K ECSA</td>
<td>0M</td>
<td>1080K</td>
<td>1080K</td>
</tr>
<tr>
<td>TOTAL ECSA</td>
<td>572K</td>
<td>1916K</td>
<td>2488K</td>
</tr>
<tr>
<td>4K DATA SPACE</td>
<td>288K</td>
<td>160K</td>
<td>448K</td>
</tr>
<tr>
<td>16K DATA SPACE</td>
<td>864K</td>
<td>160K</td>
<td>1M</td>
</tr>
<tr>
<td>32K DATA SPACE</td>
<td>224K</td>
<td>1312K</td>
<td>1536K</td>
</tr>
<tr>
<td>60K DATA SPACE</td>
<td>0M</td>
<td>0M</td>
<td>0M</td>
</tr>
<tr>
<td>180K DATA SPACE</td>
<td>0M</td>
<td>360K</td>
<td>360K</td>
</tr>
<tr>
<td>TOTAL DATA SPACE</td>
<td>1376K</td>
<td>1992K</td>
<td>3368K</td>
</tr>
<tr>
<td>TOTAL ALL SOURCES</td>
<td>1948K</td>
<td>3908K</td>
<td>5856K</td>
</tr>
</tbody>
</table>

**Fixed Current**
- Fixed Maximum = 120M
- Fixed Current = 5168K

**ECSA**
- ECSA Maximum = 100M
- ECSA Current = 3634K

**CSM Data Space 1**
- Name: 00000CSM

---

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>INUSE</th>
<th>FREE</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1018</td>
<td>EXP INCREMENT 51</td>
<td></td>
</tr>
<tr>
<td>TIMES EXP 0</td>
<td>EXP/CONT THRESH 50</td>
<td><em>NA</em></td>
<td></td>
</tr>
<tr>
<td>CURR TOTAL 1002</td>
<td>CURR AVAILABLE 870</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX TOTAL 1002</td>
<td>MAX USED 360</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXP LIMIT 3000</td>
<td>BUFFS REQUESTED 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>EXP INCREMENT 15</td>
<td><em>NA</em></td>
<td></td>
</tr>
<tr>
<td>TIMES EXP 0</td>
<td>EXP/CONT THRESH 15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURR TOTAL 30</td>
<td>CURR AVAILABLE 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX TOTAL 30</td>
<td>MAX USED 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2032</td>
<td>EXP INCREMENT 50</td>
<td><em>NA</em></td>
<td></td>
</tr>
<tr>
<td>TIMES EXP 0</td>
<td>EXP/CONT THRESH 50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURR TOTAL 76</td>
<td>CURR AVAILABLE 73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX TOTAL 76</td>
<td>MAX USED 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>697</td>
<td>EXP INCREMENT 5</td>
<td><em>NA</em></td>
<td></td>
</tr>
<tr>
<td>TIMES EXP 0</td>
<td>EXP/CONT THRESH 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURR TOTAL 5</td>
<td>CURR AVAILABLE 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX TOTAL 5</td>
<td>MAX USED 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>EXP INCREMENT 30</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>TIMES EXP 1</td>
<td>EXP/CONT THRESH 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURR TOTAL 120</td>
<td>CURR AVAILABLE 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX TOTAL 120</td>
<td>MAX USED 97</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning
The NETSTAT command can be used to monitor byte counts and window sizes. It has many parameters; use NETSTAT HELP to display them. Figure 4-5 on page 80 shows an example of the NETSTAT command.

**NETSTAT**

Figure 4-4  Output from D NET.BFRUSE

<table>
<thead>
<tr>
<th>NETSTAT</th>
<th>CMD</th>
<th>BUFF SIZE</th>
<th>EXP INCREMENT</th>
<th>EXP/CONT THRESH</th>
<th>CURRENT</th>
<th>MAXIMUM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IST920I CRPL</td>
<td>144</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>0</td>
<td>18 / NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>475</td>
<td>306</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>475</td>
<td>205</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST920I SF00</td>
<td>112</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>0</td>
<td>15 / NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>352</td>
<td>337</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>352</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST920I SP00</td>
<td>176</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>0</td>
<td>1 / NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>21</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>21</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST920I AP00</td>
<td>56</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>0</td>
<td>3 / NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>56</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>56</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST920I TI00</td>
<td>632</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>145</td>
<td>120 / 300</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>540</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>1140</td>
<td>983</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST920I CRA4</td>
<td>4080</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>0</td>
<td>20 / NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>50</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>50</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST920I CRA8</td>
<td>8176</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST921I</td>
<td>0</td>
<td>2 / NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST922I</td>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST923I</td>
<td>12</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST449I</td>
<td></td>
<td>4544K, MAXIMUM = 4989K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST790I</td>
<td></td>
<td>4989K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST790I</td>
<td></td>
<td>4989K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST449I</td>
<td></td>
<td>115250K, CURRENT = 47K, MAXIMUM = 48K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST449I</td>
<td></td>
<td>47K, MAXIMUM = 48K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST790I</td>
<td></td>
<td>48K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST595I</td>
<td></td>
<td>0K, MAXIMUM = 0K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST811</td>
<td></td>
<td>5320K, MAXIMUM USED = 10057K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST924I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST1565I</td>
<td></td>
<td>1672K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST1565I</td>
<td></td>
<td>36K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IST1565I</td>
<td></td>
<td>5459K</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4-4  Output from D NET.BFRUSE**
Figure 4-5   TCP/IP NETSTAT CONFIG command

TCP/IPSTATISTICS

TCP/IPSTATISTICS provides TCP/IP statistics when TCP/IP is shut down. To get these statistics, specify this parameter on the ASSORTEDPARMS statement in the TCP/IP profile data set. Figure 4-6 on page 81 shows an example of these statistics.
4.2.2 Tuning TCP/IP

This section contains some actions that you can take to tune TCP/IP. However, detailed TCP/IP tuning requires the skill of a specialist. If available, refer to IBM MVS TCP/IP Performance Tuning Tips and Capacity Planning, Share 92, Session 3915, for more detailed tuning information.

TCP/IP priority in z/OS
We recommend that you set the MVS dispatching priority of TCP/IP to be approximately the same as VTAM. Other TCP/IP servers, such as FTP and Telnet, should be set at a slightly lower priority than TCP/IP.

Also, make sure that TCP/IP is getting the z/OS resources that it needs by monitoring it with RMF. If TCP/IP is constrained, it will not perform well. See Chapter 2, “z/OS tuning” on page 23 for more information.

Specify REGION=0M for TCP/IP
TCP/IP buffers are dynamically allocated in the private area. REGION=0M allows TCP/IP to allocate the maximum virtual storage above and below the 16 MB line, and thus does not constrain the number of buffers it can use.

z/OS UNIX System Services network parameters
There are two parameters related to the network in the z/OS UNIX System Services startup parameters in SYS1.PARMLIB(BPXPRMxx):

- NETWORK DOMAINNAME(AF_INET) MAXSOCKETS
  TCP/IP supports AF_INET sockets for communication with OS/390 UNIX applications. Notes clients use AF_INET sockets to communicate with the Domino server. This parameter should be set to 35000. Starting with z/OS V2R7, this parameter applies to each TCP/IP stack instead of across the entire image.
NETWORK DOMAINNAME(AF_UNIX) MAXSOCKETS

AF_UNIX sockets are used when two OS/390 UNIX applications establish a connection. Use a value of 10000 for Domino.

For more information about these parameters, see the product installation guide. For supplemental details, see Lotus Domino for S/390 Release 5: Installation, Customization and Administration, SG24-2083.

There are two parameters related to storage in SYS1.PARMLIB(IEASYSxx):

- CSA (3000,262144)
- SQA (8,448)

There are three parameters related to the Communication Storage Manager (CSM) in SYS1.PARMLIB(IVTPRMxx):

- FIXED MAX (120M)
  Defines the maximum amount of storage dedicated to fixed CSM buffers.
- ECSA MAX (30M)
  Defines the maximum amount of storage dedicated to ECSA CSM buffers.
- POOL
  One POOL definition can be specified for each CSM buffer pool of a particular bufsize and bufsource combination.

For more information about these parameters, see z/OS MVS Initialization and Tuning Reference, SA22-7592. Helpful redbooks include the three-volume series entitled Communications Server for z/OS V1R2 TCP/IP Implementation Guide. The publication numbers are SG24-5227, SG24-5228, and SG24-5229.

There is a Web site for the latest TCP/IP migration hints, tips, and flashes:

http://www.ibm.com/support/techdocs

4.3 Network connection

The zSeries server can connect to the IP network using:

- zSeries Open System Adapter (OSA) family of network adaptors
- 3174 Networking Server
- 3745/3746 Communications Controller
- 3172-3 Interconnect Controller

You should monitor the connection to ensure good performance. Here are some tips.

4.3.1 Monitoring connection activity

You can monitor the connection from a z/OS perspective using RMF postprocessor or MONITOR-II. To use RMF MONITOR-II:

1. Start Mon-II data collection.
2. From TSO, enter RMFMONITOR.
3. Enter DEV COMM in the input area of the primary menu. This creates an I/O activity report for communication devices.
4. Enter D ON in the input area. This includes only the data since the last report, rather than a cumulative total.

5. Press Enter to begin.

6. Look at a couple of sequential intervals and see if there are any constraints (see the DELAY columns).

Figure 4-7 shows a sample report.

![Figure 4-7](image)

In this example, address 102 is an OSA device. The address number matches the DEVICE statement in the TCP/IP profile data set.

For more information on OSA, see *OSA Planning*, GC23-3870.

### 4.4 Network performance

In addition to considering TCP/IP and the network connection on the zSeries server, you also need to consider all of the other network components. That is beyond the scope of this book. However, here we discuss the option of splitting the network.

#### 4.4.1 Splitting the network

Under certain circumstances you may consider splitting the network, in order to improve performance or to separate different types of workloads. For example, you could have a separate network for Domino traffic. This is likely to be more costly than running all traffic over a single network, but it can offer these benefits:

- It allows Domino work to be isolated from other workloads. Therefore, Domino will not be impacted if the other workloads generate a lot of network traffic.
- It allows you to tune the separate networks for their specific workloads. You could, for example, customize the TCP/IP parameters to a specific application pattern usage, such as Domino.

**Note:** Lotus recommends that you run a separate network for inter-server traffic if you use Domino clustering, to ensure good performance for that traffic.

Some of the ways that you could split the network traffic are:

- Add more network interfaces to the host.

  By adding more network interfaces, you can balance the network traffic arriving at the zSeries server. If the interface, such as an OSA card, is shared between several MVS images, you may consider adding additional ones and dedicating them to an MVS image.
Run multiple copies of TCP/IP.
This allows you to tune each TCP/IP for the specific applications that it supports.

Define additional PORT addresses.
Many LAN gateway interfaces support more than one port per gateway, such as OSA-2 and 3172-3.

Use the equal-cost multipath function.
This is provided by OSPF dynamic routing protocol, supported by the OMPROUTE daemon. It allows you to balance the outbound traffic over multiple network interfaces. OSPF protocols in the router network provide for inbound traffic balancing over equal-cost routes.

Use MVS virtual IP addressing (VIPA).
VIPA allows you to define more than one IP address on the same TCP/IP stack. The purpose is to aid in configuring a fault-tolerant network and for security reasons. It is also useful if you want to drive the network traffic through different IP addresses for workload balancing. For inbound traffic, if you have a single TCP/IP stack on a z/OS image and multiple VIPA addresses, you can use Routing Information Protocol (RIP) to split the inbound traffic destined for each of the VIPAs across different network interfaces. Outbound traffic can also be split with RIP, but only by manipulating metrics and filters in the network routers outside of z/OS. Consult Communications Server for z/OS V1R2 TCP/IP Implementation Guide, Vol. 1: Base and TN3270 Configuration, SG24-5227 for more information on this subject.

4.5 Domino network statistics

Domino provides statistics on network use in the Notes log database log.nsf. (see 7.2.2, “Notes log” on page 142). See the views “Usage By Date” or “Usage By User.” This gives you information on the amount of data transferred across the network. It is helpful to analyze and forecast network traffic. You can monitor the network usage throughout the day to determine peak times and bottlenecks.

Using the latest Notes administrator client, a Domino administrator can track and monitor the health of the server environment.

4.6 Domino 6 performance improvements

Domino 6 includes several enhancements that can improve network performance.

4.6.1 Network compression

Network compression is a new performance feature of Notes/Domino 6. When network compression is enabled, data is automatically compressed before it is sent over the network. This improves network performance, especially over slower line speeds.

With network compression enabled on both the sending and receiving computers, the Notes client and/or Domino server will attempt to compress any data (mail being routed or documents being replicated, for example) before sending it out across the network. The compression algorithm used is LZ1.
Several factors may affect compression:

- Data that is first encrypted cannot be compressed.
- Network compression does not further compress an attachment that is already compressed. This includes files which are compressed by Notes when attachment compression is enabled on the database and files which are already compressed (jpeg or zip archive files, for example).
- Both the sender and receiver must be running Notes/Domino 6 and have network compression enabled. Network compression is performed by NRPC calls and applies to both Notes client-to-server and Domino server-to-server sessions. It does not apply, therefore, to HTTP sessions of iNotes clients.

During the beta testing of Notes/Domino 6, Lotus conducted a study of network compression. The results of the study are reported in the LDD Today article “Network compression in Domino 6”, available from the Lotus Developer Domain Web site at: http://www.lotus.com/ldd

They tested the effect of network compression on database replication. The databases had a mix of plain text documents, documents with uncompressed attachments, and documents with compressed attachments. With network compression, the size of the databases transmitted was reduced by 35 to 50 percent, a factor which could have significant benefit on a constrained network.

There is, however, a price to pay for the work the Domino server is doing to compress the data. Another test described in the same LDD article reports the impact on memory and CPU. In their test, memory use increased by 15 %, and there was an increase of 20 % in the Server task. CPU used by the Replica and Update tasks was slightly lower.

One of the early customers of Domino 6 reported that network compression significantly reduced replication time for mobile users on dial-up connections. They also reported that users on high speed lines such as T1 did not observe much difference.

We recommend that network compression be considered for users who replicate their mail over slow connections such as dial-up, or for replication between servers when network bandwidth is limited. Network compression is controlled at the Domino port level. A separate port can be set up for the server-to-server traffic on a high speed backbone (or LPARs within a footprint) that does not use compression, while the end user traffic on a different Domino port can be compressed. This will help to reduce the cost of compression. However, we recommend against a global adoption of this new feature due to the impact on CPU and memory.

4.6.2 LZ1 compaction

LZ1 is the compression algorithm used by network compression in Notes/Domino 6. It can also be used to compress file attachments in a database. LZ1 compaction is more effective than Huffman compaction, the algorithm that was used to compress file attachments in Notes/Domino R5.

By default, databases in Domino 6 still use the Huffman algorithm to ensure backwards compatibility with previous versions of Notes. To use the LZ1 compression you must enable it in the database properties. Results will vary, depending on the type and complexity of the file attachment being compressed, but our testing shows that LZ1 compaction typically reduces the attached file size by a further 5 to 40 percent over Huffman compaction.
4.6.3 Centralized directory

In Domino 6, you now have the option to move from a distributed directory architecture to a central directory architecture. To do this, you store the complete Domino Directory with all of its person and group information on one central server. The central directory information is available to all users, saving disk space because you no longer need to store the whole directory on each server. You also save time, as you no longer replicate your directory across all the servers in your domain.

With a centralized directory, the Server and Router tasks will use that directory for user authentication and mail delivery. A centralized directory will also be used when the user requests name lookup during address resolution.

We recommend that you replicate the central directory to at least one other server for failover.

On a pre-release Domino 6 server, Lotus compared the performance of a local directory with a central directory. They also compared the name lookup rate of Domino 6 with Domino 5.0.10. Preliminary results indicate:
- Using a centralized directory does not degrade server performance.
- The initial load on the network bandwidth (the increased bytes sent/received when accessing a central directory from a mail server) is about 6 KB per mail server. This load will vary depending on the number of users on the mail server.
- Domino 6 can support ten times more name lookups per minute than R5.0.10.

4.6.4 Streaming replication

Streaming replication is a new Domino 6 feature enabled by default on servers and clients. To use streaming replication, both parties involved in the replication must be running Notes/Domino 6. Streaming replication is PULL only.

This feature uses a single streamed RPC to read a number of documents and attachments. This is an improvement over the non-streaming method of requesting and acknowledging one database note at a time.

Streaming replication saves time and consumes less CPU and network bandwidth to replicate the same number of documents. Network latency is reduced, especially on slower links. Documents are also replicated in ascending size order (smallest first). This allows the client to abort a long replication, but still receive some documents. Partial replication is also supported, allowing an aborted replication to be reinitiated and to continue where it was interrupted. Views in folders are also updated incrementally while a replication is in progress (as opposed to at the end of replication). This allows users to begin responding to initial documents (for example, mail messages) while larger documents continue downloading.
Domino tuning

This chapter discusses the Domino parameters you can set that affect the performance of your system. It shows how you specify which tasks the Domino server will run, and provides information on the relative resource use of those tasks. It also identifies useful Domino information that you should monitor.

As in previous chapters, we begin by summarizing key tuning recommendations.
5.1 Recommendations

These are the key recommendations for tuning Domino 6 for z/OS:

- **Do not** change the following parameters in Domino 6 for z/OS:
  - `Server_Max_Concurrent_Trans`
  - `Server_MaxSessions`

  Leave the values at their defaults. Otherwise, you may cause problems with the Domino server on z/OS.

  **Note:** In Domino 6 for z/OS the default value of `Server_Max_Concurrent_Trans` is -1, which is the disabled setting. **Do not** change this default value.

- Do not run unnecessary Domino server tasks.

  Start only those server tasks that you need. Check the `ServerTasks` and `ServerTasksAtHour` variables in `notes.ini`, and remove any addin tasks that are not required. Also check your server configuration document, particularly the NOTES.INI Settings tab, for unnecessary tasks. For example, you may not need tasks such as logging and billing. See the Domino 6 Administrator help database for a more detailed description of the `ServerTask` and `ServerTaskAtHour` ini variables, as well as information about the configuration settings document.

  Note also that in some cases there are advantages to running multiple copies of an addin task, such as Update or the cluster replicator, to increase throughput and shorten various processing times. If you use multiple instances of these addin server tasks, be sure it is necessary to do so.

- Schedule tasks for non-peak times.

  Some tasks run on a scheduled basis. Schedule them to run at non-peak times. Examples are compact, index (to create a new full text index), update, updall, and fixup. The schedule is set by the Notes administrator, but should be agreed on by the z/OS operations staff to make sure that it does not interfere with other peak workloads, administration tasks on the server, or system and Domino backups.

- Balance work between the server and the clients.

  Lotus Notes is a client-server application. It runs on a combination of the Domino server and the Notes client. (If you are using a Web browser as the client, the work that can be done on that client is limited.) Consider running some functions on the client rather than on the server, to minimize the server load. For example, you could tell your users to:
  - Put all users that they send mail to in their local Domino Directory, to avoid searching the Domino Directory on the server when sending mail.
  - Replicate their mail and other databases to their Notes client, to offload processing from the server.

  As you consider these ideas, keep in mind how much server offload you might get. For example, you may not be doing your mail on the server (no load), but you do need to replicate with the server (a load). Also consider the management aspects of users having database replicas on their workstations.

- Educate users on replication.

  If users replicate databases to their Notes clients, educate them on the replication interval. We recommend a value of 15 to 30 minutes or more. As each replication uses server resource, even if there is no data to transfer, a longer interval will reduce the server load.

  Also educate users to use scheduled replication only on databases for which it makes sense. Within IBM we recommend that some applications be run locally by replicating
them to the client. However, for applications used infrequently (for example, to submit expense claims), we recommend that those databases do not replicate automatically. Instead the application reminds the user to replicate the database manually whenever they update information in the local replica.

5.2 Verifying BPX parameters after Domino installation

In between the Domino installation and setup steps, we recommend that you run the `dom_verify_os` command. Figure 5-1 shows a sample of its output. Note that in this case no errors were received, but a warning was displayed recommending that the value of IPCSEMNIDS be increased in the BPX parameter file. If BPX overrides are set in a server’s RACF ID (in their OMVS segment), then `dom_verify_os` may not reflect the actual setup used by the ID running the Domino DPARs. You should validate any RACF overrides against the output from `dom_verify_os`.

```
 dom_verify_os
 ****************************************************
 * Verify z/OS SYS.PARMLIB(BPXPARMxx) specified   *
 * in the Domino 6 install guide.                 *
 ****************************************************
 WARNING: IPCSEMNIDS=6000 Recommended Value >= 20000
 dom_verify_os: Found 0 Severe Errors and 1 Warnings
```

*Figure 5-1 Output from the dom_verify_os command*

Attention: Be aware that this command does not validate any RACF overrides in the OMVS segment of the IDs running your DPARs.

5.3 Domino initialization parameters

The notes.ini file provides the parameters for the initialization of the Domino server. During server startup, this file is read to set up the proper Notes environment and load the predefined tasks and programs.

The notes.ini file contains parameters to define:

- Server tasks to start
- Tuning parameters
- Other parameters

The server configuration document in the Domino Directory allows you to set the same parameters that have always been set in notes.ini. Parameters which have existed prior to Release 5 can be entered on the notes.ini settings page. Newer parameters, such as the number of mailboxes to use, are generally integrated as fields within other pages of the configuration document. Some of these fields will be written through to the notes.ini file when the server is started, so don’t be surprised to see that changes you made in the configuration document show up there.

In the following sections, we discuss the Domino server tasks and some of the performance parameters. For further information on the notes.ini file and the server configuration document, see the Domino 6 Administration Help database.
5.3.1 Changing parameters in notes.ini

To modify parameters in notes.ini you can:

- Use a server configuration document
- Use the Set Configuration server command at the console
- Edit the server document
- Edit the file directly with an editor

These are all described in the Domino 6 Administration Help database and in Lotus Domino for S/390 Release 5: Installation, Customization and Administration, SG24-2083.

5.3.2 Setting notes.ini for performance

There are 2 notes.ini parameters which can affect performance of the Server task in Domino.

**FT_FLY_INDEX_OFF=1**

When an agent is run against a database that is not indexed, and the agent requires the database be full-text indexed, a Domino feature (introduced in R5.0.9) creates a full text index on-the-fly. When the agent completes, the full text index is discarded. Each time the agent runs, a new full text index is created. This can cause performance degradation, usually reported as high CPU usage by the Update task.

This notes.ini parm will prevent these indexes from being created on the fly, and a message should be logged to log.nsf indicating that the agent didn't complete.

**IOCP_Disable Async_Notification=1**

There was a change in R5.0.9 to improve async notification to users. It applies only to users with current connections to Domino. When this is enabled, the Domino server will notify the client when new mail has arrived. Note that this does not affect the client poll to the server, which is set in the client preferences. That is still in place, even if you disable the server notification to the client.

In Domino 6, the default poll time is 60 seconds. The server goes through the list of connected users every 60 seconds to determine if they should be notified of new mail. The longer the list of connected users, the greater the impact on the server. A Domino 6 customer reported that they recently disabled the async notification in Domino 6 (by setting IOCP_DisableAsync_Notification=1) and CPU usage was reduced by 5-10%.

You can either disable async notification or you can set the interval to more than 60 seconds in notes.ini.

5.3.3 The Notes_SHARED_DPOOLSIZE parameter

The NOTES_SHARED_DPOOLSIZE parameter is a UNIX environment variable that you can set before you start the Domino server. This variable tells the Domino server how much UNIX shared memory to ask for in each request to z/OS. The default is 8 MB (8388608 bytes); we recommend that you use this value. In Domino 6, the 8 MB size provides the best balance between the number of segments required and the internal serialization that Domino performs when using these segments.

You set this as a UNIX environment variable by putting the statement

```bash
export Notes_SHARED_DPOOLSIZE = 8388608
```

in the .profile file.
The following notes.ini parameters are related to the number of physical threads per process.

- SERVER_ENABLE_THREADPOOL=1

**Important:** This environment variable is closely related to the z/OS SYS1.PARMLIB(BPXPRMxx) parameter IPCSHMPAGES. *Do not change* Notes_SHARED_DPOOLSIZE without first checking the value of this parameter.

If IPCSHMPAGES is not large enough to accommodate the specified Notes_SHARED_DPOOLSIZE value, the server will not start. The Domino Install Guide recommends that IPCSHMPAGES should be set to the maximum value (25600).


### 5.3.4 Thread parameter

The following notes.ini parameters are related to the number of physical threads per process.

- SERVER_ENABLE_THREADPOOL=1

**Attention:** Do not change this parameter. Non-threadpool servers are not supported in Domino 6.

This parameter controls whether the server will exploit the thread processing model, where a single small pool of physical threads provided by the operating system is used to support a larger set of Domino virtual threads. Using this model, each user who connects to the server gets a Domino virtual thread. This is serviced as required by one of the physical threads in the thread pool.

- SERVER_POOL_TASKS

This defines a total number of physical threads. The default value of this parameter is 100. The number that you define in this parameter corresponds to the SMF108SLTT value of the SMF108 record. SMF108 records provide information about the maximum number of physical threads in use during the interval (SMF108SLPTIUMAX).

Historically, for a mail workload, the default value of 100 threads should support around 2000 active 15 minute users. Periodic spikes in the threadpool usage should be expected due to normal Domino functions like updating names.nsf, and bulk or large mail tying up your mail.boxes. However, spikes in the thread pool are also an early indicator of a bottleneck on the server. Before adding more threads to the threadpool, you must validate that a bottleneck does not exist. Adding more threads will make the bottleneck worse.

For more information on monitoring the number of threads, see 7.1.2, “RMF Lotus Domino server report” on page 137.

### 5.4 NSF buffer pool size

This parameter specifies the maximum size of the NSF buffer pool (in bytes), a section of storage dedicated to buffering I/O transfers between the NIF indexing functions and disk storage.

NSF_BUFFER_POOL_SIZE_MB is an important parameter to support a heavy workload. This parameter specifies the maximum size (in MB) of the NSF buffer pool, a section of memory dedicated to buffering I/O transfers between the NIF indexing functions and disk storage. It can take as much as one-third of the physical memory. You will benefit from a high value for storing views.
5.4.1 Setting the parameter

**Note:** The `PercentSysAvailResource` variable in notes.ini is not supported on z/OS. You must set the NSF buffer pool size explicitly.

You may set up the NSF_Buffer_Pool_Size parameter by editing your notes.ini file. See 5.3.1, “Changing parameters in notes.ini” on page 90 to change parameters in your notes.ini file. You can also specify this setting in the NOTES.INI Settings tab of the Configuration Settings document in the Domino Directory:

```
NSF_Buffer_Pool_Size=<value>
```

**Note:** You can also use NSF_Buffer_Pool_Size_MB to set the maximum size of the NSF buffer pool. This is the same as NSF_Buffer_Pool_Size, except it sets the size in megabytes instead of bytes. The default NSF_BUFFER_POOL_SIZE for Domino for z/OS is 171 MB, since Domino would return the 2 GB address space size. This is different from the other Domino platforms that base the default of the NSF_BUFFER_POOL_SIZE on the amount of real storage available to the system. There is no Domino limit to how big you can set it. However, if it is set above 512 MB, you will tend to run out of virtual storage.

5.4.2 Monitoring

The following steps show the monitoring procedure.

1. Use the `show stat` command to view the NSF buffer pool statistics. See Figure 5-2 on page 92.


3. Check `Database.Database.BufferPool.PerCentReadsInBuffer`. This value should generally be more than 90 percent. If it is less than 90 percent, you should consider increasing your current NSF_Buffer_Pool_Size.

```
> show stat
:
Database.Database.BufferPool.MM.Reads = 0
Database.Database.BufferPool.MM.Writes = 0
Database.Database.BufferPool.Peak.Megabytes = 33
Database.Database.BufferPool.PerCentReadsInBuffer = 97.63
:
```

**Figure 5-2 Using the show stat command to monitor NFS_Buffer_Pool_Size**

The following describes the statistics revealed by the `show stat` command.

**Database.Database.BufferPool.Maximum.Megabytes** - The maximum allowed size of the NSF buffer pool. It defaults to 171 MB. If the notes.ini parameter `NSF_BUFFER_POOL_SIZE=<bytes>` is set, this parameter overrides the default. You may also use the `NSF_BUFFER_POOL_SIZE_MB` variable to set the size of the buffer pool in megabytes. It is simpler than using the `NSF_BUFFER_POOL_SIZE` variable to set the size of the buffer pool.

**Note:** Do not be alarmed if your statistics show that the values for the Peak and Used exceed the Maximum value by a small percentage. This is due to the way the server reads some data structures into the pool.

**Database.Database.BufferPool.Peak.Megabytes** - This value shows the maximum size to which the pool has grown since the server was last started.
**Database.BufferPool.PerCentReadsInBuffer** - The percentage of file read requests made by Domino which were successfully obtained from the BSF buffer pool. The higher this number, the better.

**Database.BufferPool.MM.Reads** - The cumulative number of times, in millions, that the buffer has been read.

**Database.BufferPool.MM.Writes** - The cumulative number of times, in millions, that data has been written to the buffer.

### 5.4.3 Tuning NSF_Buffer_Pool

The parameter NSF_Buffer_Pool_Size must be specified for a machine running a single server partition, or for a machine running more than one DPAR. It should always be specified for partitioned servers, since the physical memory in the machine has to be shared between multiple servers.

Other entries in the notes.ini file may be ignored by the server. Many of these now have an equivalent setting option on a server configuration document and do not need to be listed explicitly in the notes.ini file. They will not impact server performance if left in the file.

Some Domino parameters may greatly influence the performance of a UNIX Domino server. For instance, the NSF_Buffer_Pool value needs to be chosen very carefully.

Setting it to a high value (over 512 MB) will allow Domino to cache its data easily, but may cause z/OS paging overhead if central and expanded storage are in short supply. Setting this variable too low (under 171 MB) may eliminate paging by z/OS at the expense of the Domino caching.

As noted previously, setting the size of the NSF buffer pool to a value greater than 512 MB can cause virtual addressability to be exhausted for a Domino server. When this happens the server will crash, with error messages at the server console indicating that storage cannot be allocated. This can happen regardless of how much real (central or expanded) storage is available on the system.

This is because the size of any virtual address space on z/OS is limited to 2 GB in size. If any application tries to allocate more than this amount of storage in one address space, the allocation request will fail and, in the case of a Domino server, a panic will occur.

Most of the virtual storage that a running Domino server will attempt to allocate goes to the NSF buffer pool, or to other buffer pools whose size is directly determined from the NSF buffer pool size. For this reason, the setting of this ini variable is critical to the operation of the server. The default value of 171 MB is sufficient for moderate production workloads of up to about 2000 users. Values of 512 MB are appropriate for production loads of up to about 5000 registered mail users. These settings have been used in production at IBM for some time.

You can find an article (among many other helpful ones) about NSF_Buffer_Pool entitled “Optimizing Server Performance—Port Encryption and Buffer Pool Settings” at:


The specific URL is:

[http://www.lotus.com/ldd/__85256360005FBOA0.nsf/0/75AEAE86279027FB85256644004485BC3](http://www.lotus.com/ldd/__85256360005FBOA0.nsf/0/75AEAE86279027FB85256644004485BC3)
5.4.4 Recommendation

Our tests indicate that the default NSF_BUFFER_POOL_SIZE value is a good starting point when configuring the Domino 6 server. 512 MB is the maximum size you should consider.

To determine if the buffer pool size is too big, look for paging. Ensure that you have enough central storage available before increasing NSF_BUFFER_POOL_SIZE, or you will cause paging.

Another indicator is the Database.BufferPool.PerCentReadsInBuffer statistic. As you increase NSF_BUFFER_POOL_SIZE, you should see this value increase. If it decreases, you may have grown the buffer beyond the point of benefit.

You should also be aware that other Domino values (particularly the HTTP cache settings) are based on the NSF_BUFFER_POOL_SIZE. By changing this one value, you will have several changes cascading through your DPAR. You should validate all of the server’s settings after changing this value, using the show stat command, to be sure you did not affect another area of the server.

NSF_DBCACHE_MAXENTRIES

The storage used for the dbcache comes out of the NSF buffer pool. Therefore, the higher this value, the less space for other Domino caching activity. If this value is not set, Domino sets it by default to three times the number of NSF_BUFFER_POOL_SIZE megabytes. (For example, an NSF_BUFFER_POOL_SIZE of 171 Megabytes would result in a default setting of 513 for NSF_DBCACHE_MAXENTRIES.)

The key indicator to look at when determining the size of NSF_DBCACHE_MAXENTRIES is the Database.DbCache.HighWaterMark statistic. If your current setting is 1000 entries, then Domino will allow you to have 1400 NSF_DBCACHE_ENTRIES before rejecting any. For large DPARs, you can set your NSF_BUFFER_POOL_SIZE to around 400 MB (1200 DBCACHE entries) and increase the NSF_DBCACHE_MAXENTRIES to satisfy your mail workload. However, as with all changes, you should monitor and validate the impact of your changes on your DPAR.

5.5 Transaction logging

Domino Release 6 supports transaction logging and recovery. With this feature enabled, the system captures database changes and writes them to the transaction log. Then, if a system or media failure occurs, or a user needs to have documents or a whole database restored, you can use the transaction log and a backup utility to recover your database to its most current state, or to any point in time.

All changes to a Domino database are made by transactions applied to the database using a defined set of procedure calls. There is no other way of updating the Domino databases. Domino transactional logging logs both the data and the actual transaction commands used to make the change; for example: “NSFNoteOpen, NoteUpdate, NoteClose.”

Transaction logging provides these main benefits:

- In most situations, you no longer need to run the Fixup task to recover databases following a system failure.
- Faster recovery after crash: When a server terminates abnormally, databases can be left in an inconsistent state. When transaction logging is not used, the server will scan all databases it can find at startup time to see if any are in this inconsistent state.
Consistency checks, a type of fixup, are performed on these databases to remove the inconsistencies. This operation typically requires a great deal of time for large installations that have many databases.

When transaction logging is employed, only those databases that have had updates in the log since the last transaction log checkpoint was written need to be processed at server startup. Since only the small specific number of updates need to be applied to this set of databases, this greatly reduces the processing time. There is no need to scan all of the documents in the database to see if they are all correct.

Transaction logging saves processing time because it allows Domino to defer database updates to disk during periods of high server activity. Transactions are recorded sequentially in the log files, which is much quicker than database updates to random, nonsequential parts of a disk. Because the transactions are already recorded, Domino can safely defer database updates until a period of low server activity.

- **Reduced backup times**: Backing up terabytes of data is a very time consuming process, typically taking many hours to complete. When archived-style transaction logging is used the amount of time needed to do daily backups of databases can be reduced by as much as 80 percent. This savings is realized because only the transactions which have been applied to databases are backed up on a daily basis, rather than the entire contents of the databases. The exact opposite is true for restores. The longer the interval between full backups, the longer the time to restore a database, since more logs must be processed.

  You should consider not only the backup times but also the restore times to determine the right balance of log versus full backups.

  Typically, a base full backup of all databases is taken once a week. The used transaction log extent files are backed up on at least a daily basis. Even on a heavily used server the amount of data in the transactions in the log is only a small percentage of the total size of the data in the database. All transactions against all databases at release 5 or higher are recorded in the transaction log. To restore a database to any point in time all that is needed is the full backup and the transactions executed against the database. Backups are simplified since only a weekly full backup of all databases, and frequent backup of the relatively small used transaction log extent files, are needed to back up the databases, rather than full database backups on a daily basis.

- **More efficient handling of disk I/O**: Transaction logging may improve performance and save I/O processing time because it allows Domino to defer database updates to disk during periods of high server activity. Transactions are committed sequentially to the log files, typically within about 1/100 of a second of the completion of a transaction, rather than being committed to the database on disk. This is much quicker than database updates to random, nonsequential parts of a disk. Since writes to disk are deferred, they can be consolidated, and thus more work can be done with fewer operations. Because the transactions are already recorded, Domino can safely defer database updates until a period of low server activity.

Transaction logging is only valid for databases in the Domino R5 (ODS 41) or Domino 6 (ODS 43) format. Earlier Domino database formats do not support logging; they require backup and recovery to be done at the full database level. In the event of a system failure, Domino automatically performs database recovery against logged ODS 41 and ODS 43 databases during restart. The system uses the transaction logs to undo and apply database transactions not written to disk for databases that were open during the failure.

Additionally, Domino runs its standard fixup utility against databases with transaction logging disabled. In the event that a database becomes corrupted or lost, a backup utility must be invoked to recover the database from a full backup and archived transaction log files. IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino, Version 5, Release 1, Level 5.0 is IBM’s solution for this backup utility. See Chapter 14 of *Lotus Domino for S/390*.
5.5.1 Setting up transaction logging

The following procedure shows you how to set up transaction logging:

1. Ensure that all databases to be logged reside in the Domino data directory, either at the root or in subdirectories.

2. From the Domino Administrator File menu select Open Server → client. In the “Select a server to administer” dialog box select the server for which transaction logging will be enabled.

3. Click the Configuration tab.

4. Click Server, and then click Current Server Document.

5. Click the Transactional Logging tab.

6. Decide which logging style you will use. Choices here are Circular, Linear, or Archival. Circular (the default) is easiest to begin with and includes a slight performance gain. Whichever method you choose, ensure that you have enough disk space.

7. Enable “Transactional logging.” See Figure 5-3 on page 97.

8. Complete other fields as appropriate, and then save the document. See the Domino 6 Administration Help database for more details.

9. Stop and restart the Domino server for the changes to take effect.

Note: The initial setup for transaction logging can result in a long period of unavailability while the server assigns DBIIDs to each database. This could take several hours.
5.5.2 Transaction logging and the Domino 6 lock manager

Domino 6 replaces the semaphore method of locking a database used in Release 5 with a new lock manager. The new lock manager provides for greater granularity by allowing locking at the element level. This allows for greater concurrent access and updating of the database by multiple users. One of the most obvious effects of this is seen when databases are transactionally logged and the number of transactions increased substantially. Under R5, as a server became increasingly busy and the number of transactions increased, the response time would fall off in a direct proportion to the increase in transactions. This was due to an increase in the frequency of the flushing of changes from the UBM to the on-disk copy of the
database. During this time the database was locked, which directly affected response time. With the implementation of object-level locking and improvements in the flushing algorithm, users are able to access the database even during the flushing operation. The net result is improved performance.

To take full advantage of the object-level locking capabilities of the new lock manager, the databases must be in Domino 6 (ODS43) format and transaction logging must be enabled for the database. Domino 6 databases that are not transactionally logged, as well as release 5 format databases, will still use the new lock manager. However, they will not benefit from its full capabilities and will perform at a level similar to Domino release 5.

### 5.5.3 View logging

Domino 6 introduces view logging, a new feature that allows transaction logging of view elements. View logging, when enabled, logs complex view information to the transaction log. When a crash of a server occurs the log and the view can be reloaded from the transaction log, eliminating the need for a view rebuild. There is a slight cost associated with view logging, which causes increased CPU utilization when using this features, so it should only be enabled for complex views. This feature is off by default for most databases; it must be enabled per database view using the Domino Designer client.

**Enabling view logging**

To enable view logging, you use Domino Designer. In Designer, open a view or folder, select the Advanced tab, and check “Logging - Include updates in transaction log.” See Figure 5-4.

![Figure 5-4 Views property box: Set logging to include updates in transaction log](image)

### 5.5.4 Soft deletes

The process of restoring a database, in this case a mail file, on any platform can be time consuming for the administrators. This is especially true when the restore is needed to recover a single document that was recently accidently deleted by the user. To address this problem in Domino release 5, one approach was to turn on soft deletes and create a view that would allow users to recover the deleted document within a range of days. The problem with
this approach was that documents that were soft deleted were still stored in the database and counted toward quota, if mail file quotas were enabled. To reduce the size of a mail file, it needed to be compacted with the `-B` option specified. Due to this, many administrators chose not to enable soft deletes. In that case, the only way to retrieve a deleted document was by restoring the user's mail file to an alternate location, replaying the transaction logs against the database to a point in time when the document existed, and recovering the document from the recovered mail file.

The new design of the mail file standard template for Domino 6 includes a Trash view. Soft deletes are enabled by default. When a user deletes a document it is placed in the Trash view and is recoverable by the user for 48 hours, by default. The issue of quotas is addressed by specifying “Check space used in file when adding a note” in the quota enforcement field on the Transaction Logging tab (Figure 5-3 and Figure 5-6). When this option is specified and a user deletes a document, the space occupied by that message is immediately removed from the calculated size of the mail file. There is no need to run the Compact task to recover space. Users who cannot receive mail because of a quota violation can reduce the current size of the mail file immediately by archiving or deleting messages. Since the calculation of quota does not include the items in the trash (soft deletes), it is not necessary to compact the database until reaching the normal weekly database maintenance window. Users that need to undelete a document can do so without the necessity (and overhead) of having an administrator restore their mail file.

### Changing the retention period for soft deletes

The retention period of soft-deleted documents can be set by changing the value in the “Soft delete expire time in hours” value on the database properties tab shown in Figure 5-5.

#### 5.5.5 Enabling quota enforcement

1. From the Domino Administrator, click the Configuration tab, expand the Server section, and click All Server Documents.
2. Select the Server document to edit, and then click Edit Server.
3. Click the Transactional Logging tab, and in the Quota enforcement field, select one of these methods. Then click Save and Close.
5.5.6 Restoring databases

Under Domino release 5, there were two methods used to restore databases when transaction logs needed to be applied. The database could either be restored to the server from which it came, usually into an alternate directory or with a different name, or the database could be restored to an alternate server specifically configured for the purpose of restoring databases.

The option of restoring a file to the server from which it came was the easiest method to implement. However, it was the method that had the most effect on performance, both for the server and for the restore process. No additional work needed to done to set up this option. The process of restoring a database to the original server simply required that the TDP restore command be run, followed by the activatedbs command. If needed, the archived transaction log extent files are restored to the active transaction log directory and replayed against the database.

There are two problems with this method. First, the transaction logging process performs best when the head is moving sequentially across the disk surface and not moving unnecessarily. Since the archived extent files must be written to the active directory, the head needs to be repositioned to an empty portion of the disk and the file written there. The log file must then be replayed to find all transactions that need to be applied to the database. In a restore it may be necessary to restore and replay a large number of extent files. The second problem is that the transaction logging process will give absolute preference to writing new transactions to disk. The process of replaying the restored logs needs to wait for periods of lower activity to do its work. This results in extremely long restore times.

The alternate method employed to restore databases is to set up a server, or rather a server shell structure, for restores. This is never started as a Domino server; instead, it has directory tree structures mirroring those on the production server. It must be at the same level as the source server and must have a copy of the server.id and notes.ini files for source server or servers. Additionally, when a restore is requested, the most current transaction log extent files should be archived and restored to the restore server’s transaction log directory.

There is a twelve-step process, described in the Tivoli Data Protect for Domino manual, that needs to be followed to set up the restore server for restoring a database. Most administrators use a series of scripts to set up the alternate server restore environment and process the restore. While the task of setting up this environment is a bit difficult, it offers the benefit of faster restores with no effect on the production Domino servers. Since the archived extent files are restored to the transaction log directory of the restore server for replay, there is no effect on the production environment. This is the recommended method for restoring a database under Domino release 5 and is still a viable option in Domino 6. Instructions for restoring to an alternate server or partition are contained in IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX and OS/400® Installation and User’s Guide, SC32-9056 in Appendix B, “Advance restore procedures.”

Domino 6 introduces a new notes.ini parameter: TRANSLOG_RECOVER_PATH. It allows the path through which the archived transaction log extent files are to be restored to be specified on the server. Since the restored extents no longer need to be restored to the active transaction log directory, the effect on performance is minimal. Effectively this allows the
database to be restored to an alternate directory or filename to the server from which the
database was backed up. It has the same results as if it had been restored to an alternate
restore server using the alternate server method commonly used under Domino R5. This will
result in much shorter restore times and negate the need for the additional steps to make the
database available to the Notes user. This method of restore should be seriously considered
for use when restoring databases under Domino 6. The use of this method requires the use of
IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino, Version 5, Release 1,
Level 5.0, which supports this and other new enhancements to Domino 6.

To use the alternate path for restoring transaction logging extent files, add the following line to
the notes.ini on a Domino 6 server:

TRANSLOG_RECOVER_PATH=<path to restore extents to>

5.5.7 Test results of transaction logging

Table 5-1 compares recovery performance of a test system running with and without
transaction logging. Our test environment was:

- 9672-ZZ7 with three dedicated CPs
- OS/390 Release 2.6
- DFSMS 1.5 with DFSMS virtual storage 250 MB, fixed storage 250 MB
- Lotus Domino for S/390 Release 5.01
- Number of partitions: 1
- NSF_Buffer_Pool_Size_MB = 320
- Number of mail boxes: 8
- Number of connected users: 5000
- Application: Mail with calendar and scheduling

The test results are shown in Table 5-1. A large performance benefit can be obtained from the
64-bit real memory addressing capability of z/OS, and both Domino R5 and Domino 6 can run
in this mode (on z/OS versions that support it).

Table 5-1 Test results of transaction logging

<table>
<thead>
<tr>
<th>Case</th>
<th>Recovery time</th>
<th>Response time</th>
<th>CPU utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable</td>
<td>01:40:40 (100%)</td>
<td>233ms (100%)</td>
<td>48.9% (100%)</td>
</tr>
<tr>
<td>Standard</td>
<td>00:17:43 (17%)</td>
<td>216ms (93%)</td>
<td>46.3% (95%)</td>
</tr>
<tr>
<td>Favor Runtime</td>
<td>00:33:24 (33%)</td>
<td>172ms (74%)</td>
<td>41.0% (84%)</td>
</tr>
<tr>
<td>Favor Recovery</td>
<td>00:12:35 (12%)</td>
<td>223ms (96%)</td>
<td>46.7% (96%)</td>
</tr>
</tbody>
</table>

5.5.8 Recommendation

We recommend that you use transaction logging. There is a CPU cost of approximately 10 %
associated with transaction logging. However, the benefits far outweigh this cost; transaction
logging provides significantly better server restart time if the server crashes, improved data
integrity, and better scaling of end-user response times.

You must, however, be very certain that you follow the instructions for putting the transaction
log on an appropriately isolated file system or single volume. The transaction log can
potentially receive an extremely large volume of I/O, and it will become a bottleneck if not
configured properly.
## 5.6 Domino server tasks

Domino consists of a number of server tasks, which are identified in Table 5-2.

<table>
<thead>
<tr>
<th>Task name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminP</td>
<td>Administration process</td>
</tr>
<tr>
<td>AMgr</td>
<td>Agent manager</td>
</tr>
<tr>
<td>Billing</td>
<td>Billing</td>
</tr>
<tr>
<td>Calconn</td>
<td>Calendar connector</td>
</tr>
<tr>
<td>Catalog</td>
<td>Cataloger</td>
</tr>
<tr>
<td>Chronos</td>
<td>Updates full-text indexes</td>
</tr>
<tr>
<td>Cladmin</td>
<td>Cluster administration process</td>
</tr>
<tr>
<td>Clbdbdir</td>
<td>Cluster database directory manager</td>
</tr>
<tr>
<td>Clrepl</td>
<td>Cluster replicator</td>
</tr>
<tr>
<td>Collect</td>
<td>Statistic collector collects statistics for multiple servers</td>
</tr>
<tr>
<td>Compact</td>
<td>Database compactor</td>
</tr>
<tr>
<td>Design</td>
<td>Designer</td>
</tr>
<tr>
<td>DIIOP</td>
<td>Allows Domino and the browser client to use the Domino Object Request Broker (ORB) server program</td>
</tr>
<tr>
<td>Dircat</td>
<td>Directory cataloger</td>
</tr>
<tr>
<td>Event</td>
<td>Event monitor</td>
</tr>
<tr>
<td>Fixup</td>
<td>Database fixup</td>
</tr>
<tr>
<td>HTTP</td>
<td>HTTP server</td>
</tr>
<tr>
<td>IMAP</td>
<td>IMAP server</td>
</tr>
<tr>
<td>ISpy</td>
<td>Sends server and mail probes and stores the statistics</td>
</tr>
<tr>
<td>LDAP</td>
<td>LDAP server</td>
</tr>
<tr>
<td>MTC</td>
<td>Reads log files produced by the router and writes summary data about message traffic to a database for message tracking purpose</td>
</tr>
<tr>
<td>NNTP</td>
<td>NNTP server</td>
</tr>
<tr>
<td>Object</td>
<td>Object store manager</td>
</tr>
<tr>
<td>POP3</td>
<td>POP3 server</td>
</tr>
<tr>
<td>Replica</td>
<td>Replicator</td>
</tr>
<tr>
<td>Router</td>
<td>Route mail to other servers</td>
</tr>
<tr>
<td>Sched</td>
<td>Schedule manager</td>
</tr>
<tr>
<td>Server</td>
<td>Used for the Domino user address space</td>
</tr>
<tr>
<td>SH</td>
<td>Used for server base address space</td>
</tr>
</tbody>
</table>
Chapter 5. Domino tuning

103

Notes for table 5-2:

1. Some of these tasks are required, while others are optional. Some, such as Fixup, Updall, Compact, and Catalog, can be scheduled to run at non-peak times.

2. Look at the tasks that your server is running and see if there are some that you do not need or that you can schedule at non-peak times. Stats, Update, AdminP, and Billing are some possible candidates that you may not need to run at peak times. The Event and LDAP tasks start automatically.

For a detailed description of the various server tasks, see the Domino 6 Administration Help database.

### 5.6.1 Domino tasks started

When the Domino server starts, it will start a number of tasks, as specified in notes.ini. If you issue the server command `show tasks` from the Domino server console, you will see the output shown in Figure 5-7 on page 104.

Each task listed in this example is a thread in the main server address space. For more details, see 2.2.1, “z/OS address spaces” on page 25.

<table>
<thead>
<tr>
<th>Task name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statlog</td>
<td>Statistics: Records database activity in the log file</td>
</tr>
<tr>
<td>Update</td>
<td>Updates changed views and/or full-text indexes; a permanent task</td>
</tr>
<tr>
<td>Updall</td>
<td>Updates all changed views and/or full-text indexes for all databases; a temporary task</td>
</tr>
<tr>
<td>Web</td>
<td>Web retriever</td>
</tr>
</tbody>
</table>
Every task that is marked “database server” is part of the server process.

5.7 Replication

Replication is one of Domino's key functions. Replication ensures that all users on a global network can access the latest version of a database without having access to one centrally located database. By accessing a local copy, users can get better response times.

During replication, copies of a database on different systems exchange modifications in one or both directions. There are two types of replication:

- Replication between servers
- Replication between a client and a server

5.7.1 Replication between servers

Replication between servers is managed by the Notes administrator. An administrator can set up the system so that databases will be automatically replicated at intervals.

In a large Domino network, replication is a important factor in planning network traffic and system load. You should carefully plan the times, frequency, and type of database replication. To optimize this area, the Notes administrator should work closely with the z/OS system programmer and the network specialist.
For more information about settings for server-to-server replication, see the Domino 6 Administration Help database.

**Number of replicator tasks**

On a processor with a single engine, run a single replicator task. The task will handle replication requests one at a time.

On a processor with multiple engines, we recommend that you start with a single replicator task. If you have a replication backlog you can start more replicator tasks, but the maximum number should be one less than the number of engines. This will ensure that one engine is always available for time-critical work.

For more information, see the Domino 6 Administration Help database.

### 5.7.2 Replication between a client and a server

Databases can also be replicated between a client and a server. Creating a local copy of a database on your workstation will significantly increase performance and availability. It will reduce the amount of time it takes to open and access the database (such as your mail), as well as let you continue to process when the network or server is temporarily unavailable.

Client replication is managed from the Notes client. The Notes administrator cannot control this type of replication, other than by restricting databases from being replicated and by providing guidelines to the users.

To tune client-to-server replication, consider the following:

- **Large databases**
  A single replication can take a large amount of CPU time and do many I/Os to the file system, depending on the size of the database.

- **Resources used**
  Each replication uses resources even when there are no changes to the databases. Since a client-to-server replication cannot depend on an existing connection to the server, each client replication must open a new session to the server, authenticated to the server, and close the session upon termination of the replication.

**Replication frequency**

Users could use a significant amount of resources replicating databases between the clients and the server even when there are no changes to the data. Therefore, you should encourage your users to not do unnecessary replication. The replication frequency is specified in the `repeat every` keyword in the Notes client's location document. This value should not be set below 15 minutes. Thirty minutes or higher is recommended.

**Monitoring replication activity**

There is no administration tool to limit the frequency of replicating databases to a client (the minimum value is one minute). However, the Notes administrator can see the replication activity in the `log.nsf` database. You can request that client session events are logged at server setup time, or with the `notes.ini` parameter `Log_Sessions`, or with the `Log_Replication` parameter.

We recommend using the `Log_Replication` parameter and setting it to 1. This will reduce the amount of message traffic at the server console (compared to `Log_Sessions`) since there won't be messages logged every time a user opens a database on the server for non-replication purposes. `Log_Replication` will show more details about replication, depending on the value you select.
You will see messages in the Notes log for each client-to-server replication. Figure 5-8 shows that documents for the user Ingo ik Karge are checked for replication every minute.

Thus you can see how often your users have set up replication to run on their Notes clients.

Another way to monitor replications and sessions is with the Domino billing process. Among the several billing classes are replication and session. This billing process does use CPU cycles to run (the more classes that are active, the more cycles), so it is best to take periodic snapshots of your DPAR activity.

Database access on server versus local

The decision on whether it is more efficient to access a database on the server or replicate it to the client and access it locally depends on many factors, such as:

- Will the user access all of the data in the database, or just a subset of it?
- How often does the user need to replicate to have up-to-date information?
- Will all users access the server at the same time, resulting in a large workload peak on the server which could be eliminated if replication is used?

5.8 Indexing

Indexing server tasks are used to update and repair the views and the index of a database after the database has been changed. Update and Updall are the two tasks responsible for this process.

Only a user with designer or manager access to the database can create a new index for a database. For example, you can create an index on your own mail database because you have manager access level. For the common databases, especially large databases, the Notes administrator limits the number of people who can create new indexes and full text indexes. This is important because indexing can use a significant amount of system resource.

Chronos is another server task that Domino runs hourly to update the index.

5.8.1 Update and Updall tasks

Update is a standard server task, and can be started from notes.ini. This task is usually automatically started when the server is started. Update runs continually, checking its work queue and folders that require updating. When a view or folder change is recorded in the queue, Update waits approximately 15 minutes before updating all view indexes in the database so that the update can include other database changes made during the 15-minute period.
To improve view-indexing performance, you can run more than one update task. However, each update task will use additional resources, so plan this carefully.

Updall updates all views that have been accessed once and all full text indexes for all databases on the server. It runs as a temporary server task. After finishing its work, the task shuts down. Updall can also be started from notes.ini. Check whether you need this function in your environment.

**Note:** The Updall server task has a number of parameters or arguments. We recommend that you run Updall without any arguments specified. In particular, beware of the `-R` argument. It rebuilds from scratch all database views that have been accessed at least once, and updates all full text indexes. This can be time-consuming and affect server performance, so use it only as a last resort to solve corruption problems.

Both Update and Updall can be scheduled using the `ServerTasksATx=Taskname` parameter in notes.ini.

When you create an index for a database, you can select an update frequency from the list for the option “update frequency”. The options you can select are shown in Table 5-3. You can change this option after creating the index. The default is “Immediate,” but this default option consumes large amounts of CPU in the update process. We don’t recommend using the default option. A less frequent schedule is recommended. For more information, see the Domino 6 Administration Help database.

<table>
<thead>
<tr>
<th>Frequency option</th>
<th>Updates occur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>Nightly, when the Updall server program runs by default at 2 AM</td>
</tr>
<tr>
<td>Hourly</td>
<td>Every hour, as scheduled by the Chronos server task</td>
</tr>
<tr>
<td>Immediate</td>
<td>As soon as possible after you close the database</td>
</tr>
<tr>
<td>Scheduled</td>
<td>As scheduled by a Program document for the Updall server task in the Domino Directory</td>
</tr>
</tbody>
</table>

### 5.8.2 Full text index

A **full text index** is a collection of files that lets users search database information. Only a user with designer or manager rights can create a full text index.

The creation of a full text index takes system resources:

- CPU cycles are used to create the index.
- I/Os are generated to build the index.
- DASD space is used to hold the index.

Depending on the size of the database, and the amount of text, indexing can take a significant amount of resource. Therefore we recommend:

- Running indexing at times of low system load.
- Limiting which users can do indexing.

Overall, expect up to a 10 percent overhead in CPU consumption required to support full text index processing once the index has been created. Memory consumption may grow by a similar amount, and DASD activity will also climb slightly, depending on how much the users of the system make use of the indexes.
5.8.3 Number of indexing tasks

On a processor with a single engine, run a single indexing task. The task will handle indexing requests one at a time.

On a processor with multiple engines, we recommend that you start with a single indexing task. If you have an indexing backlog you can start more indexing tasks, but the maximum number should be one less than the number of engines. This will ensure that one engine is always available for other, more time-critical work.

5.8.4 Indexing parameters

We recommend the following parameter settings for indexing:

► Update_Suppression_Time
  
  This parameter specifies the delay time between full text index and view updates. Increase the value to index less often.

► Update_Suppression_Limit
  
  This parameter overrides Update_Suppression_Time and forces an update of the views when a specified number of updates have been received.

5.9 Compact

When documents and attachments are deleted from a Notes database, there are blocks of unused space left within the database. Compact is the Domino task for getting back this unused space. Databases should be compacted periodically to reclaim this space. Statistics reports can be used to monitor databases sizes.

There are several ways to run the compact task. For example, you can start it with the load compact command from the Domino console. Compact is not automatically started with the default notes.ini. It runs as a temporary task.

There are three styles of compacting used in Release 5:

► In-place compacting with space recovery
► In-place compacting with space recovery and reduction in file size
► Copy style compacting

Table 5-4 compares the three styles of compacting.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>In place, space recovery</th>
<th>In place, space recovery, with file size reduction</th>
<th>Copy style</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases that use it when compact runs without options</td>
<td>R5 logged databases with no pending structural changes</td>
<td>R5 unlogged databases with no pending structural changes</td>
<td>R4 databases; R5 databases with pending structural changes</td>
</tr>
<tr>
<td>Databases you can use it on</td>
<td>R5</td>
<td>R5 (Need -8 option to enable size reduction)</td>
<td>All (Need -c for R5 databases)</td>
</tr>
<tr>
<td>Relative speed</td>
<td>Fastest</td>
<td>Medium</td>
<td>Slowest</td>
</tr>
<tr>
<td>Users can read databases during compacting</td>
<td>Yes</td>
<td>Yes</td>
<td>No (Unless -L option used)</td>
</tr>
</tbody>
</table>
Compaction can use a lot of processor time and create a large amount of I/O activity, depending on the amount of data being compacted. Therefore, you should run it at a quiet period on the system.

We recommend that you run compact at least monthly to recover unused space, and more often if there are a lot of deletes in your databases. You should also run compact for a specific database after a lot of delete activity.

Once you install Domino 6 on a server, you can upgrade the format of databases on that server. Compacting a database using Domino 6 upgrades the database format (on-disk structure, or ODS). When migrating a name and address book database from Domino Release 5, it is extremely important to compact this database. For more information on migration of a Domino server, see the Domino 6 Administration Help database.

Notes for Table 5-4:

1. Compacting a database using copy style compacting creates a temporary copy of it in the same directory as the original. Make sure there is enough disk space available for the temporary copy in the dataset, especially if the database is large.

2. By default, compacting converts Release 5 file systems to the Release 6 file format (ODS). To prevent this, use the -r option with the load compact command. However, it is recommended that you convert Release 5 file format databases to the Release 6 format because of its many advantages.

3. Compacting a database on a server with transaction logging enabled causes the DBIID to change. This database then requires a full backup. This is handled for you if you are using the Tivoli Data Protection tool.

See the Domino 6 Administration Help database for more information on compact.

## 5.10 Unread marks

One performance enhancement made to the R5 server was to give administrators and users the ability to disable unread marks processing for specific databases. Unread marks may be disabled by a selection from the advanced features page of the database properties. When disabled, the server stops maintaining a list of documents that each user has not yet read. For example, a user may not want to know which documents they have read in a particular database. Also, a user who accesses a database with a Web browser through HTTP may not require unread marks.

It is important to keep in mind, though, that unread marks are a useful feature of Domino, and you may not want to disable them. Databases where it is probably desirable to maintain unread marks include the mail databases of your users and discussion databases where users want to know which documents are new.
5.11 Multiple MAIL.BOX databases

When a Domino server is used as a Notes mail server, the router process is used to route mail between users. This process makes use of a special database called MAIL.BOX, which serves as a temporary location for notes that are in transit. Prior to Release 5, the router could make use of only one MAIL.BOX database, and this would become a bottleneck when the server got to be heavily loaded.

For Release 5 and later, it is possible to create several MAIL.BOX databases, and to have the router use multiple execution threads so that mail routing can take place in parallel. This significantly improves the scale and performance characteristics of the server.

Normally, disk contention is not an issue for MAIL.BOX, so usually you would not need to put the multiple MAIL.BOX databases on different disks. However, GRS latch contention could be an issue with multiple MAIL.BOX databases on the same DFSMS/HFS. If you are seeing GRS latch contention on your MAIL.BOX HFS, then you will need to split them.

You can specify the number of MAIL.BOX databases in the configuration settings document. The maximum number of MAIL.BOX databases is 10. To create multiple MAIL.BOX databases:

1. Make sure you already have a Configuration Settings document for the server to be configured.
2. From the Domino Administrator client, click the Configuration tab and expand the Messaging section.
3. Click Configurations.
4. Select the Configuration Settings document for the mail server or servers you want to restrict mail on and click Edit Configuration.
5. Click the Router/SMTP - Basics tab.
6. Specify the desired value in the “number of mailboxes” field. The default is 1 and the maximum is 10.
7. Save the document.

You can confirm how many MAIL.BOX databases you currently use on the server from `sh tasks` command output. You can see sample output from this command in Figure 5-7 on page 104.

We think that a single MAIL.BOX is sufficient for a small installation; for example, fewer than 1,000 registered users. Above this number, we recommend adding one or more MAIL.BOX databases. As a very general rule, if pending mail starts to accumulate, one possible cause of the problem could be contention on MAIL.BOX databases.

From our test experiences, more than one MAIL.BOX was a big help, but adding more than two had a much less positive impact on router performance. However, the optimal number of MAIL.BOX databases varies depending on router usage in your system.

For more information, see the Domino 6 Administration Help database.

5.12 Web server

Domino has a built-in Web server, which we discuss in this section.
5.12.1 Domino Web server

Most functions available to a Notes client are available to a Web browser, although some, such as database replication, are not supported due to technology differences between the two user interfaces. The Domino server provides Web server function, and also provides a conversion between the Web HTTP and HTML protocol standards and the Notes internal standards. This is shown in Figure 5-9.

![Figure 5-9  How the Domino Web server works](image)

The Web server task is called http. You can start it by adding it to the ServerTasks parameter in notes.ini, or by issuing the Domino server command `load http`. z/OS creates an address space for the http task.

5.12.2 Domino Web server parameters

The http task reads the parameters in the HTTP server section of the server document in the Domino Directory. There you can specify the minimum and maximum number of threads that you want the http task to run. When a request is received from a Web browser, it is run on one of these threads. When the response is sent to the browser, the thread is released and is available for the next request. There is no long-running user connection. See the Domino Administrator 6 Help database for more discussion on this.

If there are no available threads when a request is received, the request is queued until a thread is released. Therefore, you should ensure that you have the maximum number of threads set sufficiently high. You can see the number of threads used by issuing the command `show stat domino` on the Domino server console.

You should set the minimum number of threads high enough to handle a typical workload. This reduces the need to create threads, and thus minimizes the system load.

For more information on the Web server task and related tuning parameters, see the Domino 6 Administration Help database.
Domino configuration advanced options

In this chapter, we cover the following advanced features of a Domino server configuration:

- Partitioning
- Clustering
- Hubs
6.1 Partitioning

Prior to Domino partitioning, each Domino server required its own copy of the operating system, which for some platforms also implied its own processor. With the ever-increasing size of processors on all platforms, this was too restrictive.

zSeries has logical partitioning (LPAR), which allows up to 15 images, each running its own copy of the operating system. Even this is restrictive for a large user population.

Domino partitioning (DPAR) permits multiple Domino servers to run in one copy of the operating system.

Thus, on zSeries, you can have several Domino servers in a z/OS image and several LPARs each running z/OS images. This enables an S/390 or zSeries processor to support a very large number of users on a single processor.

With Domino R5 for S/390, all the DPARs in an image had to be at the same release level. Domino 6 for z/OS allows DPARs in one image to be at several levels of Domino 6 for z/OS and one level of Domino R5 for S/390.

We now look at the benefits and costs of Domino partitioning. Then we discuss the current recommendations for configuring DPARs and LPARs to support a large user population.

6.1.1 Benefits of partitioning

Running multiple Domino partitions in an LPAR and multiple LPARs in a processor has many benefits over running a large number of single Domino server processors.

Domino partitions and LPAR exploit the strengths of the zSeries processor. zSeries has a history of efficiently managing multiple workloads on a single processor. Each DPAR should be considered as a distinct workload.

The benefits of partitioning include:

- It is an efficient way to support a large number of users.
- A peak within one DPAR can be satisfied by unused capacity of the other DPARs. With single Domino server processors, each processor has to be configured for its own peak demands. If using DPARs and LPARs on a single processor, you need to configure for the concurrent peak.
- You can separate sets of users from each other by placing them in different DPARs. This is usually for a business reason, such as security, rather than for any technical reason.
- Priority can be given to selected DPARs. Within an LPAR, Workload Manager can have different priorities set for the DPARs. Across LPARs, the weights can be set to favor one LPAR over the others. Examples where this may be used are:
  - Production DPARs over test DPARs
  - Mail DPARs over application DPARs (or vice versa)
  - Giving priority to a mail DPAR with important users over other mail DPARs
- One or more DPARs can have their CPU use capped. This is not recommended for normal production DPARs, but you may have a requirement to cap test DPARs when running alongside production DPARs, particularly in the same LPAR. Capping can be done within an LPAR using Workload Manager resource groups. Or, an LPAR can be capped at its weight, so that it will not exceed its share even when there is spare capacity.
As the number of users grows or more applications are added, your processor can be upgraded to accommodate them. This is more efficient than adding new extra processors. By adding partitions in an existing LPAR, you can reduce your server-to-server network traffic. By using VIPA and networking, traffic will stay in the TCP/IP stack in your LPAR if both servers are in the same LPAR. Also, an OSA Express adapter will allow traffic within the LPAR (or even to other LPARs that have a connection to the same OSA Express adapter) to bypass your network.

The infrastructure necessary to support a large number of users is much simpler on a single processor.

### 6.1.2 Costs of partitioning

Running partitioned servers uses increased resources over a single server. This is why we recommend that you fill your servers to near the recommended maximum number of users. However if you choose to implement extra DPARs, there will be extra costs, such as:

- DASD space to hold the server files and possibly multiple copies of certain databases. Each server must have its own set of files and databases. There should be no sharing of production data between servers.
- CPU to run the extra server; it uses some CPU cycles even when no users are active.
- Processor storage to run the server. There is a base storage requirement for each server.
- CPU resources for communication between servers. This includes mail routing and database replication. In a single server, these tasks are not needed.

The following sections describe our test results from multiple Domino partitioned servers on a single OS/390 server. Tests with Domino 6 on z/OS yielded similar results. Using 64-bit mode increases performance in both environments. For either release, 64-bit mode allows for the installation of the larger amounts of central storage required to scale up the number of Domino partitions on a given LPAR.

### 6.1.3 Test results with multiple partitioned Domino servers

With the advent of 64-bit real storage available on zSeries, we ran a series of three tests to verify our previous results with multiple partitions on an LPAR and demonstrate that larger numbers of Domino partitions were practical on a single LPAR. To accomplish this, we chose to simulate a fixed number of users spread over different numbers of Domino partitions. All other features of the test environment were kept fixed.

**Test environment**

- zSeries eServer™ Z900 model 2064-116, all engines dedicated
- Storage configured: central storage 32 GB
- z/OS V1R1
- DFSMS 1.6 with default DFSMS buffer settings
- Lotus Domino for S/390 Release 5.07
- NSF_Buffer_Pool_Size_MB = 896
- Application

For this series of tests, we used an internal tool with workloads similar to Notesbench Webmail workloads. We chose this over benchmarks that simulate regular Notes client access because it stresses the CPU and storage characteristics of the server much more heavily.
We supported a workload of 8,400 Webmail users with four, eight, and twelve Domino partitions on a single LPAR. The results are shown in Table 6-1 and Figure 6-1.

### Table 6-1  Test results of multiple Domino partitioned servers

<table>
<thead>
<tr>
<th>Number of DPARs</th>
<th>NotesMark</th>
<th>Response time (msec)</th>
<th>CPU utilization, %</th>
<th>Central storage (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 dpars</td>
<td>4265.00</td>
<td>356.00</td>
<td>61.22</td>
<td>6.40</td>
</tr>
<tr>
<td>8 dpars</td>
<td>4240.00</td>
<td>77.00</td>
<td>59.52</td>
<td>11.09</td>
</tr>
<tr>
<td>12 dpars</td>
<td>4282.00</td>
<td>79.00</td>
<td>60.84</td>
<td>15.12</td>
</tr>
</tbody>
</table>

This particular workload stressed each individual Domino partition pretty heavily in the four-DPAR configuration, as can be seen by the significantly higher response time to the end user. Although response time was higher for this configuration, it was certainly reasonable, and the lower resource requirements make this the most efficient of the three configurations.

CPU usage remained flat for this benchmark across all of the runs; however, we don't expect this to be the case in a production environment. Benchmark tests do not include many common production features, such as virus checking and agent manager. Expect to see incrementally higher CPU requirements for production environments that use more DPARs.

The real overhead from running more Domino partitions to support a given workload comes with the central storage requirements. You should plan for substantially more central storage as your number of Domino partitions grows.

### 6.1.4 Test result with four Domino partitioned servers

Following are our test results from running four Domino partitioned servers on a single OS/390 server. The results would be similar on z/OS.
Test environment

- 9672-ZZ7 with 12 dedicated CPs
- Storage configured: central storage 1887 MB, expanded storage 14 GB
- OS/390 Release 2.6
- DFSMS 1.5 with DFSMS virtual/fixed storage 350 MB (data21, data22), 400 MB (data23, data24)
- Lotus Domino for S/390 Release 5.01
- NSF_Buffer_Pool_Size_MB = 160
- Application: Mail with calendar and scheduling
- Data21: 1 Domino partitioned server (DPAR), 8000 connected users
  - Data22: 2 Domino partitioned servers (DPARs), 16000 connected users. Each server has 8000 users.
  - Data23: 3 Domino partitioned servers (DPARs), 24000 connected users. Each server has 8000 users.
  - Data24: 4 Domino partitioned servers (DPARs), 32000 connected users. Each server has 8000 users.

Results

The test results are shown in Table 6-2 and Figure 6-2. The CPU cost for three or four DPARs is relatively high.

Table 6-2  Test results of four Domino partitioned servers

<table>
<thead>
<tr>
<th></th>
<th>CPU utilization (%)</th>
<th>Shared storage used (MB)</th>
<th>Central storage used (MB)</th>
<th>Expanded storage used (MB)</th>
<th>Total storage used (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DPAR</td>
<td>10</td>
<td>871</td>
<td>1736</td>
<td>33</td>
<td>1769</td>
</tr>
<tr>
<td>2 DPARs</td>
<td>21</td>
<td>1732</td>
<td>1886</td>
<td>960</td>
<td>2846</td>
</tr>
<tr>
<td>3 DPARs</td>
<td>38</td>
<td>2660</td>
<td>1887</td>
<td>2226</td>
<td>4113</td>
</tr>
<tr>
<td>4 DPARs</td>
<td>61</td>
<td>3520</td>
<td>1887</td>
<td>3266</td>
<td>5153</td>
</tr>
</tbody>
</table>

Figure 6-2  CPU utilization and total storage used for four partitioned servers
6.1.5 Partitioning recommendations

Partitioning is used for two primary reasons:

1. To support a large number of users.
2. To separate a set of users from another set of users for business reasons, such as security. The two sets of users could be in the same DPAR with different organizational units. However, some businesses feel the need to keep them in separate DPARs.

The most efficient Domino configuration, in terms of CPU and real processor storage requirements in a z/OS environment, is to put as many users as possible in a DPAR and as many DPARs as possible in an LPAR.

There is a CPU and real storage cost to run an LPAR, irrespective of the number of DPARs running in the LPAR. There is also a CPU and real storage cost to run a DPAR, even when there are no users currently active on the DPAR. Thus having more DPARs than is necessary and more LPARs than is necessary to support the required number of users incurs extra CPU and real storage cost. If you have a requirement to have extra DPARs for user separation, you will need extra CPU and real storage.

The remaining discussion on partitioning recommendations assumes that you will fill DPARs to the maximum number of users. It covers only Domino servers.

There are several measurements for the Domino user population:

- Registered users
  This is the number of users who can use this Domino server. They have their mail files on this server.

  Except for DASD, the number of registered users by itself does not indicate the resource required for the server. The load on the server depends on the activity of these users.

- Connected users
  These are users who have previously been active on the Domino server and have not been inactive for the user time-out period (Server_Session_Timeout specified in notes.ini).

  Connected users who are not active are not consuming CPU, but they do take up real storage. Thus a high number of connected users would increase the real storage required on the processor.

- Active 15 minute users
  These are users who have been active on the server during the last 15 minute interval.

  Active 15 minute users is the most important measurement of the load on the Domino server. These users have consumed CPU during the previous 15 minutes and are also using more real storage than are connected inactive users.

Based on the number of registered users planned, you will need to determine the percentage who will be active 15 minute users. This information can be derived from other Domino servers that are migrating to z/OS or based on the expected activity rate.

Note that these recommendations are based on our experiences at the time of writing this book. The recommendations will change over time with more exploitation of function. You should talk to your IBM Domino representative to get the latest recommendations.

Registered users per DPAR
It is not possible to give a recommendation for the number of registered users per DPAR without knowing something about how the users will access the server.
The number of registered users in a DPAR is dependent on how active the users are going to be, meaning how often and for how long will they use the server.

**Active 15 minute users per DPAR**
The current recommendation is a maximum of 1,500 active 15 minute users in a DPAR. This number does not change from Domino Release 5 to Domino Release 6. It also does not change when migrating from 31-bit real storage to 64-bit real storage.

This number may increase significantly when Domino and z/OS UNIX System Services support 64-bit virtual storage, which will allow greater than 2 GB address spaces.

**DPARs per LPAR**
The recommendation varies between 31-bit real storage and 64-bit real storage.

For 31-bit real storage, you should have a maximum of two full DPARs (a DPAR with 1,500 active users). With much smaller DPARs you may be able to support three DPARs in an LPAR.

For 64-bit real storage, the current recommendation is to have four or five full DPARs. Theoretically you can have more DPARs, but from current practical experience we recommend this number.

You may have a requirement for a hub DPAR. This DPAR will probably have a lower CPU and real storage requirement than a mail DPAR. It may be possible to include the hub DPAR alongside the mail DPARs if they are not supporting the maximum number of users.

**LPARs per processor**
The IBM 9672 Generations 5 and 6 and zSeries Z800 and Z900 processor support 15 LPARs on a processor.

The number of LPARs you have on the processor will depend on your Domino DPARs configuration and the CPU and real storage requirements for these DPARs. You will need to install an additional processor if your CPU or real storage requirements exceed the capacity of the largest model in your processor range. You may be able to avoid installing an additional processor by upgrading to a larger processor range.

### 6.2 Clustering

A cluster is a group of Domino servers that work together. The servers pass database updates among themselves, using replication, so that the data on the various servers is mirrored on both servers. Up to six Domino servers can be in a single cluster, and they can be running on different platforms.

The benefits of clustering are:

- Higher availability. If one server fails, users can automatically be switched to another server in the cluster.
- Workload balancing across multiple servers.


Now we look at the performance aspects of clustering.
6.2.1 Cluster components

The following additional tasks run on each server in a cluster:

- The Cluster Administration process (Cladmin) performs many of the housekeeping tasks associated with a cluster.
- The Cluster Database Directory Manager task (Cldbdir) on each server creates the Cluster Database Directory and keeps it up-to-date with the most current database information. You can get status information with the show cluster command.
- The Cluster Replicator (Clrepl) task performs the replication of databases among replicas in a cluster. Whenever a change occurs to a database, the database is added to a queue of databases that need to be replicated by Cluster Replicator. When the Cluster Replicator task has finished replicating a database, it selects the next database at the head of the queue.
- The Internet Cluster Manager task manages failover and load balancing of Web client requests for data stored on clustered servers.

You will also see additional databases created:

- CLUSTA4.NSF is the Cluster Analysis database that contains analysis data on the configuration of the cluster. You can use Cluster Analysis to verify that a cluster is set up correctly. Cluster Analysis creates reports about the cluster configuration.
- CLDBDIR.NSF is the Cluster Database Directory that contains a document about each database and replica in the cluster. This document contains information, such as the database name, server, path, replica ID, and other replication and access information.

When a cluster is first defined on a server, the two cluster tasks Cldbdir and Clrepl are started. They are also added to the ServerTasks parameter in notes.ini so that they are started automatically when the server is started in the future.

You can also start additional cluster replicator tasks. It is recommended that, on each server in a cluster, you run one fewer cluster replicator tasks than there are servers in the cluster. In this way there is one cluster replicator task to handle replication to each other server in the cluster. You may need more cluster replicator tasks if you are experiencing a backlog in the cluster replications.

6.2.2 Parameters that affect clustering

The Server Restricted parameter in notes.ini affects clustering. See the Domino 5 Administration Help database for more details.

6.2.3 Costs of clustering

Running clustered servers uses increased resources in the following areas:

- CPU resources are used to send database updates to other servers and apply them to all copies of the database. A cluster with one primary and one secondary server takes 40 to 80 percent more CPU than a non-clustered configuration.
- Real storage is 2.5 to 3.5 as much as a non-clustered configuration, depending on the number of DPARs and LPARs used.
  - DPARs: four times as many will be needed if you use a primary and backup pair. The maximum recommended number of concurrently active 15 minute users (users who have driven a transaction on the server in the last 15 minutes) for a primary cluster
DPAR is 750 versus 1,500 for a non-clustered DPAR. Add to this a backup DPAR for each primary DPAR.

However, only 2.5 times as many DPARs will be needed if you put 800 active users in each DPAR and back it up with three other primary servers with 200 users each.

- LPARs: two to four times as many will be needed, depending on how many DPARs you put in each LPAR.
- I/O write activity doubles since updates are done multiple times, once on each server. There will also be an increase in I/O read activity.
- DASD space doubles if you have two copies of each database. Each server must have its own set of files and databases. It triples if you elect to have three copies of each database. There is no sharing between servers.
- Network traffic increases to pass the updates to all servers in the cluster. Lotus recommends providing a separate network for the server-to-server traffic in a cluster, to ensure good performance.

For each database, you can decide which servers in the cluster contain a replica, from only one server to all servers. Make that decision carefully for each database, in order to limit the number of updates being passed through the cluster and therefore minimize the resource use.

Depending on your configuration, the cost of clustering can be two or more times that of a non-clustered configuration. Compare this to the high availability of the zSeries and the fast (5-minute) restart capability provided with transaction logging to evaluate if this added cost is justified. If so, consider clustering only those users that need higher availability, such as senior executives of your company and other important personnel.

### 6.2.4 Monitoring clustering

You can get statistics on clustering from the log database log.nsf and from the output of the Domino server. The following commands can be used to obtain clustering statistics:

- `show tasks`
- `show statistic replica`
- `show statistic <servername>`

To look for replication backlogs, issue the server command `show statistic replica`. Look at the `Replica.Cluster.WorkQueueDepth` value, which shows the current number of modified databases waiting for replication. If this value is consistently greater than 10, you should consider running additional cluster replicator tasks and also check that you have sufficient network bandwidth between the Domino servers.

### 6.2.5 Effectiveness of Domino clustering

Cluster replication is not a synchronous process. Updates to a database are not made on the other copies of the database until the database has been replicated by the Cluster Replicator task. This may be a matter of seconds—or could be much longer—depending on the performance of the cluster replication process. Large volumes of data to be clustered or insufficient bandwidth to support the replication could cause a backlog to be created.

The data is not in sync during the time between the update to the database and the replication to the other replicas. If during this period a switch is made to an alternate Domino server, then it will not have current data. This should be taken into account when evaluating and planning Domino clustering.
6.2.6 Alternatives to Domino clustering

If the asynchronous operation or the CPU cost of Domino clustering is a concern, then there are other ways to have database files mirrored. These are implemented by Direct Access Storage Device (DASD) subsystems, such as the IBM Enterprise Storage Server.

PPRC

Peer-To-Peer Remote Copy (PPRC) is a synchronous mirroring of the data between DASD subsystems. A write to the primary DASD subsystem is passed to the secondary DASD subsystem; when this is complete, the write is signalled completed to the processor. The write is a cache hit in both primary and secondary DASD subsystems, so the only overhead is the time to pass the write to the secondary and receive the acknowledgement. All reads are satisfied in the primary subsystem with no overhead. There is no CPU overhead in the processor.

Fiber links are required between the two DASD subsystems. Thus the use of PPRC is limited to the normal fiber distances.

Figure 6-3 shows the operation of PPRC for a write I/O. The solid broad lines show the fiber connections between the processors and DASD subsystems.

The Domino production processor issues a write I/O to the primary DASD subsystem (data flow 1), which stores the data in its cache and non-volatile storage. At the same time it sends the data to the secondary DASD subsystem (data flow 2). The secondary DASD subsystem stores the data in its cache and non-volatile storage and then sends an acknowledgement that it is completed to the primary DASD subsystem (data flow 3). Upon receipt of this the primary DASD subsystem can now tell the Domino production processor that the write I/O is complete (data flow 4). The Domino backup processor plays no part in the PPRC flow; it is only waiting to take over if there is a major failure in the primary site.

Figure 6-3   PPRC operation
PPRC-XD

PPRC Extended Distance (PPRC-XD) is a new variation of PPRC which provides asynchronous updating of the secondary DASD subsystem. This allows for far greater distance between the DASD subsystems without the increased write response times, which would be unacceptable if synchronous. A procedure is provided to establish a point-in-time consistency when required.

PPRC-XD requires the same fiber connectivity as PPRC. However, while in asynchronous mode, you do not have the increased write response times of PPRC.

XRC

Extended Remote Copy (XRC) is an asynchronous mirroring of DASD. An OS/390 or z/OS image called the System Data Mover, usually in the secondary site, collects the updated data from the primary DASD subsystem and updates the secondary DASD subsystem. XRC can be implemented at much greater distances than PPRC.

XRC does not have the write response time overhead of PPRC. However, there is a requirement for an LPAR for the additional image to perform the System Data Mover function.

The System Data Mover image requires connectivity to both the primary and secondary DASD subsystems.

Figure 6-4 on page 124 shows the operation of XRC for a write I/O. The solid broad lines show the fiber connections between the processors and DASD subsystems.

The Domino production processor issues a write I/O to the primary DASD subsystem (data flow 1), which stores the data in its cache and non-volatile storage. The primary DASD subsystem now tells the Domino production processor that the write I/O is complete (data flow 2). The Domino server can now continue processing. Asynchronous to the continued Domino processing, the System Data Mover in an LPAR on the Domino backup processor reads any updated data in the primary DASD subsystem (data flow 3). The System Data Mover now issues a write I/O to the secondary DASD subsystem (data flow 4) to mirror the write already completed on the primary DASD subsystem. The secondary DASD subsystem saves the data in its cache and non-volatile storage and signals complete to the System Data Mover (data flow 5).
All three alternatives—PPRC, PPRC-XD, and XRC—maintain the second copy of the Domino data. This could include all the Domino data, not just the databases. However, they do not provide the switchover capability of Domino clustering. You would need to restart the Domino servers again using the secondary DASD subsystem data.

### 6.3 Hubs

It is common in Domino installations, especially in the ones based on other platforms, to have some servers that work like a hub to manage the mail traffic between the servers.

On the zSeries platform, this kind of installation is not normally recommended. It does not bring any performance advantages or network traffic reduction to your system. But there are some exceptions.

- When you are dealing with very high mail traffic coming from the Internet or going to the Internet, you could dedicate a Domino server to act as a gateway to the Internet. See the example in Figure 6-5 on page 125.

  This way, you reduce the mail traffic between your mail servers, and even reduce some of their workload.
You also can use this gateway server as an administration hub, to centralize the administration changes and then replicate these changes to your other Domino servers.

You may consider a Domino Hub server if you have a large number of servers and want to simplify the interconnection definitions. This way each Domino server does not need the connectivity definitions for all the other servers it may need to talk to.

Placement of the hub may be important. For example, DPARs 1 and 2 may have high usage, while DPARs 3 and 4 may have lower utilization. In that case, you would place the hub on the same system image as DPARs 1 and 2 to avoid network TCP/IP traffic. This is because network traffic between DPARS 1 and 2 and the hub would not go out onto the LAN, but would be routed within the system.
Domino monitoring

Domino 6 for z/OS provides a set of tools for monitoring the server. Some of the tools run automatically. Others have to be set up, or started using a server command or started task. We discuss the various sources of monitoring information in this chapter, including the following tools:

- SMF type 108 records
- Notes log
- Domino statistics
7.1 SMF type 108 record

The SMF type 108 record was a new feature in Lotus Domino for S/390 Release 5. It provides data about a Domino server running on an OS/390 or z/OS system. It is a powerful tool for performance monitoring, capacity planning, and problem diagnosis.

To enable this feature:

- Include SYS(TYPE(108)) in the SMFPRMxx PARMLIB member.
- z/OS Security Server (RACF) users must define the user ID of the server to have at least READ access to the BPX.SMF FACILITY class.

You can get the record format from the Domino for z/OS Web page at:


From the Documentation (PDF Format) list, select the document corresponding to your Domino release.

Tip: Although this documentation is included in the manual MVS System Management Facilities, SA22-7630, we recommend that you check the Web page to get the latest format for your Domino release.

See the Domino 6 Administration Help database for more details about monitoring a Domino server.

Note: To effectively monitor Domino, you must be aware of the nature of the data that you are dealing with. Statistics in Domino are normally cumulative, while SMF data is collected during intervals. For example, the number of Domino transactions on the server is the total since it was started or since that stat was explicitly reset. On the other hand, each SMF interval contains only the number of Domino transactions that occurred during that interval.

7.1.1 Record mapping

Table 7-1 shows the record mapping of SMF type 108 records. There are two sections: a common section which appears on all subtypes, and a unique section for each subtype.
### Table 7-1  Record mapping of SMF type 108 records

<table>
<thead>
<tr>
<th>Section</th>
<th>Section name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Sections</td>
<td>Header Section</td>
<td>The common SMF record headers and triplet fields (offset/length/number) that locate the other sections on the record.</td>
</tr>
<tr>
<td></td>
<td>Product Section</td>
<td>The general information about the server and the system that it is running on.</td>
</tr>
<tr>
<td>Subtype 1 (Server Load)</td>
<td>Self-Defining Section</td>
<td>The triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.</td>
</tr>
<tr>
<td></td>
<td>Server Load Section</td>
<td>The counters showing activity at the server level. This is generated at the expiration of the SMF Global Interval (combination of INTVAL and SYNCVAL parameters in the SMFPRMxx PARMLIB member).</td>
</tr>
<tr>
<td></td>
<td>Transaction Section</td>
<td>The data being reported for each NRPC transaction (by type) that is requested from the server by a Notes client. Only transactions with non-zero activity are included. The transaction types shown in this section are Domino internal transactions. One user action, such as opening a database, performs one or more Domino transactions.</td>
</tr>
<tr>
<td></td>
<td>Port Activity Section</td>
<td>The data being reported for each TCP/IP port to which the server is connected.</td>
</tr>
<tr>
<td>Subtype 2 (User Activity)</td>
<td>Self-Defining Section</td>
<td>The triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.</td>
</tr>
<tr>
<td></td>
<td>User Activity Data Section</td>
<td>The data for users by IP address and connection type.</td>
</tr>
<tr>
<td>Subtype 3 (Monitoring and Tuning)</td>
<td>Self-Defining Section</td>
<td>The triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.</td>
</tr>
<tr>
<td></td>
<td>Monitoring and Tuning Data Section</td>
<td>Statistics and configuration parameters for tuning the Domino server.</td>
</tr>
<tr>
<td>Subtype 6 Data Base Activity</td>
<td>Self-Defining Section</td>
<td>The triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.</td>
</tr>
<tr>
<td></td>
<td>Data Base Activity Data Section</td>
<td>The data for Domino Data Base activity.</td>
</tr>
</tbody>
</table>

**Note:** Make sure that the document corresponds to your Domino release level because the record mapping varies. Release 5.0 provided subtype 1. Release 5.02 added some fields to subtype 1. A transaction type table was added in Release 5.01. Release 5.02 added fields to the subtype 1 Product Section and also added subtype 3 tables. Release 5.03 provided subtype 2 and subtype 6. Release 6 added fields to the subtype 1 Server Load section.

**Common sections**

The following sections appear on each of the Type 108 subtype records and are included in the documentation once.
**Header section**

This section contains the common SMF record headers fields and the triplet fields (offset/length/number) that locate the other sections on the record.

Table 7-2  Fields of header section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108LEN</td>
<td>2</td>
<td>binary</td>
<td>Record length. This field and the next field (total of four bytes) form the RDW (record descriptor word).</td>
</tr>
<tr>
<td></td>
<td>SMF108SLDMSENTRAS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SMF108SEG</td>
<td>2</td>
<td>binary</td>
<td>Segment descriptor (see record length field).</td>
</tr>
<tr>
<td>4</td>
<td>SMF108FLG</td>
<td>1</td>
<td>binary</td>
<td>System indicator:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>SMF108RTY</td>
<td>1</td>
<td>binary</td>
<td>Record type 108 (X'6C').</td>
</tr>
<tr>
<td>6</td>
<td>SMF108TME</td>
<td>4</td>
<td>binary</td>
<td>Time since midnight, in hundredths of a second, that the record was moved into the SMF buffer.</td>
</tr>
<tr>
<td>10</td>
<td>SMF108DTE</td>
<td>4</td>
<td>packed</td>
<td>Date when the record was moved into the SMF buffer, in the form 0cyydddF.</td>
</tr>
<tr>
<td>14</td>
<td>SMF108SID</td>
<td>4</td>
<td>EBCDIC</td>
<td>System identification (from the SID parameter).</td>
</tr>
<tr>
<td>18</td>
<td>SMF108SSI</td>
<td>4</td>
<td>EBCDIC</td>
<td>Subsystem identification.</td>
</tr>
<tr>
<td>22</td>
<td>SMF108STP</td>
<td>2</td>
<td>binary</td>
<td>Record Subtype</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SubType Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>24</td>
<td>SMF108PRO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Product Section.</td>
</tr>
<tr>
<td>28</td>
<td>SMF108PRL</td>
<td>2</td>
<td>binary</td>
<td>Length of Product Section.</td>
</tr>
<tr>
<td>30</td>
<td>SMF108PRN</td>
<td>2</td>
<td>binary</td>
<td>Number of Product Sections (should be '1').</td>
</tr>
<tr>
<td>32</td>
<td>SMF108SSO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Self-Defining Section.</td>
</tr>
<tr>
<td>36</td>
<td>SMF108SSL</td>
<td>2</td>
<td>binary</td>
<td>Length of Self-Defining Section.</td>
</tr>
<tr>
<td>38</td>
<td>SMF108SSN</td>
<td>2</td>
<td>binary</td>
<td>Number of Self-Defining Sections (should be '1').</td>
</tr>
</tbody>
</table>
Product section
This section contains the general information about the server and the system that it is running on.

Table 7-3  Fields of product section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108PPRRVN</td>
<td>4</td>
<td>binary</td>
<td>Record Version Number (starting with '1' for release 5.0a) (Set to 2 for release 5.01) (Set to 3 for release 5.02). (Set to 4 for release 5.03).</td>
</tr>
<tr>
<td>4</td>
<td>SMF108PRPVN</td>
<td>8</td>
<td>EBCDIC</td>
<td>Product Version ('5.0' for example). This is the first eight bytes of the Product Version string and may contain text or other characters after the number.</td>
</tr>
<tr>
<td>12</td>
<td>SMF108PRSVN</td>
<td>32</td>
<td>EBCDIC</td>
<td>Server Name (used to identify partitioned servers).</td>
</tr>
<tr>
<td>44</td>
<td>SMF108PRSPN</td>
<td>8</td>
<td>EBCDIC</td>
<td>Sysplex Name (ECVTSPPLX field in cvt/ecvt).</td>
</tr>
<tr>
<td>52</td>
<td>SMF108PRSYN</td>
<td>8</td>
<td>EBCDIC</td>
<td>System Name (CVTSYNSN field in cvt/ecvt).</td>
</tr>
<tr>
<td>60</td>
<td>SMF108PROSL</td>
<td>8</td>
<td>EBCDIC</td>
<td>z/OS System Level (CVTPRNDN field in cvt/ecvt).</td>
</tr>
<tr>
<td>68</td>
<td>SMF108PRSTARTT</td>
<td>8</td>
<td>binary</td>
<td>Interval Start Time.</td>
</tr>
<tr>
<td>76</td>
<td>SMF108PRIENDT</td>
<td>8</td>
<td>binary</td>
<td>Interval End Time.</td>
</tr>
<tr>
<td>84</td>
<td>SMF108CVTTV</td>
<td>4</td>
<td>binary</td>
<td>CVTTV GMT offset time.</td>
</tr>
</tbody>
</table>

Subtype 1 - server load
This subtype contains counts of activity done by the server running on the z/OS system.

Self-defining section
This section contains the triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.

Table 7-4  Fields of self-defining section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108SLO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Server Load Section</td>
</tr>
<tr>
<td>4</td>
<td>SMF108SLL</td>
<td>2</td>
<td>binary</td>
<td>Length of Server Load Section</td>
</tr>
<tr>
<td>6</td>
<td>SMF108SLN</td>
<td>2</td>
<td>binary</td>
<td>Number of Server Load Sections (should be '1')</td>
</tr>
<tr>
<td>8</td>
<td>SMF108TRO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Transaction Section</td>
</tr>
<tr>
<td>12</td>
<td>SMF108TRL</td>
<td>2</td>
<td>binary</td>
<td>Length of Transaction Section</td>
</tr>
<tr>
<td>14</td>
<td>SMF108TRN</td>
<td>2</td>
<td>binary</td>
<td>Number of Transaction Sections (1 per transaction type processed)</td>
</tr>
<tr>
<td>16</td>
<td>SMF108PTO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Port Activity Section</td>
</tr>
<tr>
<td>20</td>
<td>SMF108PTL</td>
<td>2</td>
<td>binary</td>
<td>Length of Port Activity Section</td>
</tr>
<tr>
<td>22</td>
<td>SMF108PTN</td>
<td>2</td>
<td>binary</td>
<td>Number of Port Activity Sections (1 per TCP/IP port)</td>
</tr>
</tbody>
</table>
### Server load section

This section contains the counters showing activity at the server level (globally).

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108SLCU</td>
<td>4</td>
<td>binary</td>
<td>Current number of users</td>
</tr>
<tr>
<td>4</td>
<td>SMF108SLUA</td>
<td>4</td>
<td>binary</td>
<td>Number of currently connected users that are currently active</td>
</tr>
<tr>
<td>8</td>
<td>SMF108SLUA1M</td>
<td>4</td>
<td>binary</td>
<td>Number of currently connected users that have been active within the last 1 minute</td>
</tr>
<tr>
<td>12</td>
<td>SMF108SLUA3M</td>
<td>4</td>
<td>binary</td>
<td>Number of currently connected users that have been active within the last 3 minutes</td>
</tr>
<tr>
<td>16</td>
<td>SMF108SLUA5M</td>
<td>4</td>
<td>binary</td>
<td>Number of currently connected users that have been active within the last 5 minutes</td>
</tr>
<tr>
<td>20</td>
<td>SMF108SLUA15M</td>
<td>4</td>
<td>binary</td>
<td>Number of currently connected users that have been active within the last 15 minutes</td>
</tr>
<tr>
<td>24</td>
<td>SMF108SLUA30M</td>
<td>4</td>
<td>binary</td>
<td>Number of currently connected users that have been active within the last 30 minutes</td>
</tr>
<tr>
<td>28</td>
<td>SMF108SLDMSENTL</td>
<td>4</td>
<td>binary</td>
<td>Number of Domino mail messages delivered to local users</td>
</tr>
<tr>
<td>32</td>
<td>SMF108SLDMSENTLAS</td>
<td>4</td>
<td>binary</td>
<td>Average size of Domino mail and SMTP messages delivered to local users</td>
</tr>
<tr>
<td>36</td>
<td>SMF108SLDMSENTR</td>
<td>4</td>
<td>binary</td>
<td>Number of Domino mail and SMTP messages sent to other servers</td>
</tr>
<tr>
<td>40</td>
<td>SMF108SLDMSENTRAS</td>
<td>4</td>
<td>binary</td>
<td>Average size of Domino mail messages sent to other servers</td>
</tr>
<tr>
<td>44</td>
<td>SMF108SLSMREC</td>
<td>4</td>
<td>binary</td>
<td>Number of SMTP messages received from other servers during interval</td>
</tr>
<tr>
<td>48</td>
<td>SMF108SLSMRECAS</td>
<td>4</td>
<td>binary</td>
<td>Average size of SMTP messages received from other servers during interval</td>
</tr>
<tr>
<td>52</td>
<td>SMF108SLSMSENTR</td>
<td>4</td>
<td>binary</td>
<td>Number of SMTP messages sent to other servers during interval</td>
</tr>
<tr>
<td>56</td>
<td>SMF108SLSMSENTRAS</td>
<td>4</td>
<td>binary</td>
<td>Average size of SMTP messages sent to other servers during interval</td>
</tr>
<tr>
<td>60</td>
<td>SMF108SLTRANS</td>
<td>4</td>
<td>binary</td>
<td>Total number of transactions processed during interval [Note 1]</td>
</tr>
<tr>
<td>64</td>
<td>SMF108SLSVREPL</td>
<td>4</td>
<td>binary</td>
<td>Number of replications initiated by this server</td>
</tr>
<tr>
<td>68</td>
<td>SMF108SLNWSESIN</td>
<td>4</td>
<td>binary</td>
<td>Number of incoming (to the server from clients) sessions established during the interval [Note 4]</td>
</tr>
<tr>
<td>72</td>
<td>SMF108SLNWSESOUT</td>
<td>4</td>
<td>binary</td>
<td>Number of outgoing sessions established during the interval [Note 4]</td>
</tr>
<tr>
<td>76</td>
<td>SMF108SLNWBR</td>
<td>4</td>
<td>binary</td>
<td>Number of network bytes/1024 received during interval [Note 4]</td>
</tr>
<tr>
<td>80</td>
<td>SMF108SLNWBS</td>
<td>4</td>
<td>binary</td>
<td>Number of network bytes/1024 sent during interval [Note 4]</td>
</tr>
<tr>
<td>84</td>
<td>SMF108SLTT</td>
<td>2</td>
<td>binary</td>
<td>Total number of physical thread pool threads [Note 2]</td>
</tr>
</tbody>
</table>
Table 7-5 notes:
1. This “transaction” means Domino internal transactions. One user action, such as ceasing a database, performs one or more Domino internal transactions. The transaction section in SMF type108 subtype1 provides additional information for Domino transactions, such as the number and response time of each transaction type processed during the interval.
2. This corresponds to the value defined in SERVER_POOL_TASKS parameter in notes.ini.
3. These fields contain no data when you use Domino for the IBM HTTP Server Connector.
4. For Release 5.01 or higher, these fields will be set to zero and recorded in the Port Activity Section.
5. These are new with Domino 6 for z/OS.

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>SMF108SLVTIU</td>
<td>2</td>
<td>binary</td>
<td>Number of virtual thread pool threads currently in use</td>
</tr>
<tr>
<td>88</td>
<td>SMF108SLAIOR</td>
<td>4</td>
<td>binary</td>
<td>Number of async I/O reads during interval</td>
</tr>
<tr>
<td>92</td>
<td>SMF108SLAIOW</td>
<td>4</td>
<td>binary</td>
<td>Number of async I/O writes during interval</td>
</tr>
<tr>
<td>96</td>
<td>SMF108SLPOP3R</td>
<td>4</td>
<td>binary</td>
<td>Number of POP3 reads during interval</td>
</tr>
<tr>
<td>100</td>
<td>SMF108SLIMAPR</td>
<td>4</td>
<td>binary</td>
<td>Number of IMAP reads during interval</td>
</tr>
<tr>
<td>104</td>
<td>SMF108SLHTTPR</td>
<td>4</td>
<td>binary</td>
<td>Number of HTTP reads during interval [Note 3]</td>
</tr>
<tr>
<td>108</td>
<td>SMF108SLHTTPW</td>
<td>4</td>
<td>binary</td>
<td>Number of HTTP writes during interval [Note 3]</td>
</tr>
<tr>
<td>112</td>
<td>SMF108SLVTIUMAX</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of virtual thread pool threads in use during interval</td>
</tr>
<tr>
<td>114</td>
<td>SMF108SLTASKS</td>
<td>2</td>
<td>binary</td>
<td>Number of tasks currently in use</td>
</tr>
<tr>
<td>116</td>
<td>SMF108SLTASKSMAX</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of tasks in use during interval</td>
</tr>
<tr>
<td>118</td>
<td>SMF108SLPTIU</td>
<td>2</td>
<td>binary</td>
<td>Number of physical thread pool threads currently in use</td>
</tr>
<tr>
<td>120</td>
<td>SMF108SLPTIUMAX</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of physical thread pool threads in use during interval</td>
</tr>
<tr>
<td>122</td>
<td>SMF108SLPAD</td>
<td>2</td>
<td></td>
<td>Pad to next 4 byte boundary [Note 5]</td>
</tr>
<tr>
<td>124</td>
<td>SMF108SLDominoCache Command Count</td>
<td>4</td>
<td>binary</td>
<td>HTTP cache command count [Note 5]</td>
</tr>
<tr>
<td>128</td>
<td>SMF108SLDominoCache DesignCount</td>
<td>4</td>
<td>binary</td>
<td>HTTP cache design count [Note 5]</td>
</tr>
<tr>
<td>132</td>
<td>SMF108SLDominoCache SessionCount</td>
<td>4</td>
<td>binary</td>
<td>HTTP cache session count [Note 5]</td>
</tr>
<tr>
<td>136</td>
<td>SMF108SLDominoCache UserCount</td>
<td>4</td>
<td>binary</td>
<td>HTTP cache user count [Note 5]</td>
</tr>
<tr>
<td>140</td>
<td>SMF108SLDominoRequestsTotal</td>
<td>4</td>
<td>binary</td>
<td>HTTP total requests [Note 5]</td>
</tr>
</tbody>
</table>
**Transaction section**
This section contains the data being reported for each NRPC transaction (by type) that is requested of the server by a Notes client. Only transactions with non-zero activity counts are included.

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108TRTYPE</td>
<td>4</td>
<td>binary</td>
<td>Transaction type.</td>
</tr>
<tr>
<td>4</td>
<td>SMF108TRTYPENP</td>
<td>4</td>
<td>binary</td>
<td>Number of transactions of type processed during interval.</td>
</tr>
<tr>
<td>8</td>
<td>SMF108TRTYPETA</td>
<td>4</td>
<td>binary</td>
<td>Total accumulated response time, in milliseconds, for all transactions of type that completed during interval.</td>
</tr>
<tr>
<td>12</td>
<td>SMF108TRTYPENW</td>
<td>4</td>
<td>binary</td>
<td>Total accumulated net wait time, in milliseconds, for all transactions of type that completed during interval. This is the time the server has been waiting for clients to respond.</td>
</tr>
</tbody>
</table>

**Port activity section**
This section contains the data being reported for each TCP/IP port to which the server has a connection.

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108PTNAME</td>
<td>8</td>
<td>EBCDIC</td>
<td>The first eight bytes of the TCP/IP port. (‘TCPIP’ for example)</td>
</tr>
<tr>
<td>8</td>
<td>SMF108PTNWSESIN</td>
<td>4</td>
<td>binary</td>
<td>Number of incoming sessions processed during the interval (client to server connection)</td>
</tr>
<tr>
<td>12</td>
<td>SMF108PTNWSESOUT</td>
<td>4</td>
<td>binary</td>
<td>Number of outgoing sessions processed during the interval</td>
</tr>
<tr>
<td>16</td>
<td>SMF108PTNWBR</td>
<td>4</td>
<td>binary</td>
<td>Total number of bytes/1024 received for this port during the interval</td>
</tr>
<tr>
<td>20</td>
<td>SMF108PTNWBS</td>
<td>4</td>
<td>binary</td>
<td>Total number of bytes/1024 sent for this port during the interval</td>
</tr>
</tbody>
</table>

**Subtype 2 - User activity**
This subtype reports Domino User activity for the different protocols Domino supports.

**Self-defining section**
This section contains the triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108UDO</td>
<td>4</td>
<td>binary</td>
<td>Offset to data section</td>
</tr>
<tr>
<td>4</td>
<td>SMF108UDL</td>
<td>2</td>
<td>binary</td>
<td>Length of data section</td>
</tr>
<tr>
<td>6</td>
<td>SMF108UDN</td>
<td>2</td>
<td>binary</td>
<td>Number of data sections</td>
</tr>
</tbody>
</table>
User activity data section
This section contains the data for users by IP address and connection type.

Table 7-9  Fields of user activity data section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108UIPA</td>
<td>16</td>
<td>EBCDIC</td>
<td>IP address presenting the request for service</td>
</tr>
<tr>
<td>16</td>
<td>SMF108UTYPE</td>
<td>4</td>
<td>EBCDIC</td>
<td>Type of connection to the Domino server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>User/Type/Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NRPC/Domino/mail and database server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HTTP/Domino/http server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IMAP/IMAP/mail server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>POP3/POP3/mail server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTP/SMTP/server</td>
</tr>
<tr>
<td>20</td>
<td>SMF108UNAME</td>
<td>32</td>
<td>EBCDIC</td>
<td>Notes user name for NRPC clients</td>
</tr>
<tr>
<td>52</td>
<td>SMF108UCPU</td>
<td>8</td>
<td>binary</td>
<td>CPU time used by this user</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>STCK</td>
<td>format</td>
</tr>
<tr>
<td>60</td>
<td>SMF108UBR</td>
<td>4</td>
<td>binary</td>
<td>number of bytes read this interval</td>
</tr>
<tr>
<td>64</td>
<td>SMF108UBW</td>
<td>4</td>
<td>binary</td>
<td>number of bytes written this interval</td>
</tr>
</tbody>
</table>

Subtype 3 - Monitoring and tuning
This subtype monitors some statistics and certain configuration parameters used by the server.

Self-defining section
This section contains the triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.

Table 7-10  Fields of self-defining section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108MTO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Data Section</td>
</tr>
<tr>
<td>4</td>
<td>SMF108MTL</td>
<td>2</td>
<td>binary</td>
<td>Length of Data Sections</td>
</tr>
<tr>
<td>6</td>
<td>SMF108MTN</td>
<td>2</td>
<td>binary</td>
<td>Number of Data Sections (should be ‘1’)</td>
</tr>
</tbody>
</table>

Monitoring and tuning data section
This section contains some statistics and certain configuration parameters for tuning the Domino server.

Table 7-11  Fields of monitoring and tuning data section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108MTMAXUSERS</td>
<td>4</td>
<td>binary</td>
<td>Maximum number of users</td>
</tr>
<tr>
<td>4</td>
<td>SMF108MTMAXCONTR</td>
<td>4</td>
<td>binary</td>
<td>Limit for number of concurrent transactions</td>
</tr>
<tr>
<td>8</td>
<td>SMF108MTMAXCONSES</td>
<td>4</td>
<td>binary</td>
<td>Maximum number of sessions to run concurrently</td>
</tr>
<tr>
<td>12</td>
<td>SMF108MTSESTIMEOUT</td>
<td>2</td>
<td>binary</td>
<td>Number of minutes in time-out</td>
</tr>
</tbody>
</table>
### Subtype 6 - Database activity

This subtype reports Domino-specific data for Domino databases.

#### Self-defining section

This section contains the triplet fields (offset/length/number) that locate the specific sections for this subtype on the record.

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>SMF108MTUPMAX</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of concurrent update tasks</td>
</tr>
<tr>
<td>16</td>
<td>SMF108MTMAILBOXES</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of mail.boxes</td>
</tr>
<tr>
<td>18</td>
<td>SMF108MTREPMAX</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of replicators (concurrent)</td>
</tr>
<tr>
<td>20</td>
<td>SMF108MTNSFPOOL</td>
<td>4</td>
<td>binary</td>
<td>Maximum size of nsf buffer pool (bytes/4096)</td>
</tr>
<tr>
<td>24</td>
<td>SMF108MTNSFPOOLIU</td>
<td>4</td>
<td>binary</td>
<td>Number of bytes in nsf buffer pool (in use)</td>
</tr>
<tr>
<td>28</td>
<td>SMF108MTDBCENAB</td>
<td>4</td>
<td>binary</td>
<td>dbcache enabled = 1, 0 if disabled</td>
</tr>
<tr>
<td>29</td>
<td>RESERVECHAR</td>
<td>1</td>
<td>N/A</td>
<td>Reserved for alignment</td>
</tr>
<tr>
<td>32</td>
<td>SMF108MTDBCMAXE</td>
<td>3</td>
<td>binary</td>
<td>Maximum number of dbcache entries</td>
</tr>
<tr>
<td>36</td>
<td>SMF108MTDBCCE</td>
<td>4</td>
<td>binary</td>
<td>Number of dbcache (current entries)</td>
</tr>
<tr>
<td>40</td>
<td>SMF108MTDBCIDBO</td>
<td>4</td>
<td>binary</td>
<td>Number of dbcache (initial db opens)</td>
</tr>
<tr>
<td>44</td>
<td>SMF108MTDBCOCR</td>
<td>4</td>
<td>binary</td>
<td>Number of dbcache (overcrowding rejections)</td>
</tr>
<tr>
<td>48</td>
<td>SMF108MTDBCHITS</td>
<td>4</td>
<td>binary</td>
<td>Number of dbcache (hits)</td>
</tr>
<tr>
<td>52</td>
<td>SMF108MTDBCHWM</td>
<td>4</td>
<td>binary</td>
<td>dbcache (high water mark)</td>
</tr>
<tr>
<td>56</td>
<td>SMF108MTSATH</td>
<td>2</td>
<td>binary</td>
<td>Server availability threshold</td>
</tr>
<tr>
<td>58</td>
<td>SMF108MTSAX</td>
<td>2</td>
<td>binary</td>
<td>Server availability index</td>
</tr>
<tr>
<td>60</td>
<td>SMF108MTNIFS</td>
<td>2</td>
<td>binary</td>
<td>Database.NIFPool.Size (in bytes)</td>
</tr>
<tr>
<td>64</td>
<td>SMF108MTNIFN</td>
<td>4</td>
<td>binary</td>
<td>Database.NIFPool.Used</td>
</tr>
<tr>
<td>68</td>
<td>SMF108MTNSFS</td>
<td>4</td>
<td>binary</td>
<td>Database.NSFPool.Size (in bytes)</td>
</tr>
<tr>
<td>72</td>
<td>SMF108MTNSFN</td>
<td>4</td>
<td>binary</td>
<td>Database.NSFPool.Used</td>
</tr>
<tr>
<td>76</td>
<td>SMF108MTDBPR</td>
<td>4</td>
<td>binary</td>
<td>Number of Database.BufferPool Reads (no longer set by the server)</td>
</tr>
<tr>
<td>80</td>
<td>SMF108MTDBPW</td>
<td>4</td>
<td>binary</td>
<td>Number of Database.BufferPool Writes (no longer set by the server)</td>
</tr>
<tr>
<td>84</td>
<td>SMF108MTMMXFER</td>
<td>4</td>
<td>binary</td>
<td>Maximum number of mail transfer threads</td>
</tr>
<tr>
<td>86</td>
<td>SMF108MTMMXDLV</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of mail delivery threads</td>
</tr>
<tr>
<td>88</td>
<td>SMF108MTMMXCONXFR</td>
<td>2</td>
<td>binary</td>
<td>Maximum number of concurrent mail transfer threads</td>
</tr>
</tbody>
</table>
Table 7-12 Fields of self-defining section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108DBO</td>
<td>4</td>
<td>binary</td>
<td>Offset to Data Section</td>
</tr>
<tr>
<td>4</td>
<td>SMF108DBL</td>
<td>2</td>
<td>binary</td>
<td>Length of Data Sections</td>
</tr>
<tr>
<td>6</td>
<td>SMF108DBN</td>
<td>2</td>
<td>binary</td>
<td>Number of Data Sections</td>
</tr>
</tbody>
</table>

**Database activity data section**
This section contains the data for Domino database activity.

Table 7-13 Fields of database activity data section

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SMF108DBNAME</td>
<td>64</td>
<td>EBCDIC</td>
<td>Last 64 characters of the Database Name</td>
</tr>
<tr>
<td>64</td>
<td>SMF108DBINDEX</td>
<td>4</td>
<td>binary</td>
<td>Number of indexing operations started on this database by the server</td>
</tr>
<tr>
<td>68</td>
<td>SMF108DBREPS</td>
<td>4</td>
<td>binary</td>
<td>Number of replications on this database initiated by this server</td>
</tr>
<tr>
<td>72</td>
<td>SMF108DBOCADDS</td>
<td>4</td>
<td>binary</td>
<td>Number of documents added to this database</td>
</tr>
<tr>
<td>76</td>
<td>SMF108DBOCDELS</td>
<td>4</td>
<td>binary</td>
<td>Number of documents deleted from this database</td>
</tr>
</tbody>
</table>

### 7.1.2 RMF Lotus Domino server report

Starting with OS/390 V2R10 (or as far back as OS/390 V2R7 with PTFs applied), RMF enhanced the Postprocessor. It processes SMF record type 108 subtypes 1 and 3 written by Lotus Domino R5 and later releases. It produces reports that provide information on server load, as well as the number and type of messages that the server handles.

The Lotus Domino server report provides information about the activities of a server. The information can be used to analyze the activities of the server in case of problems.

The report consists of two parts:

- Lotus Domino server summary
  The summary contains one line for each Domino server in the SMF input file.
- Lotus Domino server details, including the following sections:
  - Definition data (provided by record type 108-3)
  - Performance data (provided by record type 108-3)
  - Load data (provided by record type 108-1)

Figure 7-1 on page 138 is sample JCL to generate a Lotus Domino server report. The first step is to sort the SMF records into time sequence and the second step is to produce a Domino report for a 30 minute interval between the times 14:00 and 15:00.
Lotus Domino server summary

Figure 7-2 is an example of a Lotus Domino server summary report; Table 7-14 defines the fields in the report.

Table 7-14 Fields in the Domino server summary report

<table>
<thead>
<tr>
<th>Field Heading</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVER NAME</td>
<td>Name of the Domino server</td>
</tr>
<tr>
<td>AVAILABLE</td>
<td>Total time (hh:mm:ss) the server was available during the interval</td>
</tr>
<tr>
<td>USERS CONNECTED</td>
<td>Average number of currently connected users</td>
</tr>
<tr>
<td>USERS ACTIVE</td>
<td>Average number of currently active users</td>
</tr>
<tr>
<td>TASKS</td>
<td>Average number of tasks currently in use</td>
</tr>
<tr>
<td>TRANSACTION RATE</td>
<td>Rate of all transactions processed during the interval</td>
</tr>
<tr>
<td>ASYNC I/O RATE - READS</td>
<td>Rate of asynchronous reads</td>
</tr>
</tbody>
</table>
Lotus Domino server details

Figure 7-3 is an example of a Lotus Domino server details report; Table 7-15 on page 140 defines the fields in the report.

<table>
<thead>
<tr>
<th>Field Heading</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASYNC I/O RATE - WRITES</td>
<td>Rate of asynchronous writes</td>
</tr>
<tr>
<td>MAIL RATE - DELIVERED</td>
<td>Rate of Domino mail messages delivered to local users</td>
</tr>
<tr>
<td>MAIL RATE - SENT</td>
<td>Rate of Domino mail messages sent to other servers</td>
</tr>
<tr>
<td>SMTP RATE - READS</td>
<td>Rate of SMTP messages received from other servers</td>
</tr>
<tr>
<td>SMTP RATE - WRITES</td>
<td>Rate of SMTP messages sent to other servers</td>
</tr>
</tbody>
</table>

Lotus Domino server details

Figure 7-3 is an example of a Lotus Domino server details report; Table 7-15 on page 140 defines the fields in the report.
### Table 7-15  Fields in the Lotus Domino server details report

<table>
<thead>
<tr>
<th>Field Heading</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Name of Domino server</td>
</tr>
<tr>
<td><strong>User Activity</strong></td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>Maximum number of users that are allowed to access the server. The value 0 means that there is no limit.</td>
</tr>
<tr>
<td>CONNECTED</td>
<td>Number of current users (connections).</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Number of active users at the point of time at the end of the interval.</td>
</tr>
<tr>
<td>WITHIN n MIN</td>
<td>Number of currently connected users that have been active within the last 1, 3, 5, 15, and 30 minutes.</td>
</tr>
<tr>
<td><strong>Tasks</strong></td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>Maximum number of tasks in use during the interval.</td>
</tr>
<tr>
<td>CURRENT</td>
<td>Number of tasks currently in use at the point of time at the end of the interval.</td>
</tr>
<tr>
<td>MAX UPDATES</td>
<td>Maximum number of concurrent update tasks.</td>
</tr>
<tr>
<td>MAX REPLICS</td>
<td>Maximum number of concurrent replicator tasks.</td>
</tr>
<tr>
<td>COUNT REPLICS</td>
<td>Number of replications initiated by this server.</td>
</tr>
<tr>
<td><strong>Messages</strong></td>
<td></td>
</tr>
<tr>
<td>MAILBOXES</td>
<td>Number of mail boxes.</td>
</tr>
<tr>
<td>MAIL DELIVERED</td>
<td>The number, rate and average size of Domino mail messages delivered to local users.</td>
</tr>
<tr>
<td>MAIL SENT</td>
<td>Domino mail messages sent to other servers.</td>
</tr>
<tr>
<td>SMTP RECEIVED</td>
<td>SMTP messages received from other servers.</td>
</tr>
<tr>
<td>SMTP SENT</td>
<td>SMTP messages sent to other servers.</td>
</tr>
<tr>
<td><strong>Access Rates</strong></td>
<td></td>
</tr>
<tr>
<td>AS I/O READ</td>
<td>Rate of asynchronous I/O reads.</td>
</tr>
<tr>
<td>AS I/O WRITE</td>
<td>Rate of asynchronous I/O writes.</td>
</tr>
<tr>
<td>POP3 READ</td>
<td>Rate of POP3 reads.</td>
</tr>
<tr>
<td>IMAP READ</td>
<td>Rate of IMAP reads.</td>
</tr>
<tr>
<td>DOMINO READ</td>
<td>Rate of Domino reads.</td>
</tr>
<tr>
<td>DOMINO WRITE</td>
<td>Rate of Domino writes.</td>
</tr>
<tr>
<td><strong>Database Cache</strong></td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td>Status of the database cache: either OK or ? (undefined).</td>
</tr>
<tr>
<td>MAX ENTRIES</td>
<td>Maximum number of database entries allowed in cache at any one time.</td>
</tr>
<tr>
<td>CURRENT ENTRIES</td>
<td>Number of current entries.</td>
</tr>
<tr>
<td>HIGH WATER MARK</td>
<td>High water mark.</td>
</tr>
<tr>
<td>INITIAL DB OPENS</td>
<td>Number of initial database opens.</td>
</tr>
<tr>
<td>REJECTIONS</td>
<td>Number of overcrowding rejections.</td>
</tr>
</tbody>
</table>
### Field Heading | Meaning
--- | ---
**HITS** | Hits in database cache.

**Virtual Threads [Note 1]**

**MAX** | Maximum number of virtual thread pool threads.
**CURRENT** | Number of virtual thread pool threads currently in use.

**Physical Threads**

**MAX** | Maximum number of physical thread pool threads in use.
**CURRENT [Note 3]** | Number of physical thread pool threads currently in use.
**TOTAL [Note 2]** | Total number of physical thread pool threads.

**Availability**

**THRESHOLD** | Server availability threshold.
**INDEX** | Server availability index.

**NSF Buffer Pool**

**MAX** | Maximum size (in bytes) of the NSF (Notes Storage Facility) buffer pool.
**CURRENT** | Number of bytes of the NSF buffer pool currently in use.

**Transaction Activity [Note 4]**

**MAXIMAL CONCURRENT** | Limit for number of concurrent transactions on a server.

**Top-10 List of Transaction Types** (Sorted by COUNT and by R/T TOTAL)

**TYPE** | Transaction type.
**NAME** | Transaction name.
**COUNT** | Number of transactions processed during interval.
**%TOTAL** | Percentage based on all transactions.
**RATE** | Rate of processed transactions.
**R/T AVG** | Average response time (milliseconds).
**R/T TOTAL** | Total response time (milliseconds) of all transactions that completed during the interval.

**Port Activity**

**MAX CONCURRENT SESSIONS** | Maximum number of sessions that can run concurrently on the server.
**SESSION TIMEOUT** | Time limit (minutes) after which idle connections are terminated.
**NAME** | Port name.
**SESSIONS IN** | Count and rate of incoming sessions (from clients to the server) established during the interval.
**SESSIONS OUT** | Count and rate of outgoing sessions established during the interval.
**SENT (KB)** | Number of K bytes sent to the network.
**RECEIVED (KB)** | Number of K bytes received from the network.
Notes for Table 7-15:

1. In Lotus Domino for S/390 Release 5, a thread mechanism called the *thread pool model* was introduced to handle communication requests from Notes clients. When a user connects to the server, a client session is created with a unique socket descriptor and a virtual thread ID is assigned. When a client makes a request to the server, a physical thread is associated with the virtual thread and does the associated work on behalf of the client session.

   For more information on the thread pool model, see “Address spaces for user connections” on page 25.

2. This value corresponds to the value specified in the SERVER_POOL_TASKS parameter in notes.ini.

3. If this value increases above normal usage, it is usually an early indicator of a bottleneck or capacity problem.

4. This “transaction” refers to Domino transactions. One user action, such as ceasing a database, performs one or more Domino transactions.

   The transaction rate should be an indication of the use of the server. The transaction rate allows you to check whether the load across several servers is balanced.

   The transaction activity statistics give measurements for the Domino internal transactions. It is difficult to relate them to the user activity on the server, so you may find analysis of these measurements provides only limited performance information.

### 7.2 Other monitoring tools

In this section we identify some additional tools and techniques that can be used to monitor Domino performance.

#### 7.2.1 The domps command

The *domps* command was introduced in Domino R5 and has been modified for Domino 6. It is a UNIX command that shows detailed information about the processes and threads that the Domino server is running in the UNIX environment. Appendix B, “Using the domps command” on page 171 shows the parameters and provides a sample of the output for each parameter.

#### 7.2.2 Notes log

Every Domino server records information about server activities in the log database (log.nsf). The information includes:

- Database usage by user
- Database size
- Mail routing, replication and other events
- Usage of the system by user, including:
  - User name
  - Connect time
  - Number of documents read and written, by database
  - Amount of data transferred across the network
  - Number of transactions run
You can choose to collect replication and client session event records when you initially set up the server. You can change these settings later with the Log_Replication and Log_Sessions parameters in notes.ini.

7.2.3 Statistics and events log

The statistics and events database events4.nsf is used to configure Notes server event handling, statistic monitoring, ACL monitoring, and replication monitoring. The database contains the names of all statistics monitored by the server, and thresholds for producing event records. It also contains error and status messages from the server.

This database can be used to:

- Configure Notes server event handling
- Look up information about a specific statistic or event message
- Set statistic thresholds
- Assign types and severities to server events

You run events by running the server task Event.

7.2.4 Statistics and reporting database

If you run the reporter task on the server, a set of reports will be created at scheduled intervals. These are put in the statistics and reporting database statrep.nsf. You can specify the interval between records. For information on setting up the reporter task, see the Domino 6 Administration Help database.

Reporting period

Be careful about the reporting period for the various statistics. There are several types of statistics in the interval record:

- A sample at the end of the interval
- Accumulation for the duration of the interval
- Maximum value since the server started

Unfortunately, there are not many values accumulated during the interval.

Domino statistics

Statistics that you can use for performance monitoring are:

- **Server.Trans.PerMinute.Peak** is the peak number of transactions during a one-minute interval since the server started. It is *not* the peak during this interval. You are given the date and time this occurred.
- **Server.Users** is the number of users with connections to the server at the end of the interval when statistics were collected.

There are also counters for the number of calendar and mail requests, and also agent processes. These may be of use if you are investigating the average profile of your mail users.

Coordinating SMF, RMF, and Domino data collection

If you wish to correlate the Domino statistics with the SMF and RMF data, you should try to coincide the start of Domino statistics collection with the start of the SMF and RMF intervals. SMF and RMF have controls to synchronize the time of interval collection to the time of day,
for example at zero and 30 minutes after each hour. Domino has no such control. It records
statistics at regular intervals after the statistics task is started. Therefore, to get Domino to
collect records at the same time as SMF and RMF, you must start the statistics task at the
desired time.

Statistics commands
You can also use the load stats <servername> command to create statistics on demand for a
remote server and the show statistics server command to display the complete list of
statistics for the server.

A sample of a show statistics command is shown in Appendix D, “Sample output of the show
statistics command” on page 181.

Platform statistics
Platform statistics are new for Domino 6, and they provide another way to gain insight into the
combined behavior of Domino and z/OS. Platform stats can be thought of as the mirror image
to the SMF 108 record. Performance information is gathered from z/OS and stored as
Domino statistics that can be collected and processed just like any other Domino stats. For
more detail, see 2.7.3, “Monitoring z/OS with Domino statistics” on page 50.

7.2.5 Processing Domino statistics
To process Domino statistics, you can:

- View them in the database.
- Use a tool to analyze the data and produce reports.
- Export the data into another format for further processing.

Exporting the contents of a database
If you wish to combine Domino statistics with SMF data, for example, you will probably need
to extract the Domino statistics into a sequential z/OS file. You can then use currently
available tools to analyze the data.

To export the contents of a database into a sequential file on your workstation:

1. Open the database on the server.
2. Select the view that you want.
3. Select the documents that you want to extract based on the collection time.
4. Select File/Export.
5. Set the file name to a file on your workstation.
6. Save as type Structured Text.
7. Click the Export button.
8. Choose:
   - Selected documents
   - “Char code 12” as the interdocument delimiter
   - Wrap words at 75 characters per line
9. Then click OK.

The file will be written on your workstation. You can then upload it to z/OS and input it into
your analysis program.
This chapter provides capacity planning guidelines in the areas of:

- Processor capacity
- Processor storage
- The use of Domino partitioned servers (DPARs)
- The use of zSeries logical partitioning (LPARs)
- DASD space and I/O rates
- Network capacity

We also provide information on a large IBM production Domino server on z/OS, and show how you can get assistance with your capacity planning.

As usual, we begin with an overview of our recommendations, then go into detail about the capacity planning issues.
8.1 Recommendations

These are our recommendations for capacity planning:

- Estimate carefully.
  
  Initial estimates of the resources required to support a Domino server are based on estimates of various factors, including:
  
  - Number of registered users
  - Number of connected users
  - Transaction rate generated by the users
  - Average CPU time used per transaction
  - Amount of data you will have, including the amount of data an average user will have in their mail file
  
  The more accurately you estimate these values, the more accurate your estimate of resource needs will be.

- Make use of the capacity planning assistance available.
  
  See 8.8, “Capacity planning assistance” on page 158.

- Base your capacity planning on your own measurements as soon as possible.
  
  Every Domino environment is different. Therefore, you should monitor your system as soon as you have a significant Domino workload, and then revise your plans based on that. If you have other Domino servers, you can collect information on your workload profile from them and use that to help validate your assumptions.

- If you will have more than 5,000 Notes client or POP3 registered users, 700 IMAP registered users, or 2,500 iNotes client registered users, consider using Domino server partitioning. See 8.4, “The use of Domino partitioned servers” on page 155.

- Remember that these guidelines may change.
  
  These guidelines were put together in November 1999 for Domino Release 5.0, OS/390 Version 2 Release 6, and 9672 generation 6 processors, and modified in November 2000 to reflect the changes that have taken place during that time. We have seen a significant increase in capacity with Domino 6 for z/OS, and with the reduction in the number of DPARs made possible with Domino 6.

- Beware of running your own performance tests and misinterpreting the results.
  
  It is possible to set up some seemingly simple performance tests. However, we have seen very misleading results when trying to use the run time of a single Domino function to estimate the performance of a production workload. Unless you understand exactly which system resources the function uses, and can accurately relate that to the resources your production workload will use, you can get erroneous results. In addition, unless you understand how Domino performs the functions, it is possible that you are measuring something quite different from what you think. You could, for example, be measuring network performance instead of server performance.

  Realistic tests are, unfortunately, complex and expensive to run. Use the information in this document (and the other information sources listed) to get capacity estimates for production workloads.
8.2 Estimating processor capacity

The processor capacity required for Domino depends on:

- The number of users
- The number of active users
- The type of users (Notes, iNotes, POP3)
- The number and type of server transactions generated by the active users
- The CPU time used per transaction
- The CPU required for background tasks and non-user-driven functions
- The configuration of DPARs and LPARs used to support the number of users
- The number of HTTP hits (reads, writes)
- The size of the database and the number of entries in the views used

In addition, many other factors affect performance. These include:

- Frequency of database searches
- Amount of full text indexing of databases
- Agents
- Replication and local replicas
- Clustering
- Type-ahead name searches

8.2.1 Factors influencing the processor capacity needed

The number of registered users, how active the users are, transaction rate per active user, and CPU time used per transaction (all discussed in this section), can be used as the basis for estimating part of the processor capacity needed. The calculation is the number of active users times the transaction rate per active user per hour times the CPU time used per transaction. This is the CPU consumed per hour on a specific processor. The percentage utilization of the processor is: (CPU seconds used per hour / number of CPs / 3600) x 100%.

A major component of the CPU consumed by a DPAR is not dependent on the active users on the DPAR. There are background tasks which are not related to the number of active users, such as replication and administration processes. For a low number of active users, these background tasks account for most of the CPU used by the DPAR. It is important to understand this low utilization effect. If you take the CPU used by a DPAR with 50 users, then 500 users will not use 10 times the CPU.

If you do not currently have Domino on a zSeries platform, begin with 8.2.9, “Making an initial estimate” on page 150.

8.2.2 Which user measurement?

There are a number of user measurements supplied by the Domino server. There are registered users, connected users, instantaneous active users, active 1 minute users, active 3 minute users, active 5 minute users, active 15 minute users, and active 30 minute users.

It is important to know which measure of users you are talking about. In Domino 6 for z/OS and the previous Lotus Domino for S/390 levels, we used the active 15 minute users as a measure of the business of a Domino server. As you will see, this more accurately influences the CPU consumed by the server.

Other platforms may use the number of connected (or even registered) users as the measurement. This number of users may be significantly higher than the active 15 minute user count. You should always make sure that you are comparing like for like.
The following topics include a description of the different user counts, as well as some CPU metrics that are useful in understanding your server.

8.2.3 Registered users

You can get the number of registered users on your server relatively easily for mail applications; it is the number of users whose mail files are located on that specific server. For a Notes application, it is the number of users authorized for each database. If you allow anonymous browser access, you will need to estimate how many users might be logged on at one time.

The number of registered users is not reported in the SMF record type 108. However, the number of registered users for a server is usually known by the Domino administrator.

The number of registered users and the average mail database size per user can be used to determine the disk space required for user mail files.

8.2.4 Connected users

Connected users have been active recently, and have not been inactive long enough to have timed-out on the server. The number of connected users is very much influenced by the user time-out parameter `Server_Session_Timeout` in `notes.ini`. For example, changing this value to 59 minutes from the default of 4 hours could show up in statistics as a 50% drop in the number of users supported! While you may have relieved a storage problem and improved performance, the `server.user` stat that is often referred to would indicate your server is now much less efficient, unless you take this setting into account.

If you have a large user time-out period, then you will need more real storage for the information that is retained about connected users. See section 8.3.1 for more information about the amount of storage that your connected users may require.

A suggested `Server_Session_Timeout` value is 59 minutes.

8.2.5 Active users

Active users are those users who have been active in the previous specified period. We recommend that you use the number of active users in the previous 15 minutes. This is the active 15 minute users, SMF108SLUA15M in the Server Load section of the SMF 108 record.

The active user rate is important because it influences the CPU and processor storage required for the server.

The active 15 minute users divided by the number of registered users gives a measure of how often your users use the server.

8.2.6 Domino transaction rate

A Domino transaction is not a user interaction. It is a Domino call to a high level function, such as opening a database. Thus, one user interaction will involve several Domino transactions. This has the advantage of making a transaction more consistent, because a complicated user interaction will involve more transactions.

The rate at which transactions are processed by the Domino server depends on the number of active users and the rate at which they are working. For example, if an installation with 1,000 registered users has an average 200 active users during the hour, each working at a
rate of 90 server transactions per hour, then the server would need to process 18,000 transactions per hour.

You can easily find out the transaction rate. The number of transactions run in an interval is given in the SMF108 record field SMF108SLTRANS in the Server Load section. To get the one hour total, add the numbers for the intervals within the hour.

Note that the transaction rate per user could be higher during a pilot when users are “experimenting” with this new tool. Alternatively, the transaction rate per user could be lower until you have a critical mass of users on the server.

The number of transactions divide by the number of active 15 minute users gives a measure of how busy your users are when they are using the server.

### 8.2.7 CPU time used per transaction

There are many variables that affect the CPU time used by a server transaction. The size of the address book used for address resolution, the size of the items being sent, the number of items in the user's mail file, and the size of items in the user's mail file are just a few of the variables that affect the CPU time per transaction.

You can measure the CPU time used per transaction on your system. For the time interval you select, you can get the total CPU time used by a Domino DPAR by adding up the CPU time used by each of the DPAR's address spaces. You can get that information from the SMF records (see 2.7.2, “Monitoring z/OS with SMF” on page 47) or from the RMF Workload report for the DPAR's Service Class.

You should include a proportion of the CPU time used by z/OS, TCP/IP, system monitors, and other system address spaces, depending on the workload mix on the system. Divide the CPU time used by the number of Domino transactions recorded for the same interval, as described previously.

If Domino is the only workload running in the LPAR, then divide the CPU time used by the LPAR for a given time interval by the number of transactions recorded.

Beware of the low utilization effects with a small number of users. This is discussed in section 8.2.1. Similarly, during migration of the users, the CPU use will be higher, particularly when the users first open their mail databases or when you are still replicating the mail database from the old server.

### 8.2.8 Consider the peak times

There are times during the day when the capacity needed by Domino is higher than the average and there are times when it is lower.

If Domino will represent less than 50% of the total workload on the z/OS server, and where the other workloads are of lower priority, you may be able to use the average workload for planning. The z/OS WLM will manage the overall resources and make them available to Domino at the peak times if you define the Domino workload to have a high priority.

For dedicated Domino servers, or for z/OS systems where Domino is not the highest priority workload, a peak load factor should be considered to ensure consistently good response times.

Determining the peak load factor to use requires some understanding of the work schedule of the users. If the users are teachers who can only use the system between classes, or at the
beginning or end of the day, then the peak load factor could be much higher than two times the average. If the majority of the users are using a local copy of their mail database on their workstation, then the peak load factor could be much less than two times the average.

Also, if your users span multiple time zones, then you will have lower peaks but during a wider window. Yearly and business trends will also affect your measurement number. For example, we have seen an increase of 30% in the number users and transactions from August to September on production servers as people return from summer vacations.

The use of a local copy by a high percentage of users results in a more uniform workload throughout the business day, but could also extend it. Users may start their replication earlier so that it finishes before they arrive at work.

If Domino will share a z/OS server with other high priority workloads, consider the Domino peak in relation to the other workloads. If the users of your Domino system are also the users of your CICS or IMS system, they are unlikely to be using both systems heavily at the same time of day. Since the peaks of the two workloads occur at different times of the day, the total capacity you need is less than the sum of the peaks for the two workloads. By recognizing this, many S/390 installations will require less capacity than if they ran each workload on a separate server.

If you are migrating the Domino users from other platforms, you may be able to determine the peak times and peak-to-average ratios from those.

8.2.9 Making an initial estimate

See 8.8.1, “TechXpress” on page 158 for information on how to request an initial sizing for a planned Domino 6 for z/OS installation.

8.2.10 Capacity planning

Capacity planning is not optional. You need to periodically measure the resources that your Domino installation is using and adjust as needed. Whether or not you do capacity planning at the transaction level is up to you.

8.3 Estimating processor storage

The measured and estimated storage in the following sections is the total amount of storage, in megabytes, needed on the S/390 to support Domino. It includes the storage required for z/OS and other prerequisite software as well as for Domino. This storage is the sum of processor storage and paging storage. However, paging to auxiliary storage should be avoided since it will increase CPU usage as well as response time.

The amount of storage needed also depends on the number of physical threads acquired by a DPAR, as specified in the SERVER_POOL_TASKS parameter in the notes.ini file, and the size of the DFSMS buffer pool as specified in the FILESYSTYPE TYPE statement in SYS1.PARMLIB(BPXPRMxx).

The amount of storage needed depends more specifically on the value of certain Domino and z/OS parameters that have been described in other sections of this book. These include:

- The size of the NSF buffer pool (see section 5.4)
- The number of threads in the threadpool (see section 5.3.4)
- The size of any zFS or HFS cache or buffer pool settings (see chapter 3)
The number and type of add-in servers and companion products that are running (such as virus checkers)

The following sections describe how to estimate the storage needed by Domino for the following types of clients:

- Notes
- iNotes
- POP3
- IMAP

### 8.3.1 Estimating processor storage for Notes clients

Figure 8-1 on page 152 shows the amount of processor storage that is needed to support a Domino 6 server on z/OS V1R2, based on the number of connected Notes client users. The following storage equation yields a good estimate of processor storage:

\[
\text{Total storage per LPAR (MB)} = 950 + n(400 + 0.135 \times \text{CU})
\]

where \( n \) is the number of Domino partitions (DPARs), and \( \text{CU} \) is the number of connected users per DPAR.

**Attention:** This formula is intended to recommend enough storage to start a server and router in a pilot environment. As you add functions and users to your server, your storage requirements can increase to two or three times that amount.

Note that this equation yields a central storage requirement that is about 20% higher for a Domino 6 server than for an R5 server under the same user load. If you are migrating from an R5 Domino server environment where central storage is in short supply, you should consider adding more storage in order to prevent paging in Domino 6.

These results were generated with the new R5 Notes client benchmark workload. The important storage-related parameters of this environment include:

- NSF_BUFFER_POOLSIZE_MB = 256
- DFSMS 1.6 virtual storage = 256 MB
- DFSMS 1.6 fixed storage = 256 MB

Also note that if you use the previous equation to estimate your storage requirements, you arrive at a minimum value of 1350 MB. Our lab measurements have demonstrated that the server can actually be started on a system with less than 512 MB of processor storage. However, by the time 1000 users are simulated against the server, the processor storage requirements closely match those predicted by the equation. This is displayed graphically in Figure 8-1.
If you intend to experiment with a Domino 6 server or run a server with a very light user load, you may be able to run with less than the amount of storage predicted by the equation. However, you should closely monitor RMF to be sure that paging is not an issue. When you are ready to run in production mode, you should use the equation to calculate an appropriate storage value, and use that value. In both cases, you should closely monitor your storage usage to be sure the system isn’t constrained. Usage will change as you use other features of the Domino 6 server.

8.3.2 Estimating processor storage for POP3 clients

Table 8-1 shows the amount of processor storage that is needed to support POP3 mail users on a single Domino for z/OS server, based on the number of active users. The “Measured Storage” is the actual storage that was used in the series of tests show in the table. The “Estimated Storage” was calculated using the following formula:

$$\text{Total Storage per LPAR (MB)} = 700 + 0.024 \times \text{APU}$$

where APU is the maximum number of POP3 users that are active during a 15-minute interval.

**Attention:** This formula is intended to recommend enough storage to start a server and router in a pilot environment. As you add functions and users to your server, your storage requirements can increase to two or three times that amount.

<table>
<thead>
<tr>
<th>Active Users</th>
<th>Measured Storage (MB)</th>
<th>Estimated Storage (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>702</td>
<td>719</td>
</tr>
<tr>
<td>1600</td>
<td>747</td>
<td>738</td>
</tr>
<tr>
<td>3200</td>
<td>774</td>
<td>777</td>
</tr>
<tr>
<td>4800</td>
<td>804</td>
<td>815</td>
</tr>
</tbody>
</table>
Table 8-1 notes:
1. The measured and estimated storage is the total amount of storage needed, in megabytes, and includes z/OS and other prerequisite software.
2. NSF buffer pool size: 256 MB
3. DFSMS VIRTUAL: 100 MB
4. DFSMS FIXED: 0 MB
   **Note:** We recommend setting the same values for DFSMS virtual and fixed.
5. All the test runs were with one DPAR. The storage estimates reflect the use of one DPAR.

8.3.3 Estimating storage for IMAP clients

Figure 8-2 shows the amount of processor storage that is needed to support a Domino 6 server on z/OS V1R2, based on the number of active IMAP users. The following storage equation yields a good estimate of processor storage:

$$\text{Total storage per LPAR (MB)} = 1900 + (0.032 \times \text{AU})$$

where AU is the expected number of active users, which is the number of users who you expect to generate a “hit” against the server in a 15 minute interval. The term “connected user” is not appropriate in this context, since IMAP users don't create a server connection.

**Attention:** This formula is intended to recommend enough storage to start a server and router in a pilot environment. As you add functions and users to your server, your storage requirements can increase to two or three times that amount.

Note that IMAP has been substantially re-written for Domino R6 to enhance performance. The most important of these changes from a scale standpoint is the implementation of a threadpool, similar to what exists today for Notes clients. In addition to allowing for substantial increases in the number of users that can be supported by a single server, the storage usage characteristics are very different. IMAP under Domino 6 requires more storage up front to get the server running, but the per-user cost is much lower. If you are migrating from an R5 IMAP environment, be sure to re-evaluate your storage needs according to the previous equation.

These results were generated with the new Domino 6 iNotes benchmark workload, which is considerably heavier and more production-like than the R5 version of this benchmark which has been used for previous capacity planning efforts. The important storage-related parameters of this environment include:

- NSF_BUFFER_POOLSIZE_MB = 512
- DFSMS 1.6 virtual storage = 256 MB
- DFSMS 1.6 fixed storage = 256 MB
If you intend to experiment with a Domino 6 server or run a server with a very light user load, you may be able to run with less than the amount of storage predicted by the equation. However, you should closely monitor RMF to be sure that paging is not an issue. When you are ready to run in production mode, you should use the equation to calculate an appropriate storage value, and use that value. In all situations, you should closely monitor your storage usage to be sure the system isn’t constrained. Usage will change as you use other features of the Domino 6 server.

### 8.3.4 Estimating storage for iNotes Web Access clients

Figure 8-3 on page 155 shows the amount of processor storage that is needed to support a Domino 6 server on z/OS V1R2, based on the number of active iNotes Web Access users. The following storage equation yields a good estimate of processor storage:

Total storage per LPAR (MB) = 2000 + (0.0.099 AU)

where AU is the expected number of active users, which is the number of users who you expect to generate a “hit” against the server in a 15 minute interval. The term “connected user” is not appropriate in this context, since iNotes Web Access users don’t create a server connection.

**Attention:** This formula is intended to recommend enough storage to start a server and router in a pilot environment. As you add functions and users to your server, your storage requirements can increase to two or three times that amount.

Note that this equation yields a central storage requirement that is about 25% higher for a Domino 6 server than for an R5 server under the same user load. If you are migrating from an R5 Domino server environment where central storage is in short supply, you should consider adding more to prevent paging in Domino 6.

These results were generated with the new Domino 6 iNotes benchmark workload, which is considerably heavier, and more production-like than the R5 version of this benchmark, which...
has been used for previous capacity planning efforts. The important storage-related parameters of this environment include:

- NSF_BUFFER_POOLSIZE_MB = 512
- DFSMS 1.6 virtual storage = 256 MB
- DFSMS 1.6 fixed storage = 256 MB

Also note that if you use the previous equation above to estimate your storage requirements, you arrive at a minimum value of roughly 2 GB. Our lab measurements have demonstrated that the server, and associated Domino HTTP server addin, can actually be started on a system with about 1.5 GB of processor storage. However, storage requirements quickly rise above 2 GB as users begin to use the server. This is displayed graphically in Figure 8-4. The test measured up to 4000 users in a single partition.

![Central Storage Requirements](image)

*Figure 8-3  Processor storage requirements for iNotes clients*

If you intend to experiment with a Domino 6 server or run a server with a very light user load, you may be able to run with less than the amount of storage predicted by the equation. However, you should closely monitor RMF to be sure that paging is not an issue. When you are ready to run in production mode, you should use the equation to calculate an appropriate storage value, and use that value. In both cases, you should closely monitor your storage usage to be sure the system isn't constrained. Usage will change as you use other features of the Domino 6 server.

### 8.4 The use of Domino partitioned servers

With multiple Domino partition environments, it is important to carefully plan your real storage usage. To demonstrate the implications for storage usage that different Domino partition deployments drive, we performed an experiment in the lab with constant workload and varying numbers of Domino partitions. Our workload consisted of an identical 8,400 user Webmail workload that was supported using 4, 8, and 12 Domino partitions. Although our particular results didn't show any CPU differences, the real storage requirements are obvious in Figure 8-4 on page 156.
It is important to note that, while this particular lab measurement did not show any difference in CPU usage, normally CPU overhead in a live production environment will be higher with larger numbers of Domino partitions, even under similar workloads. Our lab results here did not include many of the processes that are present in a production environment, such as agent processing and virus checking.

If you plan to have more than 5,000 registered Notes client users on your Domino for z/OS system, you will need to run multiple Domino servers to support them.

We have observed no degradation in performance with up to 10,000 connected Notes client users in a single Domino partition (DPAR) running in a controlled environment with adequate resources. Going beyond 10,000 connected users results in the inability of new users to establish a connection to the server.

The largest Domino for z/OS DPAR in production as of December 2002 has over 5,000 registered users. However, we suggest that you monitor your system as it grows beyond 2,000 registered users to determine the maximum number of active users per DPAR. A reasonable target is 2,000 active users per DPAR for Notes clients.

### 8.5 The use of zSeries LPARs

The number of Domino partitions that can run on a given LPAR depends primarily on two factors: the amount of CPU and the amount of storage available. From a CPU standpoint, LPARs can generally be made as large as necessary, limited only by the amount of CPU that the system architect is willing to give to the LPAR. The story is different for storage, though. S/390 and zSeries machines in 31-bit mode allow for 2 GB of central storage and 2 GB more of expanded storage, giving a total of 4 GB of real storage that can be used by an LPAR. This is most often the factor that limits the number of Domino partitions that can run on one LPAR.
Our experience has shown that two Domino partitions heavily loaded with Notes client users will allocate roughly 2 GB of virtual storage, plus some additional overhead for other processes associated with the server. However, the working set of this virtual storage that actually needs to be backed by central storage at any instant in time is substantially smaller than this. One LPAR can effectively manage the virtual storage needs of two Domino partitions, given 2 GB each of central and expanded real storage.

You can run up to 15 z/OS systems on a single zSeries server using logical partitioning.

For more information on the cost of a DPAR and the measurement results, see 6.1.2, “Costs of partitioning” on page 115. For information on monitoring Domino using statistics, see 2.7.3, “Monitoring z/OS with Domino statistics” on page 50. You can track the storage use on your system using the high UIC value. See Figure 2-9 on page 45.

### 8.6 Estimating DASD required

Configuring the DASD I/O subsystem required to support your Domino servers requires that you estimate the amount of space you will need and the I/O rate you will need to sustain. The following sections provide some guidance in estimating your DASD space and I/O rate requirements. As with any system, you should measure the DASD space used and I/O rates on your system periodically and base your follow-on estimates on these measurements.

#### 8.6.1 Estimating DASD space

The amount of DASD space that is needed for Domino depends on the amount of data that you will store. It is the same as for other Domino platforms.

The amount of DASD space required for mail databases depends on the amount of data that each user will store. This is normally in the range of 40 to 100 megabytes per user, and will typically grow as the users become more advanced in their use of Domino. Installations with thousands of users will therefore require hundreds of gigabytes (GB), if not terabytes, of DASD space. Add to this 17 GB for each DPAR and 15 GB for each LPAR.

z/OS provides a standard UNIX file system for applications such as Domino that use z/OS UNIX. You should plan your use of the file system carefully to avoid the need to move data around later. See the discussion about Notes directory structure in *Lotus Domino for S/390 Release 5: Installation, Customization and Administration*, SG24-2083 for more information.

#### 8.6.2 Estimating DASD I/O rates

The DASD I/O rate for Domino will vary by client type, number of connected users, and the amount of work being done by the users. We recommend that you use the following for your initial estimate:

- 0.5 to 1 I/O per second per active user

These results are from our laboratory measurements; the rate may vary in your environment.

### 8.7 Network considerations

Proper planning of the network will minimize delays due to an overloaded network. As users begin to exploit the advanced functions of Lotus Notes (such as attaching documents, spreadsheets, and presentations to notes), the amount of data flowing through the network will grow.
One difference with a large Domino server, such as z/OS, is that you will have a lot of network traffic into one point. zSeries users already understand how to do that, but it is different from an implementation where there are many small servers. Make sure that you configure at least two paths into the zSeries server, for both performance and availability.

A large server also results in less network traffic overall. It eliminates the server-to-server network traffic for mail routing, database replication, and calendar tasks.

We recommend that you use the following for your initial estimate:

3 to 4 Kbps of bandwidth per active user

These results are from our laboratory measurements; the rate may vary in your environment.

For more information on designing Domino networks, we recommend reading *Lotus Notes and Domino Scalable Network Design* by John Lamb and Peter Lew.

### 8.8 Capacity planning assistance

An initial capacity planning estimate for a Domino workload is available through your IBM or Lotus marketing representative. A more detailed analysis is available through various chargeable IBM offerings.

#### 8.8.1 TechXpress

Your IBM or Lotus marketing representative can request an initial sizing for Domino on z/OS. They can request a form from TechXPress, complete the form with your help, and submit it. TechXPress will route it to the department that does the sizings for Domino on z/OS.

The following information is needed to complete this form:

- Estimate the peak number of registered, connected, and active users that you expect Domino to support.
  - The number of registered users is the total number of users that you plan to authorize to use the servers that you are requesting the sizing for.
  - The number of connected users is the maximum number of users that you expect to be in session with Domino at one time. A reasonable starting point is 70 percent of the total number of registered users. This is used to estimate the amount of processor storage that will be needed for your servers.
  - The number of active users is the maximum number of users that you expect to be doing productive work during a 15-minute interval. A reasonable starting point is somewhere in the range of 20 to 40 percent of the number of registered users. This is used to estimate the size of the processor that will be needed.

- State the percent of active users that are Notes, POP3, IMAP, or iNotes clients.
  - For Notes and iNotes clients, state the percentage that are doing light mail, experienced mail, experienced mail plus calendaring/scheduling, experienced mail plus calendaring/scheduling and applications. If applications are planned, then an estimate of the application workload, as a percent of the mail workload, is requested.
  - For POP3 and IMAP clients, state the percentage that are doing light mail and the percent that are doing experienced mail.

- Tell us the processors that you would like sized, if any; for example, an existing zSeries that has excess capacity.
Upon receipt of this form, TechXPress will provide you with an estimate of the processor size you will need to support this workload. IBM and Lotus employees can submit a TechXpress request on the IBM intranet or by phone. Find out more at this Web site:

http://w3-1.ibm.com/support/techxpress.html

8.8.2 SNAP/SHOT®

Host SNAP/SHOT Services System Capacity Planning supports many applications, including Domino for z/OS. SNAP/SHOT can show you the effect of adding Domino to your existing zSeries workload. It can help you determine what changes, if any, should be made to your zSeries and network configuration to accommodate this additional workload. For more information, visit the Web site:


You can also call the following toll-free numbers:

Telephone (USA) 1-888-474-TEST
Telephone (outside) (301) 240-8535

8.8.3 IBM Testing Services

IBM Testing Services provides the hardware, software, drivers, network, and expertise to perform comprehensive stress testing. Whether for proof-of-concept testing, capacity and performance testing, or full stress testing, there are times when only a "real" test will do. For more information, visit the Web site:

http://www.as.ibm.com/asus/svs.html

You can also call the following toll-free numbers:

Telephone (USA) 1-888-474-TEST
Telephone (outside) (301) 240-8535

8.9 Performance management versus capacity planning

It is important for you to understand the different reporting requirements for capacity planning versus performance management.

- For capacity planning, you need to look at the big picture over a long period of time. You are not looking for problems, but to see how the system is growing and what changes are occurring in the way the system is used.

- With performance management, you are looking at current problems and also trying to prevent problems. For this you need to look at much more detail for a short period of time.

The next section suggests the kind of reporting you may wish to implement. However, it is important to keep the raw SMF data for several months in order to investigate previous periods in more detail if required.

Typically, rates (I/O and CPU) are capacity indicators, while response times are performance indicators. We have seen cases where more users were added to Domino (on multiple platforms) yet the CPU used by the server actually dropped. This is not because Domino suddenly became more efficient, but rather an existing bottleneck caused more work to be queued and less work to actually flow through the system. The NRPC (Notes client) and end-user response times increased significantly when these users were added.
Also, having two identical I/O devices with different I/O rates does not tell you which one is a performance problem. The device with the lower rate could have significantly higher response times (due to fragmentation, disk cache problems, channel issues, and so forth, that are causing server problems), in contrast to the device with a higher rate but lower response times. Also, flat line rates (of CPU and I/O) below the maximum typically indicate a bottleneck in the system. A server on a processor that is at 50% may not be able to get any more work or use any more CPU due to a I/O or network problem.

8.9.1 What is required for capacity planning

It is important that you look at data going back over a long time period, certainly many months if not years. You are looking for trends in the data over time in order to see if increases are a short term problem or an ongoing feature.

Because of the long period under consideration, you need to report on daily totals or averages. You probably need to look at the 24 hour picture as well as the prime shift (usually the online day, for example 09:00 to 17:00). Usually the resource peaks are in the online day and need to be tracked.

It is preferable to look at each DPAR separately, as well as each LPAR and the whole processor.

Examples of information you may want to trend are:

- Average CPU% for the processor
- Average CPU% for each LPAR
- Each DPAR’s average CPU%
- Each DPAR’s average active 15 minute users in prime shift
  - You need to also track the number of registered users in the DPAR (which is not available in the SMF records) to see if the existing users spend more time in Domino.
- Each DPAR’s average transaction rate
- Each DPAR’s average transaction per active 15 minute user
- Each DPAR’s average CPU% per active 15 minute user
- Each DPAR’s average CPU% per transaction

You may also wish to track the split of CPU across the DPAR address spaces.

When calculating the averages you can use the hourly average or the interval average. If you use the interval, do not change your reporting interval from, say, 15 minute to 10 minutes, since this will make it difficult to trend across the change. It is preferable if you do hourly averages.

8.9.2 What is required for performance management

A key recommendation is that for successful Domino problem and performance management the z/OS people and the Domino administrators must work closely together. An observation made by one group may be easily explained by the other group.

For performance management you will be looking at data covering a much shorter period, probably about two weeks. You need to have sufficient time to understand what is the “normal” profile, as well as data for the time of the potential problem. If you do not know what is normal then you cannot spot what is wrong when there is a problem.
For performance management you need to report at a much more granular level. Hourly intervals are almost certainly too long because some Domino functions are activated on the hour every hour and their impact will be lost. Instead, you should be reporting by SMF intervals. Either 10 or 15 minutes is a good interval; 30 minutes is probably too long.

You may want to report on the full day’s data and also the prime shift, as was recommended for capacity planning.

Until you have all your users on the Domino server, and have been running for some time with this user population, you will need to monitor the performance of the Domino servers regularly. During periods of change, particularly when starting a new DPAR, it is recommended that you look at performance daily. When you have a stable environment you may wish to extend to reviews once or twice a week.

The best way to present the data is to produce graphs with each point being one 15 minute interval for a period of one or two weeks.

Examples of information you may want to monitor are:

- CPU% for the processor
  To see whether the processor is running at or near capacity, which may constrain the Domino LPARs.

- CPU% for the LPAR
  If the LPAR is near 100% utilization at any time, you may need to add more logical CPs. With high utilization you should also run some RMF Partition Data reports to see if the weights need adjusting.

- LPAR Share Utilization
  While the LPAR may not be at 100% of its capacity, it may be at its defined share. In a fully loaded box, this will mean that there are no more cycles to give the LPAR. Also, an LPAR that is over its share will be forced to give back its extra cycles if the box becomes fully utilized.

- Paging to Auxiliary
  The LPAR should not be paging from auxiliary page datasets. If there is paging, then the Domino server will not perform well.
  In 31-bit real storage, you should also monitor the movement of pages to and from expanded storage.

- Total CPU% for each DPAR
  This will show the peaks and troughs for this DPAR. You should track your busy days and periods to understand if the use of the DPAR is changing.

- Split of CPU between address spaces for each DPAR
  You should accumulate the CPU for address spaces performing the same function; for example, multiple Replicator address spaces should be added together.
  To make this easier, define WLM Report Classes for each function within each DPAR. For example, address space MAIL1REx should be in one report class and MAIL2REx in another.
  With the CPU for each Domino function separated, you can see unusual activity for a function, such as a spike at certain times of day or long periods of high CPU where it is not expected, for example, spikes in Replication, Agent Manager, or Administration. An example of this is shown in Figure 8-5 on page 164.
- Active 15 minute users for each DPAR
  This can be compared to the number of registered users to understand your user profile and if this has changed for some reason.

- Transaction rate for each DPAR
  This is an indication of the load on the server.

- Transactions per active 15 minute user for each DPAR
  This shows how heavily an active user is driving the system. If the users are new to Domino then this may increase as they learn to use the functions available.

- CPU% per active 15 minute user for each DPAR
  This is the CPU cost per active user, which should remain fairly consistent. As you load users onto the server this number should come down, as the user population increases, because of the low utilization effect.

- CPU% per transaction for each DPAR
  As with the CPU per active 15 minute user, this should be consistent once all the users are on the DPAR.

- Physical thread use for each DPAR
  Domino Notes uses the physical thread pool to dispatch virtual threads to perform work. The number of physical threads can be a good indicator of problems elsewhere within the server. A short duration increase in the number of physical threads used can indicate that there are delays in this Domino server at that time. You may reach the maximum number of threads for a short period of time, and increasing the maximum number of physical threads allowed may make the problem worse. The cause of the delays should be investigated first. By establishing a baseline of what constitutes a normal number of active users per physical thread, you can quickly determine if you need more threads or have a bottleneck.

  If you are consistently running for long periods of time near the maximum number of threads and there are no problems in the server, you may wish to increase the maximum number of physical threads allowed.

Other things that will need monitoring that are not easy to produce on a graph are:

- Latch contention
  **Note:** This is for DFSMS 1.5 HFS users. It is not applicable to a full zFS implementation.
  A file request can be queued within DFSMS 1.5 HFS waiting for access to the file. If all the requests for a file are reads then they can be performed concurrently. If one request is for a write, it will have exclusive control of the file until completed and all other requests, whether reads or writes, must wait for it to complete. This waiting is not measured in any RMF or Domino SMF record.

  The way to see if there are any latch contentions is to automatically schedule the issuing of the D GRS,C every few minutes. This will write a message showing where there are any delays, the holder of the resource, and a list of the requests waiting for the resource.

  You need to periodically check that you are not getting latch contention. An isolated latch contention may not cause problems, but any buildup of queues should be avoided. This is particularly true in the case of key databases such as the Name and Address book.

- DBCACHE Rejects
  DBCACHE rejects can lead to Domino rebuilding database information unnecessarily. When it attempts to cache the database information and there is no room then it is thrown away and this is called a reject. The DBCACHE is not managed on a least recently used
basis. When the database is next required, Domino must rebuild the information instead of just reading it from the DBCACHE.

Rejects overnight may not be a problem, but you should avoid rejects during the online day.

► DASD response times

Monitor the Domino DASD volumes periodically for poor response times.

You should also track the disconnect times for file systems (both zFS and DFSMS) to see if they are increasing. This may be an indicator that the dataset needs reorganizing. See 3.5.6, “File system reorganizations” on page 66 for how to do this.

► HFS dataset use

You need to track when an HFS dataset passes 85% full.

DFSMS performance degrades above this utilization, so you need to detect when this happens. See 3.5.3, “File system space monitoring” on page 64 for how to do this.

For zFS, there is no automatic warning given, so you will need to report regularly on how full the zFS aggregates have become.

An example of the kind of graphs you may wish to produce is shown in Figure 8-5 on page 164. This is a Domino mail DPAR with data for two day’s prime shifts with each vertical column being 15 minutes. This graph highlights a problem with the DPAR. At the same time each day (late morning), the Agent Manager address space (AM in the graph) uses a large amount of CPU. Investigation reveals a problem with the scheduling of the agent.

You can also see that the Server address space (SE in the graph) is consistently the largest user of CPU. This should be expected in a Domino mail server, because the Server address space is where the user processing is performed. For a Web-based Domino server, the HTTP address spaces would be the largest.

The graph is much clearer to read in color because of the large number of different Domino functions represented.
Figure 8-5  Sample of CPU use by a DPAR
Workload Manager (WLM) example

This appendix contains an example of the Workload Manager goal mode definitions for a Lotus Domino production environment. There are six Domino servers: ML96NP, ML97NP, ML98NP, ML99NP, ML90NP and DB99NP. These servers run in three LPARs of a Sysplex and are the prime workloads for these LPARs.

Use the ISPF administrative application to enter or change your WLM goal mode definitions.

Workload and service class descriptions

Table A-1 shows the workloads that the Workload Manager will use. A workload is a named collection of work to be reported as a unit.

Table A-1 Workload descriptions

<table>
<thead>
<tr>
<th>Workload</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMEAPP</td>
<td>APPC workload</td>
</tr>
<tr>
<td>PRIMEBAT</td>
<td>Batch workload</td>
</tr>
<tr>
<td>PRIMESTC</td>
<td>STC workload</td>
</tr>
<tr>
<td>PRIMETSO</td>
<td>TSO workload</td>
</tr>
<tr>
<td>PRIMOMVS</td>
<td>UNIX System Services workload</td>
</tr>
</tbody>
</table>

Table A-2 on page 166 shows the service classes that the different workloads will be assigned to. A service class is a named group of work within a workload with similar performance characteristics. For Domino, we specified two service classes called NOTESLOW and NOTESSRV. The default service class for Domino is NOTESSRV. Service class NOTESLOW is not used for normal operation.
Table A-2  Service class descriptions

<table>
<thead>
<tr>
<th>Service class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCHLOW</td>
<td>Low priority batch</td>
</tr>
<tr>
<td>HOTTSO</td>
<td>Support TSO users</td>
</tr>
<tr>
<td>NOTESLOW</td>
<td>Lotus Domino Server ResetToClass</td>
</tr>
<tr>
<td>ML96NP</td>
<td>Lotus Domino server ML96NP</td>
</tr>
<tr>
<td>ML97NP</td>
<td>Lotus Domino server ML97NP</td>
</tr>
<tr>
<td>ML98NP</td>
<td>Lotus Domino server ML98NP</td>
</tr>
<tr>
<td>ML99NP</td>
<td>Lotus Domino server ML99NP</td>
</tr>
<tr>
<td>ML90NP</td>
<td>Lotus Domino server DBL90NP</td>
</tr>
<tr>
<td>DB99NP</td>
<td>Lotus Domino server DB99NP</td>
</tr>
<tr>
<td>OMVSLOW</td>
<td>Low priority OMVS Work</td>
</tr>
<tr>
<td>OMVSTASK</td>
<td>OMVS Support Tasks</td>
</tr>
<tr>
<td>OMVSUSER</td>
<td>UNIX System Services users</td>
</tr>
<tr>
<td>STCLOW</td>
<td>Low priority STC</td>
</tr>
<tr>
<td>STCMED</td>
<td>Better performance for ADSM</td>
</tr>
<tr>
<td>TSOPRIME</td>
<td>Normal TSO users</td>
</tr>
</tbody>
</table>

Report classes

Table A-3 shows the report classes set up to provide more granular reporting of the system workloads.

Table A-3  Report classes

<table>
<thead>
<tr>
<th>Workload</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSM</td>
<td>ADSM report class</td>
</tr>
<tr>
<td>DFHSM</td>
<td>DFHSM report class</td>
</tr>
<tr>
<td>JES2</td>
<td>JES2 report class</td>
</tr>
<tr>
<td>MVSNFS</td>
<td>MVSNFS report class</td>
</tr>
<tr>
<td>OMVS</td>
<td>OMVS kernel report class</td>
</tr>
<tr>
<td>RMFIII</td>
<td>RMF Monitor III report class</td>
</tr>
<tr>
<td>SYSBMAS</td>
<td>SYSBMAS report class</td>
</tr>
<tr>
<td>TCPML</td>
<td>TCP/IP report class</td>
</tr>
</tbody>
</table>

Table A-4 on page 167 shows report classes set up to allow reporting for individual Domino server functions. Thus it reports on the address spaces (Server, Router, Replicator, and so forth) separately. The table gives an example for ML96NP. Similar definitions will be needed for each of the other Domino servers. The address spaces running on your servers will depend on the task you specify in notes.ini or start manually. You should use the list in the table as an example of possible address spaces.
Classification rules

Classification rules specify how a workload, such as a Domino address space, is assigned to a service class. In this example the Domino servers have been each assigned their own service class.

The Qualifier type column shows on what input the classes are selected. In our example (Table A-5), Workload Manager assigns service classes and report classes to defined Domino servers based on job name (qualifier type TN, values are in column Qualifier name). There are other workloads, identified by user ID (qualifier type UI).

<table>
<thead>
<tr>
<th>Level</th>
<th>Qualifier type</th>
<th>Qualifier name</th>
<th>Service class</th>
<th>Report class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TN</td>
<td>ML96NP</td>
<td>ML96NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML97NP</td>
<td>ML97NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML98NP</td>
<td>ML97NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML99NP</td>
<td>ML99NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>DB99NP</td>
<td>DB99NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>MVSNFS</td>
<td>MVSNFS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>OMVS</td>
<td>OMVSTASK</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>UI</td>
<td>OMVS</td>
<td>OMVSTASK</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>UI</td>
<td>SERVER</td>
<td>NOTESSRV</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ADSM</td>
<td>OMVSLOW</td>
<td>ADSM</td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ADSM</td>
<td>OMVSLOW</td>
<td>ADSM</td>
</tr>
<tr>
<td>1</td>
<td>UI</td>
<td>OMVSTASK</td>
<td>OMVSLOW</td>
<td>OMVS</td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ADSM</td>
<td>OMVSLOW</td>
<td>OMVS</td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>NOTESSRV</td>
<td>NOTESSRV</td>
<td>NOTESSRV</td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML96NP</td>
<td>ML96NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML97NP</td>
<td>ML97NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML98NP</td>
<td>ML97NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>ML99NP</td>
<td>ML99NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>DB99NP</td>
<td>DB99NP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>MVSNFS</td>
<td>MVSNFS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>OMVS</td>
<td>OMVS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TN</td>
<td>SYSBMAS</td>
<td>SYSBMAS</td>
<td></td>
</tr>
</tbody>
</table>
You will also need to add report class definitions for each of the function in Table A-4 for each of your servers.

### Service definition goals for service classes

Having put workloads in the defined service classes, you now need to define the performance goals required for these service classes. These goals may be velocity (the rate at which you expect work to be processed), response times (a percentage completed within a specified time), or discretionary (WLM-defined goal for workload for which you do not have any specific requirement). You also assign a number to specify how important it is that the workload achieves its goal.

You can also specify the relative importance of a given workload and describe your performance requirements. The Domino service classes have each been given an importance of 1, but you could give some servers a lower importance if required.

z/OS Workload Manager recognizes importance in five levels: 1 to 5, where 1 is the highest importance. The Domino service classes specify velocity 60 as their performance goal. You could give each Domino server a different velocity if required. Velocity is a percentage from 1 to 99 and specifies how fast a given workload should run when ready, without being delayed by waiting for WLM-managed resources. Domino servers should always have a velocity performance goal.

In our example, we specified all Domino server performance goals within one period only. Other workloads have periods specified, because they have variable resource requirements, so their goals change as they use more resources. You should not specify more than one period for a Domino server.
Occasionally you may want to limit resources used by Domino servers because of a more important workload in the system. In that case, you can use the NOTESLOW service class with business importance of 3 and velocity 20. Use service policy overrides to adjust WLM goals temporarily or during certain periods like weekends or holidays. Limiting the resources of a production Domino server is not recommended during normal operation.

<table>
<thead>
<tr>
<th>Service class</th>
<th>Workload</th>
<th>Period</th>
<th>Duration</th>
<th>Importance</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCHLOW</td>
<td>PRIMBAT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOTTSO</td>
<td>PRIMETSO</td>
<td>1</td>
<td>400</td>
<td>1</td>
<td>90% 00:00:00.500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>1400</td>
<td>1</td>
<td>80% 00:00:01.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>40000</td>
<td>1</td>
<td>70% 00:00:03.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
<td></td>
<td>Velocity 30</td>
</tr>
<tr>
<td>NOTESLOW</td>
<td>PRIMOMVS</td>
<td>1</td>
<td></td>
<td>3</td>
<td>Velocity 20</td>
</tr>
<tr>
<td>ML96NP</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 60</td>
</tr>
<tr>
<td>ML97NP</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 60</td>
</tr>
<tr>
<td>ML98NP</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 60</td>
</tr>
<tr>
<td>ML99NP</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 60</td>
</tr>
<tr>
<td>ML90NP</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 60</td>
</tr>
<tr>
<td>DB99NP</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 60</td>
</tr>
<tr>
<td>OMVSLOW</td>
<td>PRIMOMVS</td>
<td>1</td>
<td></td>
<td></td>
<td>Discretionary</td>
</tr>
<tr>
<td>OMVSTASK</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>Velocity 30</td>
</tr>
<tr>
<td>OMVSUSER</td>
<td>PRIMOMVS</td>
<td>1</td>
<td>400</td>
<td>2</td>
<td>80% 00:00:05.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2000</td>
<td>2</td>
<td>Velocity 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>Discretionary</td>
</tr>
<tr>
<td>STCLOW</td>
<td>PRIMETSO</td>
<td>1</td>
<td>3</td>
<td></td>
<td>Velocity 20</td>
</tr>
<tr>
<td>STCMED</td>
<td>PRIMETSO</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Velocity 40</td>
</tr>
<tr>
<td>TSOPRIME</td>
<td>PRIMETSO</td>
<td>1</td>
<td>200</td>
<td>1</td>
<td>90% 00:00:00.500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>400</td>
<td>1</td>
<td>90% 00:00:02.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td></td>
<td>Velocity 30</td>
</tr>
</tbody>
</table>
Using the domps command

The domps command has been modified for Domino 6. It can be used to monitor the performance of Domino interactively.

B.1 Parameters

Its parameters can be displayed by using the help (-h) parameter shown in Figure B-1.

```
    domps -h

    Valid arguments are:
    domps -p   -Detailed process information with no thread data
    domps -t   -Detailed thread information
    domps      -Detailed thread and detailed process information
```

Figure B-1   Parameters for the domps command

B.2 Displaying virtual storage

The process parameter, domps -p, shows virtual storage usage for Domino servers. For example, Figure B-2 shows server D01MC96 using just over 1 Gigabyte of virtual storage.

```
<table>
<thead>
<tr>
<th>PID</th>
<th>PPID</th>
<th>ASID</th>
<th>THCNT</th>
<th>VIRTUAL</th>
<th>TIME ELAPSED</th>
<th>JOBNAME</th>
<th>RUSER COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>67108934</td>
<td>16777291</td>
<td>5a</td>
<td>155</td>
<td>1018880</td>
<td>3h23</td>
<td>ML96SER</td>
<td>ML96NP server</td>
</tr>
<tr>
<td>83886176</td>
<td>16777306</td>
<td>6e</td>
<td>159</td>
<td>513024</td>
<td>1h19</td>
<td>ML83SER</td>
<td>ML83NP server</td>
</tr>
<tr>
<td>104</td>
<td>83866176</td>
<td>7b</td>
<td>45</td>
<td>453508</td>
<td>10:03</td>
<td>ML83R01</td>
<td>ML83NP router</td>
</tr>
</tbody>
</table>
```

Figure B-2  Using the -p parameter to display virtual storage
B.3 Output examples

We have included examples of the output you get when using domps with its -t and -p parameters and with no parameters.

B.3.1 Sample output for domps

Example B-1 shows the output from issuing the domps command with no parameters. This gives detailed information about threads and processes.

Example: B-1  Output from domps command

<table>
<thead>
<tr>
<th>PID</th>
<th>ASID</th>
<th>STID</th>
<th>TCBADDR</th>
<th>WTIME</th>
<th>SNUM</th>
<th>SVAL</th>
<th>LASTSYSC</th>
<th>SYSC</th>
<th>TIME</th>
<th>S</th>
<th>JOBNAME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>33554519</td>
<td>89</td>
<td>73</td>
<td>484</td>
<td>0:00</td>
<td></td>
<td></td>
<td></td>
<td>ML83</td>
<td>10:54:00</td>
<td>10:54:00</td>
<td>ML83NP</td>
<td>sh -c</td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/bin/tools/startup.sh 'ML83' '/d01mlc83'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33554520</td>
<td>83886176</td>
<td>5b</td>
<td>363520</td>
<td>13:35</td>
<td>10:51:42</td>
<td>ML83SER</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/clrepl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>1</td>
<td>74</td>
<td>4568</td>
<td>0:00</td>
<td>10:54:01</td>
<td>ML83NP</td>
<td>/usr/lpp/lotus6b/lotus/bin/server ML83 /d01mlc83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16777306</td>
<td>33554519</td>
<td>75</td>
<td>1000</td>
<td>0:00</td>
<td>10:54:00</td>
<td>ML832</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/bin/tools/startup.sh 'ML83' '/d01mlc83'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33554520</td>
<td>83886176</td>
<td>6e</td>
<td>513024</td>
<td>1h19</td>
<td>10:53:51</td>
<td>ML83SER</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/server ML83 /d01mlc83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16777313</td>
<td>83886176</td>
<td>6d</td>
<td>358400</td>
<td>5:29</td>
<td>10:53:45</td>
<td>ML83SER</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/logasio NOTESLOGGER reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>83886176</td>
<td>76</td>
<td>365568</td>
<td>6:03</td>
<td>10:53:27</td>
<td>ML83EV1</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/event</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>83886176</td>
<td>77</td>
<td>371712</td>
<td>35:34</td>
<td>10:52:41</td>
<td>ML83UP1</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/update</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>83886176</td>
<td>78</td>
<td>364544</td>
<td>2:32</td>
<td>10:52:41</td>
<td>ML83AD1</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/adminp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>83886176</td>
<td>79</td>
<td>362496</td>
<td>0:03</td>
<td>10:52:40</td>
<td>ML83CA1</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/sched</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>83886176</td>
<td>80</td>
<td>362496</td>
<td>2:53</td>
<td>10:52:39</td>
<td>ML83AM1</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/amgr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>83886176</td>
<td>81</td>
<td>362496</td>
<td>0:41</td>
<td>10:52:39</td>
<td>ML83TM1</td>
<td>ML83NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/lpp/lotus6b/lotus/notes/60000/os390/tmmscan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

172  Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning
B.3.2 Sample output for domps with -p

Example B-2 shows the output from issuing the domps command with the -p parameter. This gives detailed process information.

<table>
<thead>
<tr>
<th>PID</th>
<th>PPID</th>
<th>ASID</th>
<th>THCNT</th>
<th>VIRTUAL</th>
<th>TIME ELAPSED</th>
<th>JOBNAME</th>
<th>RUSER</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>33554519</td>
<td>89</td>
<td>73</td>
<td>1</td>
<td>484</td>
<td>0:00</td>
<td>10:54:11</td>
<td>ML83</td>
<td>sh -c /usr/lpp/lotus6b/lotus/bin/tools/startup.sh 'ML83' '/d01mlc83'</td>
</tr>
<tr>
<td>33554520</td>
<td>83886176</td>
<td>5b</td>
<td>3</td>
<td>363520</td>
<td>13:35</td>
<td>10:51:53</td>
<td>ML83ER</td>
<td>ML83NP /usr/lpp/lotus6b/lotus/notes/60000/os390/clrepl</td>
</tr>
<tr>
<td>89</td>
<td>1</td>
<td>74</td>
<td>1</td>
<td>4568</td>
<td>0:00</td>
<td>10:54:12</td>
<td>ML83</td>
<td>ML83NP /usr/lpp/lotus6b/lotus/bin/server ML83 /d01mlc83</td>
</tr>
<tr>
<td>16777306</td>
<td>83886176</td>
<td>75</td>
<td>1</td>
<td>1000</td>
<td>0:00</td>
<td>10:54:11</td>
<td>ML83</td>
<td>ML83NP sh -c /usr/lpp/lotus6b/lotus/bin/tools/startup.sh 'ML83' '/d01mlc83'</td>
</tr>
<tr>
<td>16777306</td>
<td>83886176</td>
<td>6e</td>
<td>159</td>
<td>513024</td>
<td>1h19</td>
<td>10:54:02</td>
<td>ML83SER</td>
<td>ML83NP /usr/lpp/lotus6b/lotus/notes/60000/os390/server ML83 /d01mlc83</td>
</tr>
<tr>
<td>16777313</td>
<td>33554519</td>
<td>75</td>
<td>1</td>
<td>1000</td>
<td>0:00</td>
<td>10:54:11</td>
<td>ML83</td>
<td>ML83NP sh -c /usr/lpp/lotus6b/lotus/bin/tools/startup.sh 'ML83' '/d01mlc83'</td>
</tr>
<tr>
<td>16777334</td>
<td>83886176</td>
<td>89</td>
<td>2</td>
<td>360448</td>
<td>0:05</td>
<td>10:52:13</td>
<td>ML83SER</td>
<td>ML83NP /usr/lpp/lotus6b/lotus/notes/60000/os390/clrepl</td>
</tr>
</tbody>
</table>

Example: B-2  Output from domps command with -p
Example: B-3  Output of domps with -t

<table>
<thead>
<tr>
<th>PID</th>
<th>ASID</th>
<th>STID</th>
<th>TCBADDR</th>
<th>WTIME</th>
<th>SNUM</th>
<th>SVAL</th>
<th>LASTSYSC</th>
<th>SYSC</th>
<th>TIME</th>
<th>S</th>
<th>JOBNAME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>33554519</td>
<td>-</td>
<td>-</td>
<td>00000001</td>
<td>7f1860</td>
<td>0:00</td>
<td>-</td>
<td>?&amp;?L?&amp;?L?&amp;?L?&amp;?L?&amp;?L?&amp;?L</td>
<td>1SLP</td>
<td>0:18</td>
<td>SU</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>33554520</td>
<td>5b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16777306</td>
<td>75</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>83886176</td>
<td>6e</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16777313</td>
<td>6d</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Example B-2 on page 173 shows the output from issuing the domps command with the `-t` parameter, which gives detailed thread information.
Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning

---

- - 0000000f 7dfc0 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:05 DJV - -
- - 00000010 7da60 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -
- - 00000011 7d8c8 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -
- - 00000012 7d638 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:05 DJV - -
- - 00000013 7df3a8 0:00 5 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:18 DJV - -
- - 00000014 7df118 0:00 5 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:14 DJV - -
- - 00000015 7ded90 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -
- - 00000016 7de70 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:02 DJV - -
- - 00000017 7de870 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:02 DJV - -
- - 00000018 7de5e0 0:00 6 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:02 DJV - -
- - 0000002b 7cb20 0:00 5 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -
- - 0000002c 7cb920 0:00 5 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -
- - 00000000 7e6360 0:00 - - ISOP1SOP1SOP1SOP1SOP1SOP1SOP 1:41 SU - -
- - 00000001 7e2528 0:06 7 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 1:05 DJV - -
- - 00000002 7e2298 0:00 11 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:00 DJV - -
- - 0000002b 7cb20 0:00 5 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -
- - 0000002c 7cb920 0:00 5 6553 ISOP1SOP1SOP1SOP1SOP1SOP1SOP 0:04 DJV - -

176  Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning

---
Sample notes.ini file

This is an example of a notes.ini file.

[Notes]
Directory=/notesdata
KitType=2
SetupDB=setupweb.nsf
SERVER_POOL_TASKS=125
SERVER_ENABLE_THREADPOOL=1
PhoneLog=2
Log=log.nsf,1,0,5,120000
Ports=TCP
TCPIP0A=TCP,0,15,0,,45056,
TCPIP0A_TCPIPADDRESS=0,9.117.250.33:1352
Serial1=XPC,1,15,0,
Serial2=XPC,2,15,0,
DisabledPorts=Serial1,Serial2
LOG_REPLICATION=1
LOG_SESSIONS=1
SETUP_PERCENTDONE=100
ADMINPINTERVAL=15
ADMINPMODIFYPERSONDOCUMENTSAT=0
LOG_SESSIONS=1
SETUP_STATUS=Updating network settings
KeyFilename=domsrv.id
MailServer=CN=domsrv/OU=01/OU=M/O=IBM
Domain=IBMUS
ServerKeyFileName=domsrv.id
Admin=CN=Elsie Ramos/OU=Poughkeepsie/O=IBM
ServerTasks=Router,Amgr,Update,Adminp,Sched,CalConn,Collect,Tmmscan,tmmscan,tmmscan,dbscan
mail1.box,dbscan mail2.box,dbscan mail3.box,tivaddin
ServerTasksAt2=UpdAll
TemplateSetup=6010
Setup=6023
ServerSetup=6010
Timezone=5
DST=1
MTEnabled=0
SCHEDULE_VERSION=4
MAILTIMEOUT=3
ADMINPINTERVAL=15
ADMINPMODIFYPERSONDOCUMENTSAT=0
LOG_AGENTMANAGER=1
SHOW_TASK_DETAIL=1
LOG_VIEW_EVENTS=1
Fixup_Tasks=75
names=names
SERVER_CLUSTER_ON=0
server_restricted=0
DSTLaw=4,1,1,10,-1,1
AgentManagerVerboseMode=1
da_reload_interval=60
log_update=1
AMgr_DisableMailLookup=1
VIEW_REBUILD_DIR=/notesdata/edircat/viewbuild/
CleanSetup=1
EventSetup=600
SERVER_NAME_LOOKUP_NOUPDATE=1
SERVER_SHOW_PERFORMANCE=1
Passthru_LogLevel=0
Console_LogLevel=2
DOETimeout=10
NAMEDSTYLE0=0300426173696300000000000000000000000000000000000000000000000000000000000000010
10100000A0000000000000100A0050A000006400A0050A0000000000000000000000000000000000000000000000000000000000000000940400000000000
NAMEDSTYLE0_FACE=Default Sans Serif
NAMEDSTYLE1=030042756C6C6574000000000000000000000000000000000000000000000000000000000000010
101010B0C0000000000000100A0050A000006400A0050A0000000000000000000000000000000000000000000000000000000000000000494040000000000
NAMEDSTYLE1_FACE=Default Sans Serif
NAMEDSTYLE2=0300486561646C696E6500000000000000000000000000000000000000000000000000010
10101010B0C0000000000000100A0050A000006400A0050A0000000000000000000000000000000000000000000000000000000000000000494040000000000
NAMEDSTYLE2_FACE=Default Sans Serif
Information Center on the Internet
DefaultMailTemplate=mail6.ntf
Preferences=32
ServerTasksAt1=Catalog
NSF_BUFFER_POOL_SIZE_MB=384
server_availability_threshold=0
FileDlgDirectory=/notesdata
TCPIP1A=TCP,0,15,0,,45056,
COLLECT_DB_LOCK_WAITS=0
RouterOptimizePathRouting=1
SetupLeaveServerTasks=1
RouterAllowConcurrentXFERToAll=1
DEBUG_BTREE_ERRORS=0
MTCDoInsertsFirst=1
RouterIgnoreForeignSMTPDomains=1
ServerTasksAt4=Statlog
SMTPTranslateAddresses=2
DEBUG_THREADID=1
DEBUG_NTI_IOCP=0
MailCompactDisabled=0
disable_bcc_group_expansion=1
TRANSLOG_AutoFixup=1
TRANSLOG_UseAll=0
TRANSLOG_Style=0
TRANSLOG_Performance=2
TRANSLOG_MaxSize=2000
TRANSLOG_Path=/mlc96log
TRANSLOG_Status=1
Previous_TRANSLOG_Status=1
Previous_TRANSLOG_Path=/mlc96log/
Previous_TRANSLOG_Style=0
SMTPMaxSessions=24
REPLICATOR_LOG_NONEVENTS=1
DEBUG_MailLeaveSessionsOpen=0
WebAdminSetup=503
DominoConfigLevel=3
SERVER_SESSION_TIMEOUT=61
DEBUG_RM_DEBUG=1
MailConvertMIMEonTransfer=1
EXTMGR_ADDINS=smlnred
SmLicense=SNEF-9990-6749-5781-8518
SMRELAY_USER=smln_router
SmScanMem=4000
RouterEnableMailByDest=1
MailType=0
IOCP_DISABLE_ASYNC_NOTIFICATION=1
log_mailrouting=20
TivoliAuth=daa600be68d84c09f563d78019b684bc
TivoliPort=3001
TivoliNLSPATH=/usr/lpp/Tivoli/lcf/generic/msg_cat/%L/%N.cat:/usr/lpp/Tivoli/lcf/generic/msg_cat/%l/%N.cat:/usr/lib/nls/msg/%L/%N
TivoliTISDIR=/etc/Tivoli/lcf/dat
FaultRecovery_Build=Build V60_09262002GOLD
SHARED_MAIL=0
NAMELOOKUP_TRUST_DIRCAT=0
ServerName=DOMSRV/01/M/IBM
ServerNameNative=04030D35C4F0F1D4D3C3F9F661FOF161D461C9C2D4
NSF_QUOTA_METHOD=2
ServerController=0
FaultRecovery=1
CleanupTimeout=1800
CleanupScriptPath=/usr/lpp/lotus/notesapi/rc.domino.fatal
LDAPSERVER=LDAP://d01mlwt1.watson.ibm.com:389
UPDATE_TIMER=09/14/2002 02:51:05
ATCLastRunDate=10/11/2002 01:24:00
CATALOG_UPDATED_BY_BUILD=510
MIME_SAVE_CONVERT_FAILURES_DBNAME=badmail.nsf
MIME_SAVE_CONVERT_FAILURES_SENDTO=Jon Champlin/Westford/IBM
MIME_SAVE_CONVERT_FAILURES=1
MIME_SAVE_CONVERT_FAILURES_LASTCOPYTIME=09/26/2002 09:36:18
MIME_SAVE_CONVERT_FAILURES_MSLASTHOUR=2
JrnlEnbld=0
TIV2Xsrvname=DOMSRV/01/M/IBM
Sample output of the show statistics command

The following is a sample output of the show statistics command.

> sh stat

Agent.Daily.AccessDenials = 272
Agent.Daily.ScheduledRuns = 10053
Agent.Daily.TriggeredRuns = 1232
Agent.Daily.UnsuccessfulRuns = 1191
Agent.Daily.UsedRunTime = 22393 Seconds
Agent.Hourly.AccessDenials = 27
Agent.Hourly.ScheduledRuns = 778
Agent.Hourly.TriggeredRuns = 133
Agent.Hourly.UnsuccessfulRuns = 146
Agent.Hourly.UsedRunTime = 1419 Seconds
Calendar.Total.All.Appts.Reservations = 343477
Calendar.Total.All.Users.Resources = 2464
Calendar.Total.Appts = 343477
Calendar.Total.Reservations = 0
Calendar.Total.Users = 2464
Database.DAFailoverCount = 0
Database.DARefreshServerInfoCount = 31
Database.DAReloadCount = 0
Database.Database.BufferPool.MM.Reads = 49
Database.Database.BufferPool.MM.Writes = 29
Database.Database.BufferPool.Peak.Megabytes = 382
Database.Database.BufferPool.PerCentReadsInBuffer = 96.11
Database.DbCache.CurrentEntries = 1231
Database.DbCache.HighWaterMark = 1728
Database.DbCache.Hits = 765182
Database.DbCache.InitialDbOpens = 1910289
Database.DbCache.Lookups = 1970503
Database.DbCache.MaxEntries = 1152
Database.DbCache.OvercrowdingRejections = 16451
Database.ExtMgrPool.Peak = 65,406
Database.ExtMgrPool.Used = 314
Database.FreeHandleStack.FreeHandleStackHits = 1853490
Database.FreeHandleStack.HandleAllocations = 1856696
Database.FreeHandleStack.MissRate = 0
Database.LDAP.NAMELookupBindFailures = 0
Database.LDAP.NAMELookupBinds = 0
Database.LDAP.NAMELookupBytesReceived = 0
Database.LDAP.NAMELookupEntries = 0
Database.LDAP.NAMELookupFailures = 0
Database.LDAP.NAMELookupTotal = 0
Database.LDAP.NAMELookupTotalLookupTime = 0
Database.NAMELookupCachePool.Peak = 16,777,216
Database.NAMELookupCachePool.Used = 16,293,472
Database.NAMELookupCacheResets = 0
Database.NAMELookupMisses = 802,041
Database.NAMELookupTotal = 3063009
Database.NAMELookupTotalLookupTime = 63,515.981
Database.NIFPool.Peak = 10,485,760
Database.NIFPool.Used = 8,324,448
Database.NSFPool.Peak = 52,428,800
Database.NSFPool.Used = 33,037,344
Database.NSF.ClusterHashTable.EntriesWithSameIndex = 0
Database.NSF.ClusterHashTable.FreedEntriesOnCleanup = 0
Database.NSF.ClusterHashTable.HashedEntries = 0
Database.NSF.ClusterHashTable.HashIsFull = 0
Database.NSF.ClusterHashTable.MissedHashHits = 0
Database.NSF.ClusterHashTable.SuccessfullHashHits = 0
Database.NSF.Replicate.NotesMergedBack = 0
Database.NSF.Replicate.NotesReceived = 0
Database.NSF.Replicate.NotesReopened = 0
Database.NSF.Replicate.NotesSent = 0
Database.NSF.SignatureCache.Hits = 181186
Database.NSF.SignatureCache.Tries = 323698
Database.RM.Current.K.Restart.Redo.Hold = 126192
Database.RM.Current.K.UnCommited.Undo.Hold = 1
Database.RM.Current.MaxChkptScan.msec = 16789
Database.RM.Current.UnCommited.Trans = 2833
Database.RM.LastChkpt.Avg.K.Logged.PerTran = 14
Database.RM.LastChkpt.Interval.Sec = 229
Database.RM.LastChkpt.K.Logged = 41418
Database.RM.LastChkpt.Time = 10/11/2002 14:28:12 EDT
Database.RM.LastChkpt.Total.DBs = 1672
Database.RM.LastChkpt.Trans = 2833
Database.RM.Media.Log.Recs.Read = 0
Database.RM.Peak.Avg.K.Logged.PerTran = 1040
Database.RM.Peak.Interval.Sec = 10
Database.RM.Peak.K.Logged = 41629
Database.RM.Peak.Time = 10/08/2002 18:38:37 EDT
Database.RM.Peek.Total.DBs = 1112
Database.RM.Peek.Trans = 40
Database.RM.Peek.Trans.PerMin = 240
Database.RM.Restart.Duration.Sec = 406
Database.RM.Restart.K.Processed = 97144
Database.RM.Restart.Log.Rec.As.Analyzed = 83095
Database.RM.Restart.Log.Rec.Applied = 28980
Database.RM.Restart.Log.Rec.Read = 170349
Database.RM.Restart.Time = 10/08/2002 10:03:34 EDT
Database.RM.Restart.Total.DBS = 978
Database.RM.Restart.Undo.Trans = 0
Database.RM.SinceChkpt.K.Logged = 297
Database.RM.SinceChkpt.Trans = 41
Database.RM.SinceStartup.Aborts = 80
Database.RM.SinceStartup.Critical.Log.Times = 0
Database.RM.SinceStartup.Log.DiagRecs.Written = 691173
Database.RM.SinceStartup.Log.Rec.Applied = 8569
Database.RM.SinceStartup.Log.Rec.Read = 20492
Database.RM.SinceStartup.Log.Rec.Written = 23743797
Database.RM.SinceStartup.M.Logged = 31916
Database.RM.SinceStartup.Trans = 2867983
Database.RM.Sys.Logged = Enabled
Database.RM.Sys.Log.Type = Circular
Database.RM.Sys.M.Chkpt.Interval = 40
Database.RM.Sys.M.Log.Size = 1984
Database.RM.Sys.M.Redo.Limit = 120
Database.RM.Sys.Phase = Normal
Database.UBM.Buffer_pool_size = 98304
Disk.Fixed = 1
Disk.Remote = 0
Disk./notesdata.Free = 2,434,248,704
Disk./notesdata.Size = 7,382,384,640
Disk./Free = 231,706,624
Domino.Cache.Database.HitRate = 0
Domino.Cache.Database.Count = 3
Domino.Cache.Database.DisplaceRate = 0
Domino.Cache.Database.HitRate = 91.6666666666667
Domino.Cache.Database.MaxSize = 128
Domino.Cache.Session.Count = 0
Domino.Cache.Session.MaxSize = 1,000
Domino.Cache.User.Count = 0
Domino.Cache.User.MaxSize = 64
Domino.Command.CopyToFolder = 0
Domino.Command.CreateDocument = 0
Domino.Command.DeleteDocument = 0
Domino.Command.DeleteDocuments = 0
Domino.Command.EditDocument = 0
Domino.Command.GetOrbCookie = 0
Domino.Command.MoveToFolder = 0
Domino.Command.Navigate = 0
Domino.Command.OpenAbout = 0
Domino.Command.OpenAgent = 0
Domino.Command.OpenCssResource = 0
Domino.Command.OpenDatabase = 0
Domino.Command.OpenDocument = 0
Domino.Command.OpenElement = 0
Domino.Command.OpenFileResource = 0
Domino.Command.OpenForm = 0
Domino.Command.OpenHelp = 0
Domino.Command.OpenIcon = 0
Domino.Command.OpenImageResource = 0
Domino.Command.OpenJavascriptLib = 0
Domino.Command.OpenNavigator = 0
Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning

Domino.Command.OpenPreferences = 0
Domino.Command.OpenServer = 0
Domino.Command.OpenView = 0
Domino.Command.ReadForm = 0
Domino.Command.Redirect = 0
Domino.Command.RemoveFromFolder = 0
Domino.Command.RequestCert = 0
Domino.Command.SaveDocument = 0
Domino.Command.SearchDomain = 0
Domino.Command.SearchSite = 0
Domino.Command.SearchView = 0
Domino.Command.Total = 0
Domino.Command.Unknown = 0
Domino.Config.ActiveThreads.Max = 40
Domino.Config.ActiveThreads.Min = 20
Domino.Config.AllowDirectoryLinks = 1
Domino.Config.Directory.CGI = /notesdata/domino/cgi-bin
Domino.Config.DNSLookup = 0
Domino.Config.EnforceAccess = 0
Domino.Config.HomeURL = /?Open
Domino.Config.HostName =
Domino.Config.Image.Format = GIF
Domino.Config.Image.Interlaced = 1
Domino.Config.Log.Filter =
Domino.Config.Log.TimeStamp = 0
Domino.Config.PortNumber = 80
Domino.Config.PortStatus = 1
Domino.Config.Timeout.CGI = 5
Domino.Config.Timeout.IdleThread = 0
Domino.Config.Timeout.Input = 2
Domino.Config.Timeout.Output = 20
Domino.Config.URLpath.CGI = /cgi-bin
Domino.Config.URLpath.Icons = /icons
Domino.Config.URLpath.JavaRoot = /domjava
Domino.Config.URLpath.Servlet = /servlet
Domino.Config.View.Lines = 30
Domino.Config.WelcomePage = default.htm
FT.Index.Bytes.PerHour = 24579111
FT.Index.Count = 13947
FT.Index.Documents.Added = 32061
FT.Index.Documents.Deleted = 24380
FT.Index.Documents.Updated = 12366
FT.Index.Total.Bytes = 261,495,691
FT.Index.Total.TimeMS = 38,300,185
FT.Search.Average.TimeMS = 459
FT.Search.Count = 10992
FT.Search.Max.ActualHits = 17,625
FT.Search.Max.Results = 17,625
FT.Search.Open = 19
FT.Search.Results.Zero = 10080
FT.Search.Total.ActualHits = 17,625
FT.Search.Total.Results = 17,625
FT.Search.Total.TimeMS = 5,041,355
Mail.AverageDeliverTime = 8
Mail.AverageServerHops = 1
Mail.AverageSizeDelivered = 63
Mail.CurrentByteDeliveryRate = 100602
Mail.CurrentByteTransferRate = 6916
Mail.CurrentMessageDeliveryRate = 1
Mail.CurrentMessageTransferRate = 0
Mail.DBCacheEntries = 1151
Mail.DBCacheHits = 214394
Mail.DBCacheReads = 310707
MAIL.Dead = 0
Mail.Delivered = 298449
Mail.DeliveredSize.10KB_to_100KB = 86684
Mail.DeliveredSize.1MB_to_10MB = 3098
Mail.DeliveredSize.Over_100MB = 3
Mail.DeliveredSize.Under_1KB = 9143
Mail.Deliveries = 311329
Mail.DeliveryThreads.Active = 0
Mail.DeliveryThreads.Max = 25
Mail.DeliveryThreads.Total = 25
Mail.Domain = IBMUS
MAIL.Hold = 3
Mail.Mailbox.AccessConflicts = 1057
Mail.Mailbox.Accesses = 118475
Mail.Mailbox.AccessWarnings = 25395
Mail.Mailbox.CurrentAccesses = 1
Mail.Mailbox.MaxConcurrentAccesses = 35
Mail.MailboxOpens = 120283
Mail.MailMaxConcurrentXferThreads = 0
Mail.MailMaxDeliveryThreads = 25
Mail.MailMaxTransferThreads = 25
Mail.Mailbox.Number_Of_Mailboxes = 25
Mail.MaximumDeliverTime = 14752
Mail.MaximumServerHops = 17
Mail.MaximumSizeDelivered = 390664
Mail.MinimumDeliverTime = 1
Mail.MinimumServerHops = 1
Mail.MinimumSizeDelivered = 1
Mail.PeakByteDeliveryRate = 346081
Mail.PeakByteTransferRate = 187367
Mail.PeakMessageDeliveryRate = 10
Mail.PeakMessageDeliveryTime = 10/08/2002 10:17:13 EDT
Mail.PeakMessagesDelivered = 1067
Mail.PeakMessagesTransferred = 248
Mail.PeakMessageTransferRate = 4
Mail.PeakTotalBytesDelivered = 34262113
Mail.PeakTotalBytesTransferred = 11429447
Mail.SMLNAmountMails = 0
Mail.SMLNBlockedMails = 0
Mail.SMLNCleanedMails = 0
Mail.SMLNDeletedFiles = 8
Mail.SMLNInfectedFiles = 0
Mail.SMLNInfectedMails = 0
Mail.SMLNProcessedMails = 44878
Mail.SMLNQuarantinedFiles = 0
Mail.SMLNRemovedMacrosFiles = 0
Mail.SMLNScannedAmount = 173754654
Mail.SMLNScannedFiles = 6097
Mail.SMLNTotalScanned = 0
Mail.SMLNUnCleanableMails = 0
Mail.SMLNUnCleanedFiles = 0
MAIL.TotalFailures = 795
Mail.TotalKBDelivered = 19,667,336
Mail.TotalKBTransferred = 8,146,702
Mail.TotalKBTransferred.NRPC = 7,629,136
Mail.TotalKBTransferred.SMTP = 517,566
Mail.TotalPending = 51
Mail.TotalRouted = 448764
Mail.TotalRouted.NRPC = 430207
Mail.TotalRouted.SMTP = 18557
Mail.TransferFailures = 28
Mail.TransferFailures.SMTP = 28
Mail.Transferred = 99305
Mail.TransferredSize.100KB_to_1MB = 7853
Mail.TransferredSize.10KB_to_100KB = 27090
Mail.TransferredSize.10MB_to_100MB = 79
Mail.TransferredSize.1KB_to_10KB = 54686
Mail.TransferredSize.1MB_to_10MB = 1238
Mail.TransferredSize.Under_1KB = 8359
Mail.Transferred.NRPC = 88104
Mail.Transferred.SMTP = 11201
Mail.TransferThreads.Active = 13
Mail.TransferThreads.Concurrent.Highest = 12
Mail.TransferThreads.Concurrent.Max = 12
Mail.TransferThreads.Max = 25
Mail.TransferThreads.Total = 25
MAIL.Waiting = 52
Mail.WaitingForDeliveryRetry = 5
MAIL.WaitingForDIR = 0
MAIL.WaitingForDNS = 0
MAIL.WaitingRecipients = 780
MailByDest.1-714-966-8495.TotalFailures = 1
MailByDest.br.LastFailure = 10/08/2002 10:37:06 EDT
MailByDest.br.TotalFailures = 1
MailByDest.CN=DOMSRV1/OU=01/OU=H/O=IBM.LastSuccess = 10/11/2002 14:28:14 EDT
MailByDest.CN=DOMSRV1/OU=01/OU=H/O=IBM.TotalKBTransferred = 1,539,619
MailByDest.CN=DOMSRV1/OU=01/OU=H/O=IBM.TotalRouted = 31132
MailByDest.CN=DOMSRV1/OU=01/OU=H/O=IBM.Transferred = 15033
MailByDest.CN=DOMSRV2/OU=01/OU=H/O=IBM.LastFailure = 10/10/2002 11:14:56 EDT
MailByDest.CN=DOMSRV2/OU=01/OU=H/O=IBM.LastSuccess = 10/11/2002 14:26:13 EDT
MailByDest.CN=DOMSRV2/OU=01/OU=H/O=IBM.TotalKBTransferred = 470,049
MailByDest.CN=DOMSRV2/OU=01/OU=H/O=IBM.TotalRouted = 10464
MailByDest.CN=DOMSRV2/OU=01/OU=H/O=IBM.Transferred = 5158
MailByDest.CN=DOMSRV/OU=01/OU=M/O=IBM.LastFailure = 10/11/2002 14:25:38 EDT
MailByDest.CN=DOMSRV/OU=01/OU=M/O=IBM.TotalKBTransferred = 675
MailByDest.com.LastFailure = 10/08/2002 14:26:59 EDT
MailByDest.com.TotalFailures = 1
MailByDest.[$LocalDelivery].Delivered = 298449
MailByDest.[$LocalDelivery].LastSuccess = 10/11/2002 14:28:14 EDT
MailByDest.[$LocalDelivery].TotalRouted = 298449
Mem.Allocated = 938872726
Mem.Allocated.Process = 178521262
Mem.Allocated.Shared = 760351464
Mem.Availability = Plentiful
Mem. PhysicalRAM = 535822336
Monitor.AGENT MANAGER.Warning(High) = 9,096
Monitor.DATABASE COMPACTOR.Failure = 7
Monitor.Last.AGENT MANAGER.Warning(High) = 13,089
Monitor.Last.DATABASE COMPACTOR.Failure = 13,064
Monitor.Last.ROUTER.Warning(High) = 13,270
Monitor.Last.SCHEDULE MANAGER.Warning(High) = 13,103
Monitor.Last.Server.Failure = 2,055
Monitor.Last.Server.Warning(High) = 582
Monitor.ROUTER.Warning(High) = 31
Monitor.SCHEDULE MANAGER.Warning(High) = 1,869
Monitor.Server.Failure = 454
Monitor.Server.Warning(High) = 5,780
NET.GroupCache.Hits = 626,886
NET.GroupCache.Misses = 62,794
NET.GroupCache.NumEntries = 2,700
NET.GroupCache.Size = 719,466
NET.GroupCache.Used = 668,206
NET.Log.DOMSRV/01/M/IBM.PeakUnwrittenEntries = 81
NET.Log.DOMSRV/01/M/IBM.UnwrittenEntries = 5
NET.TCPIPOA.BytesReceived = 19,678,739,828
NET.TCPIPOA.BytesSent = 72,335,295,450
NET.TCPIPOA.Sessions.Established.Incoming = 838647
NET.TCPIPOA.Sessions.Established.Outgoing = 104406
NET.TCPIPOA.Sessions.Limit = 65535
NET.TCPIPOA.Sessions.LimitMax = 65535
NET.TCPIPOA.Sessions.LimitMin = 10
NET.TCPIPOA.Sessions.Peak = 1358
NET.TCPIPOA.Sessions.Recycled = 0
NET.TCPIPOA.Sessions.Recycling = 0
Platform.ActiveNumOfDominoPartitions = 2
Platform.Memory.AvailableFrameCount = 125
Platform.Memory.AvailableFrameCount.Min = 8
Platform.Memory.InQueue = 90
Platform.Memory.OutReadyQueue = 0
Platform.Memory.OutReadyQueue.Avg = 0
Platform.Memory.OutReadyQueue.Min = 0
Platform.Memory.OutWaitQueue = 70
Platform.Memory.OutWaitQueue.Avg = 73.4
Platform.Memory.OutWaitQueue.Min = 59
Platform.Memory.PageHighUIC = 220
Platform.Memory.PagesPerSec = 0
Platform.Memory.PagesPerSec.Avg = 0
Platform.Memory.PagesPerSec.Peer = 0
Platform.Memory.RAM.TotalMBytes = 512
Platform.Process.ActiveDomino.TotalCpuUtil = 73.9
Platform.Process.adminp.1.CurrentLocation = 0
Platform.Process.adminp.1.PctCpuUtil = 0.2
Platform.Process.amgr.2.CurrentLocation = 0
Platform.Process.amgr.2.PctCpuUtil = 0.1
Platform.Process.amgr.3.CurrentLocation = 0
Platform.Process.calconn.1.CurrentLocation = 0
Platform.Process.calconn.1.PctCpuUtil = 0
Platform.Process.collect.1.PctCpuUtil = 0.2
Platform.Process.event.1.CurrentLocation = 0
Platform.Process.pupdate.2.CurrentLocation = 0
Platform.Process.router.1.CurrentLocation = 0
Platform.Process.router.1.PctCpuUtil = 8
Platform.Process.sched.1.CurrentLocation = 0
Platform.Process.sched.1.PctCpuUtil = 0.2
Platform.Process.server.1.PctCpuUtil = 33.2
Platform.Process.server.2.CurrentLocation = 0
Platform.Process.server.2.PctCpuUtil = 1.9
Platform.Process.tivaddin.1.PctCpuUtil = 0.1
Platform.Process.tmmscan.1.PctCpuUtil = 0.7
Platform.Process.tmmscan.2.CurrentLocation = 0
Platform.Process.tmmscan.2.PctCpuUtil = 0.8
Platform.Process.tmmscan.3.CurrentLocation = 0
Platform.System.LPARPctCpuUtil = 21
Platform.System.LPARPctCpuUtil.Peak = 65
Platform.System.PctCombinedCpuUtil = 22
Platform.System.PctCombinedCpuUtil.Peak = 100
Platform.System.TotalNumProcs = 160
Platform.System.TotalNumProcs.Avg = 164.9
Platform.System.TotalNumProcs.Peak = 197
Platform.Time.SampleRateInMins = 1
Sem.Timeouts = 039E:6 4245:1
Server.Administrators = Admin,D01D02Iris,D04TEST2,D01 Servers
Server.Administrators.Access = Admin,D01D02Iris,D04TEST2,D01 Servers
Server.AvailabilityIndex = 0
Server.AvailabilityThreshold = 0
Server.BootID = 5022300
Server.BusyTimeQuery.ReceivedCount = 93953
Appendix D. Sample output of the show statistics command

Server.CPU.Count = 8
Server.ElapsedTime = 3 days 04:22:27
Server.ExpansionFactor = 426.095679555633
Server.MailBoxes = 3
Server.Name = CN=DOMSRV/OU=01/OU=M/O=IBM
Server.OpenRequest.MaxUsers = 0
Server.OpenRequest.Restricted = 0
Server.Path.ConfigFile = /notesdata/notes.ini
Server.Path.Data = /notesdata
Server.Ports = TCPIP0A
Server.PoweredBy = Notes
Server.Sessions.Dropped = 1687
Server.SharedMail = 0
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:26:19 EDT]
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:26:44 EDT]
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:26:50 EDT]
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:27:05 EDT]
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:27:00 EDT]
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:26:34 EDT]
Server.Task = Router: Searching for mail to deliver: [10/11/2002 14:26:50 EDT]
Server.Task = Router: Searching for mail to transfer: [10/11/2002 14:26:13 EDT]
Server.Tasks = 1146
Server.Title = Pougkeepsie Version 5 Server
Server.Trans.PerMinute = 3491
Server.Trans.PerMinute.Peak = 10583
Server.Trans.Total = 12494270
Server.Users = 929
Server.Users.1MinPeak = 192
Server.Users.1MinPeakTime = 10/08/2002 10:37:52 EDT
Server.Users.5MinPeak = 464
Server.Users.5MinPeakTime = 10/10/2002 13:31:10 EDT
Server.Users.Active = 7
Server.Users.Active15Min = 549
Server.Users.Active1Min = 116
Server.Users.Active30Min = 728
Server.Users.Active3Min = 227
Server.Users.Active5Min = 308
Server.Users.Peak = 1352
Server.Version.Notes.BuildNumber = 190
Server.Version.OS = OS/390 13.00 03
Server.WorkThreads = 125
SMTP.Command.DATA = 91591
SMTP.Command.EHLO = 90613
SMTP.Command.HELO = 4
SMTP.Command.MAIL = 91604
SMTP.Command.QUIT = 90613
SMTP.Command.RCPT = 98332
SMTP.Command.RSET = 991
SMTP.MessagesProcessed = 91591
SMTP.MsgRecipients.Ave = 1
SMTP.MsgRecipients.Max = 19
SMTP.MsgRecipients.Min = 1
SMTP.MsgRecipients.Total = 98332
SMTP.MsgSizeKBytes.Ave = 18
SMTP.MsgSizeKBytes.Max = 11791
SMTP.MsgSizeKBytes.Min = 1
SMTP.MsgSizeKBytes.Total = 1654687
SMTP.Server.Running = TRUE
SMTP.SessionDuration.10Sec_to_15Sec = 665
SMTP.SessionDuration.15Sec_to_20Sec = 286
SMTP.SessionDuration.20Sec_to_25Sec = 285
SMTP.SessionDuration.25Sec_to_30Sec = 259
SMTP.SessionDuration.30Sec_to_35Sec = 175
SMTP.SessionDuration.35Sec_to_40Sec = 133
SMTP.SessionDuration.40Sec_to_45Sec = 77
SMTP.SessionDuration.45Sec_to_50Sec = 47
SMTP.SessionDuration.50Sec_to_55Sec = 41
SMTP.SessionDuration.55Sec_to_60Sec = 32
SMTP.SessionDuration.5Sec_to_10Sec = 2227
SMTP.SessionDuration.Ave = 1
SMTP.SessionDuration.Max = 384
SMTP.SessionDuration.Min = 0
SMTP.SessionDuration.Over_60Sec = 329
SMTP.SessionDuration.Under_5Sec = 86061
SMTP.Inbound.Accept.Queue = 0
SMTP.Inbound.Active = 1
SMTP.Inbound.BytesReceived = 0
SMTP.Inbound.BytesSent = 0
SMTP.Inbound.Peak = 25
SMTP.Inbound.Total = 90643
SMTP.Inbound.Total.SSL = 0
SMTP.Inbound.Total.SSL.Bad_Handshake = 0
SMTP.Outbound.Active = 0
SMTP.Outbound.BytesReceived = 7,073,532
SMTP.Outbound.BytesSent = 756,577,983
SMTP.Outbound.Peak = 5
SMTP.Outbound.Total = 11169
SMTP.Outbound.Total.SSL = 0
SMTP.Threads.Busy = 0
SMTP.Threads.Idle = 26
SMTP.Threads.InThreadPool = 26
SMTP.Threads.Peak = 26
Stats.Time.Start = 10/08/2002 09:56:46 EDT
918 statistics found
Appendix E. Domino modify zFS command

The following is a sample output of the `modify zfs,query,all` command.

Example: E-1  f zfs,query,all output

<table>
<thead>
<tr>
<th>Operation</th>
<th>Count</th>
<th>Avg Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>zfs_opens</td>
<td>649382</td>
<td>0.102</td>
</tr>
<tr>
<td>zfs_closes</td>
<td>648532</td>
<td>0.627</td>
</tr>
<tr>
<td>zfs_reads</td>
<td>20811268</td>
<td>2.132</td>
</tr>
<tr>
<td>zfs_writes</td>
<td>20457725</td>
<td>0.538</td>
</tr>
<tr>
<td>zfs_ioctl</td>
<td>12</td>
<td>0.005</td>
</tr>
<tr>
<td>zfs_getattr</td>
<td>12075707</td>
<td>0.008</td>
</tr>
<tr>
<td>zfs_setattr</td>
<td>35776</td>
<td>0.154</td>
</tr>
<tr>
<td>zfs_accesses</td>
<td>33924888</td>
<td>0.013</td>
</tr>
<tr>
<td>zfs_lookup</td>
<td>7225927</td>
<td>0.052</td>
</tr>
<tr>
<td>zfs_create</td>
<td>35685</td>
<td>0.914</td>
</tr>
<tr>
<td>zfs_remove</td>
<td>35616</td>
<td>2.594</td>
</tr>
<tr>
<td>zfs_link</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_rename</td>
<td>5075</td>
<td>0.857</td>
</tr>
<tr>
<td>zfs_mkdir</td>
<td>7919</td>
<td>8.742</td>
</tr>
<tr>
<td>zfs_rmdir</td>
<td>16870</td>
<td>0.973</td>
</tr>
<tr>
<td>zfs_readdir</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_symlink</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_readlink</td>
<td>726679</td>
<td>0.004</td>
</tr>
<tr>
<td>zfs_fsync</td>
<td>49148</td>
<td>11.680</td>
</tr>
<tr>
<td>zfs_trunc</td>
<td>60</td>
<td>13.135</td>
</tr>
<tr>
<td>zfs_lockctl</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_audit</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_inactive</td>
<td>1534697</td>
<td>0.006</td>
</tr>
<tr>
<td>zfs_recovery</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_vget</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_pfsctl</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>zfs_statfss</td>
<td>3534</td>
<td>2.894</td>
</tr>
<tr>
<td>zfs_mount</td>
<td>30</td>
<td>763.910</td>
</tr>
<tr>
<td>zfsUnmount</td>
<td>0</td>
<td>0.000</td>
</tr>
</tbody>
</table>
**TOTALS**            98244530             0.587

Number of service threads: 10  (stacksize=OK)
Requests: 30  Queued: 0  (0.0%)

User File (VM) Caching System Statistics
----------------------------------------------

External Requests:
-------------------
Reads 20810947  Fsyncs 1707204  Opens 2987402
Writes 20457713  Setattrs 49981  Unmaps 35623
Async Reads 6036950  Getattrs 9604992  Schedules 316656
Flushes 0

File System Reads:
---------------------
Reads Faulted 7494374 (Fault Ratio 36.11%)  Writes Faulted 941971 (Fault Ratio 4.604%)
Read Wait 491033 (Wait Ratio 2.359%)
Total Reads 9594417

File System Writes:
---------------------
Scheduled Writes 3793649  Sync Waits 1952213
Error Writes 0  Error Waits 0
Scheduled Deletes 0
Page Reclaim Writes 114389  Reclaim Waits 19487
Write Waits 25114 (Wait Ratio 0.122%)

File Management: (File struct size=168)
---------------------------------------------
Max Files: 8192  Allocated: 8192
Lookups 2987402  Hits 2946895 (Hit Ratio 98.644%)

Page Management (Segment Size = 64K) (Page Size = 4K)
---------------------------------------------
Total Pages 65536  Free 65
Segments 11827
Steal Invocations 1471943  Waits for Reclaim 2978

Number of dataspaces used: 4  Pages per dataspace: 16384

<table>
<thead>
<tr>
<th>Dataspace</th>
<th>Allocated Segments</th>
<th>Free Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZFSUCD00</td>
<td>1348</td>
<td>5</td>
</tr>
<tr>
<td>ZFSUCD01</td>
<td>1468</td>
<td>22</td>
</tr>
<tr>
<td>ZFSUCD02</td>
<td>1425</td>
<td>25</td>
</tr>
<tr>
<td>ZFSUCD03</td>
<td>1356</td>
<td>13</td>
</tr>
</tbody>
</table>

zFS Vnode Op Counts

<table>
<thead>
<tr>
<th>Vnode Op</th>
<th>Count</th>
<th>Vnode Op</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>efs_hold</td>
<td>0</td>
<td>efs_readdir</td>
<td>497380</td>
</tr>
<tr>
<td>efs_rele</td>
<td>0</td>
<td>efs_create</td>
<td>35685</td>
</tr>
<tr>
<td>efs_inactive</td>
<td>0</td>
<td>efs_remove</td>
<td>35616</td>
</tr>
<tr>
<td>efs_getattr</td>
<td>16959507</td>
<td>efs_rename</td>
<td>5075</td>
</tr>
<tr>
<td>efs_setattr</td>
<td>45038</td>
<td>efs_mkdir</td>
<td>7919</td>
</tr>
</tbody>
</table>
Appendix E. Domino modify zFS command

<table>
<thead>
<tr>
<th>Command</th>
<th>Count</th>
<th>Wait</th>
<th>Cancel</th>
<th>Merge</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>efs_access</td>
<td>34574155</td>
<td>16870</td>
<td>0</td>
<td>0</td>
<td>File System Metadata</td>
</tr>
<tr>
<td>efs_lookup</td>
<td>7288545</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Log File</td>
</tr>
<tr>
<td>efs_getvolume</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache read</td>
</tr>
<tr>
<td>efs_getlength</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async delete write</td>
</tr>
<tr>
<td>efs_ofsfid</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache write</td>
</tr>
<tr>
<td>efs_fid</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async write</td>
</tr>
<tr>
<td>efs_vmread</td>
<td>9594403</td>
<td>5035858</td>
<td>0</td>
<td>0</td>
<td>User file cache direct read</td>
</tr>
<tr>
<td>efs_vmwrite</td>
<td>3908033</td>
<td>115886</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async delete write</td>
</tr>
<tr>
<td>efs_clrsetid</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async write</td>
</tr>
<tr>
<td>efs_atime</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async write</td>
</tr>
<tr>
<td>Total zFS Vnode Ops</td>
<td>90555370</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vnodes</th>
<th>Requests</th>
<th>Hits</th>
<th>Ratio</th>
<th>Allocates</th>
<th>Deletes</th>
</tr>
</thead>
<tbody>
<tr>
<td>16384</td>
<td>5234920</td>
<td>5229101</td>
<td>99.8%</td>
<td>43604</td>
<td>43605</td>
</tr>
</tbody>
</table>

LFS Vnode structure size: 296 bytes

SAF Access Cache Requests: 0 hits: 0 (hit ratio 0.0%)

Metadata Caching Statistics

<table>
<thead>
<tr>
<th>Buffers (K bytes)</th>
<th>Requests</th>
<th>Hits</th>
<th>Ratio</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096</td>
<td>32768</td>
<td>59776713</td>
<td>95.7%</td>
<td>19930741</td>
</tr>
</tbody>
</table>

Directory Cache Statistics

<table>
<thead>
<tr>
<th>Dir Blocks (K bytes)</th>
<th>Requests</th>
<th>Hits</th>
<th>Ratio</th>
<th>Deletes</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1024</td>
<td>17247769</td>
<td>97.1%</td>
<td>7988</td>
</tr>
</tbody>
</table>

Transaction Cache Statistics

Transactions started: 4789371 Lookup on tran: 43429013 EC Merges: 904
Allocated Transactions: 17843 (Act= 0, Pend= 0, Comp=17112, Free= 731)

I/O Summary By Type

<table>
<thead>
<tr>
<th>Count</th>
<th>Waits</th>
<th>Cancel</th>
<th>Merge</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>173710</td>
<td>91120</td>
<td>0</td>
<td>0</td>
<td>File System Metadata</td>
</tr>
<tr>
<td>230100</td>
<td>82965</td>
<td>0</td>
<td>16887</td>
<td>Log File</td>
</tr>
<tr>
<td>13500233</td>
<td>10924568</td>
<td>11735</td>
<td>17673</td>
<td>User File Data</td>
</tr>
</tbody>
</table>

I/O Summary By Circumstance

<table>
<thead>
<tr>
<th>Count</th>
<th>Waits</th>
<th>Cancel</th>
<th>Merge</th>
<th>Circumstance</th>
</tr>
</thead>
<tbody>
<tr>
<td>95376</td>
<td>95361</td>
<td>0</td>
<td>0</td>
<td>Metadata cache read</td>
</tr>
<tr>
<td>9587637</td>
<td>8918915</td>
<td>0</td>
<td>22</td>
<td>User file cache direct read</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>Log file read</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async delete write</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache write</td>
</tr>
<tr>
<td>28969</td>
<td>3929</td>
<td>42</td>
<td>0</td>
<td>Metadata cache lazy write</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async delete write</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache async write</td>
</tr>
<tr>
<td>3878749</td>
<td>1996774</td>
<td>11693</td>
<td>17651</td>
<td>User File cache direct write</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Metadata cache file sync write</td>
</tr>
<tr>
<td>83273</td>
<td>698</td>
<td>0</td>
<td>0</td>
<td>Metadata cache sync daemon write</td>
</tr>
</tbody>
</table>
0           0           0           0  Metadata cache aggregate detach write
0           0           0           0  Metadata cache buffer block reclaim write
0           0           0           0  Metadata cache buffer allocation write
0           0           0           0  Metadata cache file system quiesce write
0           0           0           0  Metadata cache log file full write
230070       82935           0       16887  Log file write
0           0           0           0  Metadata cache shutdown write

zFS I/O by Currently Attached Aggregate

<table>
<thead>
<tr>
<th>Volser</th>
<th>I/Os</th>
<th>Mode</th>
<th>Reads</th>
<th>K bytes</th>
<th>Writes</th>
<th>K bytes</th>
<th>Dataset Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>N67115</td>
<td>2</td>
<td>R/W</td>
<td>763282</td>
<td>11229532</td>
<td>105066</td>
<td>24963676</td>
<td>ML96.ZFS.DO1MLC96</td>
</tr>
<tr>
<td>N67089</td>
<td>2</td>
<td>R/W</td>
<td>324836</td>
<td>7509696</td>
<td>33226</td>
<td>639032</td>
<td>ML96.ZFS.DO1MLC96.MAIL1</td>
</tr>
<tr>
<td>SK4303</td>
<td>1</td>
<td>R/W</td>
<td>258575</td>
<td>6591068</td>
<td>24257</td>
<td>398936</td>
<td>ML96.ZFS.DO1MLC96.MAIL2</td>
</tr>
<tr>
<td>N65484</td>
<td>2</td>
<td>R/W</td>
<td>415994</td>
<td>10296364</td>
<td>42787</td>
<td>835076</td>
<td>ML96.ZFS.DO1MLC96.MAIL3</td>
</tr>
<tr>
<td>N6548C</td>
<td>2</td>
<td>R/W</td>
<td>382927</td>
<td>8986884</td>
<td>34567</td>
<td>610852</td>
<td>ML96.ZFS.DO1MLC96.MAIL4</td>
</tr>
<tr>
<td>N65482</td>
<td>2</td>
<td>R/W</td>
<td>214962</td>
<td>5400908</td>
<td>26163</td>
<td>564124</td>
<td>ML96.ZFS.DO1MLC96.MAIL5</td>
</tr>
<tr>
<td>N67000</td>
<td>2</td>
<td>R/W</td>
<td>211901</td>
<td>5626684</td>
<td>18532</td>
<td>359164</td>
<td>ML96.ZFS.DO1MLC96.MAIL6</td>
</tr>
<tr>
<td>N67214</td>
<td>2</td>
<td>R/W</td>
<td>244814</td>
<td>6833336</td>
<td>41823</td>
<td>1204644</td>
<td>ML96.ZFS.DO1MLC96.MAIL7</td>
</tr>
<tr>
<td>N67216</td>
<td>2</td>
<td>R/W</td>
<td>223517</td>
<td>6741412</td>
<td>23894</td>
<td>504060</td>
<td>ML96.ZFS.DO1MLC96.MAIL8</td>
</tr>
<tr>
<td>N67217</td>
<td>2</td>
<td>R/W</td>
<td>278968</td>
<td>7290068</td>
<td>40151</td>
<td>705748</td>
<td>ML96.ZFS.DO1MLC96.MAIL9</td>
</tr>
<tr>
<td>N67211</td>
<td>2</td>
<td>R/W</td>
<td>264418</td>
<td>6206552</td>
<td>37085</td>
<td>792928</td>
<td>ML96.ZFS.DO1MLC96.MAIL10</td>
</tr>
<tr>
<td>N67213</td>
<td>2</td>
<td>R/W</td>
<td>229512</td>
<td>5815784</td>
<td>21391</td>
<td>300028</td>
<td>ML96.ZFS.DO1MLC96.MAIL11</td>
</tr>
<tr>
<td>N65487</td>
<td>2</td>
<td>R/W</td>
<td>374960</td>
<td>9140408</td>
<td>40151</td>
<td>705748</td>
<td>ML96.ZFS.DO1MLC96.MAIL12</td>
</tr>
<tr>
<td>N67212</td>
<td>2</td>
<td>R/W</td>
<td>241248</td>
<td>6741412</td>
<td>23894</td>
<td>504060</td>
<td>ML96.ZFS.DO1MLC96.MAIL13</td>
</tr>
<tr>
<td>N67215</td>
<td>2</td>
<td>R/W</td>
<td>311878</td>
<td>7083896</td>
<td>34705</td>
<td>555656</td>
<td>ML96.ZFS.DO1MLC96.MAIL14</td>
</tr>
<tr>
<td>N65488</td>
<td>2</td>
<td>R/W</td>
<td>339631</td>
<td>8162300</td>
<td>31007</td>
<td>522344</td>
<td>ML96.ZFS.DO1MLC96.MAIL15</td>
</tr>
<tr>
<td>N65483</td>
<td>2</td>
<td>R/W</td>
<td>252009</td>
<td>4034248</td>
<td>1910227</td>
<td>20557724</td>
<td>ML96.ZFS.DO1MLC96.TRANSLOG</td>
</tr>
<tr>
<td>N65416</td>
<td>3</td>
<td>R/W</td>
<td>811306</td>
<td>4034248</td>
<td>1910227</td>
<td>20557724</td>
<td>ML96.ZFS.DO1MLC96.TRANSLOG</td>
</tr>
</tbody>
</table>

*TOTALS*

Total number of waits for I/O:  11098591
Average I/O wait time:  4.549 (msecs)

Locking Statistics

Untimed sleeps:  95 Timed Sleeps:  0 Wakeups:  93
Total waits for locks:  109044
Average lock wait time:  7.826 (msecs)
Total monitored sleeps:  3002
Average monitored sleep time:  35.378 (msecs)
### Top 15 Most Highly Contended Locks

<table>
<thead>
<tr>
<th>Thread</th>
<th>Async</th>
<th>Spin</th>
<th>Wait</th>
<th>Disp.</th>
<th>Resol.</th>
<th>Pct.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>66299</td>
<td>0</td>
<td>40954</td>
<td>49.8%</td>
<td>49.8%</td>
<td></td>
<td></td>
<td>Vnode lock</td>
</tr>
<tr>
<td>31071</td>
<td>0</td>
<td>219</td>
<td>14.5%</td>
<td>14.5%</td>
<td></td>
<td></td>
<td>User-cache segment lock</td>
</tr>
<tr>
<td>42</td>
<td>27374</td>
<td>1209</td>
<td>13.3%</td>
<td>13.3%</td>
<td></td>
<td></td>
<td>Unclassified locks</td>
</tr>
<tr>
<td>0</td>
<td>18726</td>
<td>0</td>
<td>8.7%</td>
<td>8.7%</td>
<td></td>
<td></td>
<td>Cache Services association main lock</td>
</tr>
<tr>
<td>310</td>
<td>0</td>
<td>6617</td>
<td>3.2%</td>
<td>3.2%</td>
<td></td>
<td></td>
<td>Async global device lock</td>
</tr>
<tr>
<td>5455</td>
<td>0</td>
<td>29</td>
<td>2.5%</td>
<td>2.5%</td>
<td></td>
<td></td>
<td>Vnode-cache access lock</td>
</tr>
<tr>
<td>1036</td>
<td>0</td>
<td>2019</td>
<td>1.4%</td>
<td>1.4%</td>
<td></td>
<td></td>
<td>Log system map lock</td>
</tr>
<tr>
<td>200</td>
<td>145</td>
<td>2485</td>
<td>1.3%</td>
<td>1.3%</td>
<td></td>
<td></td>
<td>User file cache main segment lock</td>
</tr>
<tr>
<td>2020</td>
<td>0</td>
<td>167</td>
<td>1.1%</td>
<td>1.1%</td>
<td></td>
<td></td>
<td>User-cache file lock</td>
</tr>
<tr>
<td>0</td>
<td>1858</td>
<td>5</td>
<td>0.8%</td>
<td>0.8%</td>
<td></td>
<td></td>
<td>Volser I/O queue lock</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>1250</td>
<td>0.6%</td>
<td>0.6%</td>
<td></td>
<td></td>
<td>Metadata-cache buffer lock</td>
</tr>
<tr>
<td>0</td>
<td>1176</td>
<td>0</td>
<td>0.5%</td>
<td>0.5%</td>
<td></td>
<td></td>
<td>OSI Global queue of threads waiting for locks</td>
</tr>
<tr>
<td>741</td>
<td>0</td>
<td>97</td>
<td>0.3%</td>
<td>0.3%</td>
<td></td>
<td></td>
<td>Anode bitmap allocation handle lock</td>
</tr>
<tr>
<td>378</td>
<td>0</td>
<td>449</td>
<td>0.3%</td>
<td>0.3%</td>
<td></td>
<td></td>
<td>Anode handle lock</td>
</tr>
<tr>
<td>68</td>
<td>278</td>
<td>440</td>
<td>0.3%</td>
<td>0.3%</td>
<td></td>
<td></td>
<td>Transaction-cache main lock</td>
</tr>
</tbody>
</table>

Total lock contention of all kinds: 215055

### Top 5 Most Common Thread Sleeps

<table>
<thead>
<tr>
<th>Thread</th>
<th>Pct.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2978</td>
<td>99.2%</td>
<td>User file cache Page Reclaim Wait</td>
</tr>
<tr>
<td>24</td>
<td>0.7%</td>
<td>Transaction allocation wait</td>
</tr>
<tr>
<td>0</td>
<td>0.0%</td>
<td>OSI cache item cleanup wait</td>
</tr>
<tr>
<td>0</td>
<td>0.0%</td>
<td>Directory Cache Buffer Wait</td>
</tr>
<tr>
<td>0</td>
<td>0.0%</td>
<td>User file cache Page Wait</td>
</tr>
</tbody>
</table>

**zFS Primary Address Space Storage Usage**

---

Total Bytes Allocated: 170012066 (166027K) (162M)
Total Pieces Allocated: 240925
Total Allocation Requests: 241635
Total Free Requests: 710

### Storage Usage By Component

<table>
<thead>
<tr>
<th>Bytes Allocated</th>
<th>No. of Pieces</th>
<th>No. Of Allocs</th>
<th>No. Of Frees</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>132842</td>
<td>35</td>
<td>35</td>
<td>0</td>
<td>USS Interface</td>
</tr>
<tr>
<td>843844</td>
<td>32932</td>
<td>32932</td>
<td>0</td>
<td>Media Manager I/O driver</td>
</tr>
<tr>
<td>33555876</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>Trace Facility</td>
</tr>
<tr>
<td>266156</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>Message Service</td>
</tr>
<tr>
<td>40244</td>
<td>46</td>
<td>46</td>
<td>0</td>
<td>Miscellaneous</td>
</tr>
<tr>
<td>7616</td>
<td>91</td>
<td>91</td>
<td>0</td>
<td>Aggregate Management</td>
</tr>
<tr>
<td>122454</td>
<td>189</td>
<td>191</td>
<td>2</td>
<td>Filesystem Management</td>
</tr>
<tr>
<td>9680</td>
<td>15</td>
<td>134</td>
<td>119</td>
<td>Administration Command Handling</td>
</tr>
<tr>
<td>5312804</td>
<td>39</td>
<td>39</td>
<td>0</td>
<td>Vnode Management</td>
</tr>
<tr>
<td>11647040</td>
<td>66639</td>
<td>66669</td>
<td>30</td>
<td>Anode Management</td>
</tr>
<tr>
<td>1074892</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>Directory Management</td>
</tr>
<tr>
<td>68723616</td>
<td>8384</td>
<td>8416</td>
<td>32</td>
<td>Log File Management</td>
</tr>
<tr>
<td>34405632</td>
<td>4275</td>
<td>4336</td>
<td>61</td>
<td>Metadata Cache</td>
</tr>
<tr>
<td>3304884</td>
<td>2227</td>
<td>2233</td>
<td>6</td>
<td>Transaction Management</td>
</tr>
<tr>
<td>3187988</td>
<td>9677</td>
<td>9679</td>
<td>2</td>
<td>Asynchronous I/O Component</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>159300</td>
<td>1096</td>
<td>1096</td>
<td>0</td>
<td>Lock Facility</td>
</tr>
<tr>
<td>2992</td>
<td>61</td>
<td>61</td>
<td>0</td>
<td>Threading Services</td>
</tr>
<tr>
<td>106236</td>
<td>35707</td>
<td>36156</td>
<td>449</td>
<td>Cache Services</td>
</tr>
<tr>
<td>21456</td>
<td>59</td>
<td>60</td>
<td>1</td>
<td>Configuration parameters processing</td>
</tr>
<tr>
<td>6101056</td>
<td>79395</td>
<td>79403</td>
<td>8</td>
<td>User File Cache</td>
</tr>
<tr>
<td>29368</td>
<td>42</td>
<td>42</td>
<td>0</td>
<td>Storage Management</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File System Name</th>
<th>Aggr #</th>
<th>Flg</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML96.ZFS.DO1MLC96</td>
<td>100000</td>
<td>AM</td>
<td>40803823</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC83</td>
<td>100023</td>
<td>AM</td>
<td>15118812</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.NOTESLOG</td>
<td>100018</td>
<td>AM</td>
<td>8499654</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.DIRCAT</td>
<td>100017</td>
<td>AM</td>
<td>3629239</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC83.NOTESLOG</td>
<td>100025</td>
<td>AM</td>
<td>3322012</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL7</td>
<td>100007</td>
<td>AM</td>
<td>2865519</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL9</td>
<td>100020</td>
<td>AM</td>
<td>2345923</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL10</td>
<td>100009</td>
<td>AM</td>
<td>1920709</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL12</td>
<td>100010</td>
<td>AM</td>
<td>1836175</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL14</td>
<td>100004</td>
<td>AM</td>
<td>1822233</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL15</td>
<td>100019</td>
<td>AM</td>
<td>1658708</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL16</td>
<td>100005</td>
<td>AM</td>
<td>1486696</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL17</td>
<td>100001</td>
<td>AM</td>
<td>1458093</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL18</td>
<td>100012</td>
<td>AM</td>
<td>1424410</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL19</td>
<td>100026</td>
<td>AM</td>
<td>1155690</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL11</td>
<td>100013</td>
<td>AM</td>
<td>1014333</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL12</td>
<td>100014</td>
<td>AM</td>
<td>825401</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL13</td>
<td>100015</td>
<td>AM</td>
<td>815730</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL14</td>
<td>100006</td>
<td>AM</td>
<td>693928</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL15</td>
<td>100002</td>
<td>AM</td>
<td>554402</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL16</td>
<td>100008</td>
<td>AM</td>
<td>538396</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL17</td>
<td>100011</td>
<td>AM</td>
<td>511061</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL18</td>
<td>100016</td>
<td>AM</td>
<td>409044</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.MAIL19</td>
<td>100024</td>
<td>AM</td>
<td>189904</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.OLDLOGS</td>
<td>100021</td>
<td>AM</td>
<td>107665</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.OLDLOGS</td>
<td>100022</td>
<td>AM</td>
<td>80396</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.OLDLOGS</td>
<td>100027</td>
<td>AM</td>
<td>5208</td>
</tr>
<tr>
<td>ML96.ZFS.DO1MLC96.OLDLOGS</td>
<td>100029</td>
<td>AM</td>
<td>6</td>
</tr>
</tbody>
</table>

IOEZ00025I zFS kernel: MODIFY command - QUERY,ALL completed successfully
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 200.

- **Lotus Domino for S/390 Release 5: Installation, Customization and Administration**, SG24-2083
- **Lotus Domino for S/390 Release 5: Running a Large Domino System**, SG24-5984
- **Debugging UNIX System Services, Lotus Domino, Novell Network Services, and other Applications on OS/390**, SG24-5613
- **OS/390 Version 2 Release 6 UNIX System Services Implementation and Customization**, SG24-5178
- **Hierarchical File System Usage Guide**, SG24-5482
- **Communications Server for z/OS V1R2 TCP/IP Implementation Guide, Volume 1: Base and TN 3270 Configuration**, SG24-5227
- **Communications Server for z/OS V1R2 TCP/IP Implementation Guide, Volume 2: UNIX Applications**, SG24-5228

Other resources

These publications are also relevant as further information sources:

- **OS/390 UNIX System Services Planning**, SC28-1890 (edition 07), Chapter 16, “Tuning performance”
- **S/390 PR/SM Planning Guide**, GA22-7236
- **OS/390 MVS System Management Facilities (SMF)**, GC28-1783
- **z/OS V1R4.0 Distributed File Service zSeries File System Administration**, SC24-5989
- **z/OS: UNIX System Services Command Reference**, SA22-7802
- **OS/390 SecureWay Communications Server IP User’s Guide**
- **z/OS RMF User’s Guide**, SC33-7990
- **z/OS RMF Report Analysis**, SC33-7991
- **z/OS RM Performance Management Guide**, SC33-7992
- **z/OS MVS System Management Facilities (SMF)**, SA22-7630
- **z/OS MVS Initialization and Tuning Reference**, SA22-7592
- **OSA Planning**, GC23-3870
IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide, SC32-9056


Referenced Web sites

These Web sites are also relevant as further information sources:

- http://www.eclipse.org
- http://www.share.org
- http://www.lotus.com/idd
- http://www.networking.ibm.com
- http://www.as.ibm.com/asus/

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.
Index

Symbols
/notesdata 72
/notesdata/mail 72
/usr/lpp/lotus 72

Numerics
3172-3 Interconnect Controller 82
3174 Networking Server 82
3380 56
3390 model 3 66
3745/3746 Communications Controller 82
64 bit real memory addressing 101, 115
64-bit real memory addressing 33
9672 146

A
access level 106
ACL
entries 5
activatedbs command 100
active 15 minute users 118–119, 147, 160
active users 148, 157
activity
    cache 57
    DASD 57
add-in server process 51
address book improvements 10
address space 23–24, 32, 48, 103
    name 26
reusing 25
addressing, virtual IP (VIPA) 84
administrator 3
Administrator client 110
administrators
    what’s new 12
Advanced Services, Domino 119
AF_INET sockets 81
agent
    improvements 17
aggregate 70
analysis program 144
analysis tools 52
analyzer for Lotus Domino 7
APAR OW44631 64
APPL% 47
application design 15
archive file 72
assistance, capacity planning 158
ASSORTEDPARMS 80
availability 119
average workload 149

B
background 25
backup 2
    reduced time 95
billing 88
bottleneck 101
BPX parameters 89
BPX.SMF FACILITY class 128
BPXPRMxx 5, 24, 28, 35, 40–41, 64, 81, 89, 91, 150
browser, Web 88
buffer pool size 64
buffer, TCP/IP 79

C
C/C++ 36
cache hit rate 66
cache hit ratio 69
caching 36, 122
    activity 57
    DASD 56
    Domino 93
    HTTP 94
    performance 49
calendar and scheduling improvements 10
CALLS390 158
capacity 28
    planning 1, 145
    recommendations 145
    processor 30
    estimating 147
    sizing 158
capacity planning 150, 159
capping, LPAR 45
case, program name 27
CEEDUMP 74
central storage 152
channel
    path 58
    utilization 56, 58
channel paths
    monitoring 58
channels
    OSA 58
characteristics, Domino 2
Chronos 106
CICS 2, 57, 150
Cladmin 120
class
    report 166
    service 30, 165
classification rules 167
cldbdir.nsf 120

© Copyright IBM Corp. 2003. All rights reserved.
client
  replication 105
  Web 88
client-server 75, 88
Clrepl 120
CLUSTA4.NSF 120
cluster administration process 120
Cluster Database Directory Manager 120
clustering 83, 119
  alternatives 122
  components 120
  cost of 120
  monitoring 121
  performance 119
  replication 120
CMD field 27
cmd
  show tasks 103
  statistics 144
Compact 99
command
  show tasks 103
  statistics 144
configuration
  configuration settings document 110
  connected users 118, 148, 151
connections, user 26
console support 8
contention 26
Copytree 59, 66
cost of partitioning 115
CPU
  busy 40
  resource 115, 120
  time 115, 120, 146, 149
  utilization 39–40, 52, 76, 107
crash 74
CSA 82
CSM buffer pool 82

D
D A.L. 26
D GRS,C 56, 65
D OMVS,A=ALL 27
D OMVS,U=userid 27
DASD
  3380 56
  activity 57
  administrator 3
  average response time 40
  fast write (DFW) 56
  I/O rate 145, 157
  monitor 57
  performance 55
  recommendations 56
  requirements 157
  response time 39, 57–58
  response times 163
  space 109, 115, 121, 145, 157
  tuning 3, 55
  DASD cache
    monitoring 59
  DASD devices
    monitoring 61
data
  backup 2
data set
  HFS 56
database
  buffer pool maximum 92
  buffer pool reads 92
  caching 94
  index 106
  mail 157
  restore 100
  size 142
  statistics and reporting 143
  usage 142
  view 106
dataset
  HFS 56
  paging 34
dataspaces 24
DB2 2, 57
dbcache 94
DBIID 96, 109
dbcache 94
definitions, workload manager 165
delete document 108
deletes
  soft 99
demand paging 40
designer access 106
designers/developer
  new features 15
DFSMS 56, 64
directory and LDAP enhancements 20
disconnect time 61
disk space 109, 115, 121, 157
disks
  shared access 2
dispatching priority 114
document
  delete 108
document improvements 10
dom_verify_os 5, 24, 28
dom_verify_os command 89
DOMAIN_NAME, NETWORK 81
Index

DOMCON 8

Domino

Administrator client 110
administrators
new features 12
Advanced Services 119
caching 93
capacity
network 157
partitioned servers 155
characteristics 2
clustering 83
configuration 113
efficient 118
DASD requirements 157
Designer
view logging 98
designers
new features 15
HTTP server 111
initialization parameters 89
interfaces to z/OS 23
lock manager 97
mail.box 110
monitoring 127
network statistics 84
parameters 87
partitioned server (DPAR) 28, 30, 145, 156
partitioning (DPAR) 114
recommendations 88
server tasks 102
sizing help 158–159
statistics 7, 50, 143
processing 144
tasks 103
team 3
transaction rate 148
tuning 87
virtual storage allocation 93
workload 1
Domino 6 for z/OS
what is new 1
Domino parameters 93
Domino partition (DPAR)
definition 28
domps command 32, 142, 171
DPAR 114
multiple levels 7
DPAR (see Domino partitioned server) 156
DPARs
per LPAR 119
same release level 114
DPOOLSIZE 90
DSAPI filter 6
Dynamic Alias Management 31

engines
multiple 105, 108
single 105, 108
Enterprise 56
Enterprise Storage Server (ESS) 56
Storwatch 57
ESCON channel 56, 58
ESQA 29, 36, 152
ESS 56
PAV feature 31
estimating
processor capacity 147
processor storage 150
resources 146
event 143
database 143
task 143
events4.nsf 143
example, workload manager 165
expanded processor storage 2
export 144
extended link pack area (ELPA) 36
Extended Remote Copy (XRC) 123

F
factor, peak load 149
failover 120
fault ratio 69
fault recovery 5
fiber links 122
file
activity 49
performance 49
system lookup 62
file system 64
multivolume 66
reorganizations 66
file system space
monitoring 64
FILESYSTYPE TYPE(HFS) 150
fixed buffers 64
FIXED MAX 29, 82
fixed storage
DFSMS 151
FIXUP utility 88
flashes 82
flexibility 28
FORKCOPY 29
free space 64
APAR OW44631 64
fsync
operations 71
request 64
FTP 81
full text indexing 106–107

G
gateway 84
global network 104

ECSA 29
ECSA MAX 29, 82
ELPA 35–36
goals 168
GRS 56
GRS latch contention 110

H
Hardware Management Console (HMC) 36
HFS
data set 49, 56
planning 157
shared 65
hierarchical file system (HFS) 63
multi-volume file system 66
hit rate 66
hit ratio 70
host 83
HTTP 111
cache 94
database 2
engine 3
server 111
task 111
httpssetup 5

I
I/O
activity 121
performance 49
priority management 31
priority queueing 56
queue 57
rate 57
response time 55, 57–58
I/O queueing
monitoring 58
I/O Queuing Activity report 58
I/O tuning 56
IBM 3390 model 3 66
IBM Enterprise Storage Server (ESS) 56, 66
IBM Tivoli Analyzer for Lotus Domino 7, 52
IBM Tivoli Storage Manager for Mail
Data Protection for Lotus Domino 101
ICF number 44
ID, user 26
IDCAMS LISTDATA 59
IEFUSI 76
exit 76
IMAP 146, 151
performance 153
IMAP clients
storage 153
importance 168
improved performance 98
IMS 2, 57, 150
independent, platform 2
index
full text 107
indexing 88, 106
tasks 108
INFOAPARS 76
initialization parameters, Domino 89

iNotes
Web Access 6 clients 3
iNotes clients
storage 154
instancename 51
instancenum 51
Intelligent Resource Director 37
intermediate archive file 72
Internet
Cluster Manager (ICM) 120
interval 48
recording (SMF) 47–48
synchronization 48
interval recording 49
INTVAL 48
IP
network 82
virtual addressing (VIPA) 84
IPCSEMNIDS 29
IPCSEMNOPS 29
IPCSHMMPAGES 29, 91
IPCSHMNIDS 29
IPCSHMNSEGSEGS 29
IPCSHMSPAGES 29

J
Java
runtime environment 5
job
name prefix 27
scheduler 2
jobname, duplicate 25

K
kernel activity report 40
kerninfo 62

L
latch contention 65, 110, 162
latch mechanism 67
libnotes 35
link pack area (LPA) 36
linkage conventions 31
LISTDATA 59
load factor, peak 149
load http 111
load stats 144
local access 106
lock manager
Domino 97
log 142
Notes 84, 106
statistics and events 143
transaction 94
log file cache 70
log.nsf 84, 105, 121, 142
Log_Replication 105, 143
Log_Sessions 105, 143
logical CPs 36
logical partitioning (LPAR) 28, 43, 114
lookup, file system 62
Lotus Enterprise Integrator (LEI) enhancements 18
low utilization effect 147
lower case 27
LPA 36
LPAR 28, 114, 156
RMF report 43
LPARs per processor 119

M
mail 105
database 157
routing 142
workload 32
mail directory 74
mail file
standard template 99
mail improvements 9
mail.box
multiple 110
MailCluster_Failover 120
manager access 106
marks
unread 109
MAXASSIZE 29
MAXCPUTIME 29
MAXFILEPROC 29
maximum
transactions 88
users 118
MAXQUEUEDESIGS 29
MAXSHAREPAGES 29
MAXSOCKETS 29, 81
MAXTHREADS 29
MAXTHREADTASKS 29
MAXUSER 29
measurements 146
memory 2, 26
processor 32
metadata
cache 70
pointers 66
migrating data 72
migration age 45
MONITOR-II, RMF 82
monitoring 3, 92, 127
channel paths 58
collection activity 82
DASD 57
cache 59
devices 61
Domino 127
file system space 64
I/O queueing 58
network connection 82
NSF buffer pool size 92
replication 105
RMF 57
statistics 143
TCP/IP 76
z/OS 38
mount 28
multiple
allegiance 56
Domino servers 28
engines 105, 108
processors 28
threads 28
multiple DPAR levels 7
multiple instances
of Domino in same LPAR 7
multiprocessor 105, 108
multivolume file system 66
MVS
virtual IP addressing (VIPA) 84

N
name
address space 26
program 27
nested calls 31
NETSTAT command 76, 79
network
administrator 3
capacity 145
connection by zSeries 82
collection, monitoring 82
DOMAINAME 81
global 104
interfaces 83
performance 83
splitting the network 83
planning 157
protocol 75
recommendations 76
specialist 104
splitting 83
statistics 84
statistics, domino 84
traffic 104, 121, 158
tuning 3, 75
network tuning 75
networking 115
Notes
administrator 3, 88, 106
client 151
number of users 146
log 84, 106, 127, 142
mail.box 110
Notes clients
storage 151
notes.ini 25, 28, 88–89, 91, 103, 105, 118, 133, 177
parameter changes 90
sample 177
Notes_SHARED_DPOOLSIZE 90–91
NSD dump 5
NSD log 74
NSF 91–92
NSF buffer pool size 92
NSF_BUFFER_POOLSIZE_MB 151
number of users 143, 146, 148, 156

O
OBEY statement 77
ODS (on-disk structure) 109
OMPROUTE daemon 84
OMVS
  kernel activity report 40
OMVS KERN 31
Open System Adapter (OSA) 82
OS/390
  image 28
  memory 24
  monitoring 3
  parameter 28
  performance specialist 3
  recommendations 24
  Security Server (RACF) 128
  system programmer 104
  tuning 24
  UNIX System Services 2, 8
OSA 82
  card 83
  device 83
OSA channels 58
OSA Express adapter 115
OSPF dynamic routing protocol 84
overruns 40–41

P
page datasets 34
paging 24, 32, 93, 150
  data sets 34
  demand 40
  RMF report 45
  storage 152
  use 39
Parallel Access Volumes (PAV) 56, 66
parallel channel 56, 58
parameter
  BPX 89
  clustering 120
  Domino 87
  Domino initialization 89
  OS/390 28
  recommended value 28
  settings 3
  Web server 111
parameters
  Domino 93
  PARMLIB 91, 150
partition data report 43
partitioned server 93
partitioning 114
  benefits 114
cost of 115
  Domino 28, 30, 156
  resource 115
  server 156
path
  activity 57
  channel 58
PAV 56
  PAV feature of ESS 31
  pax command 72
  peak
    load factor 149
    time 88
    transaction rate 143
    workload 150
Peer-To-Peer Remote Copy (PPRC) 122
Perform Locked Operation 146
performance 1, 107
  clustering 119
  constraining 3
  goals 168
  group 46
  I/O 49
  improvement 98
  ESS 57
  zFS 67
  monitoring 3
  network 83
  specialist, OS/390 3
  test 146
performance management 159
physical CPs 36
physical thread 26, 91
pilot 149
PING command 76
planning
  assistance 158
  capacity 145
  DASD 157
  HFS 157
  network 157
  storage 150
platform
  independent 2
  statistics 6, 50, 144
    gathering 52
PLO 146
POOL definition 82
POP3 146, 151
  client storage 152
Poughkeepsie 6
PPRC Extended Distance (PPRC-XD) 123
PR/SM Logical Partition (LPAR) 28
priority 24, 30
  dispatching 114
  TCP/IP 81
procedure calls 94
processor
  capacity 30
  estimating capacity 147
  storage 115, 150
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>processor storage</td>
</tr>
<tr>
<td>Notes clients 151</td>
</tr>
<tr>
<td>processors, multiple 28</td>
</tr>
<tr>
<td>production mode 152</td>
</tr>
<tr>
<td>profile, TCP/IP 77</td>
</tr>
<tr>
<td>program</td>
</tr>
<tr>
<td>name 27</td>
</tr>
<tr>
<td>programming 15</td>
</tr>
<tr>
<td>@formulas and @commands 16</td>
</tr>
<tr>
<td>Java 17</td>
</tr>
<tr>
<td>JSP tag libraries 18</td>
</tr>
<tr>
<td>Lotus Connectors 17</td>
</tr>
<tr>
<td>LotusScript 17</td>
</tr>
<tr>
<td>XML 16</td>
</tr>
<tr>
<td>protocol</td>
</tr>
<tr>
<td>network 75</td>
</tr>
</tbody>
</table>

| Q |
| queue, I/O 57 |
| quotas 99 |

| R |
| RACF |
| authentication 6 |
| overrides 89 |
| rate, transaction 142, 148 |
| read ahead 68 |
| real memory 24 |
| real-time replication 119 |
| recommendations 2 |
| capacity planning 146 |
| DASD 56 |
| Domino 88 |
| network 76 |
| OS/390 24 |
| recovery 94 |
| Redbooks Web site 200 |
| Contact us xii |
| REGION 76 |
| TCP/IP 81 |
| region 81 |
| size 76 |
| registered users 118, 148 |
| Replica.Cluster.WorkQueueDepth 121 |
| replication 88, 104, 142 |
| client 105 |
| frequency 105 |
| monitor 105 |
| real-time 119 |
| servers 104 |
| replicator task 105 |
| report |
| class 46, 166 |
| LPAR 43 |
| performance group 46 |
| summary 39 |
| reporter 143 |
| requirements, Domino 2 |

| resource |
| contention 26 |
| estimate 146 |
| indexing 107 |
| partitioning 115 |
| Resource Measurement Facility (RMF) 38, 76 |
| RMFMON (Monitor II) 38 |
| RMFWDM (Monitor III) 38 |

| resource usage |
| compact 109 |
| response time 24, 30, 46, 168 |
| DASD 57–58 |
| I/O 55, 58 |

| restart |
| fast 121 |

| restore |
| databases 100 |
| reduced time 95 |
| restricted, server 120 |
| RMF 57 |
| cache subsystem activity 59 |
| cache subsystem activity report 59 |
| I/O Queuing Activity report 58 |
| LPAR report 43 |
| MONITOR-II 82 |
| monitoring DASD 57 |
| server report 137 |
| summary report 57 |
| RMF Monitor II 6 |
| RMF MONITOR-II 82 |
| RMFMON (Monitor II) 38 |
| RMFMONITOR 82 |
| RMFWDM (Monitor III) 38 |
| router 110 |
| multiple threads 110 |
| routing, mail 142 |
| rules, classification 167 |

| S |
| S/390 |
| hardware 8 |
| logical partitioning (LPARs) 145 |
| scalability 2, 119 |
| scaling 101 |
| scheduled task 88, 104, 107 |
| security 114 |
| enhancements 21 |
| server |
| configuration 113 |
| configuration document 90, 93 |
| console traffic 105 |
| Domino tasks 102 |
| partitioned 93 |
| partitioning 114 |
| process 25 |
| replication 104 |
| shell structure 100 |
| tasks 102 |
| Web 110 |
server load record 149
server tasks 89
Server.Trans.PerMinute.Peak 143
Server.Users 143
Server_Cluster_On 120
SERVER_ENABLE_THREADPOOL 91
Server_Max_Concurrent_Trans 88
Server_MaxSessions 88
Server_MaxUsers 118
SERVER_POOL_TASKS 91, 133, 150
Server_Restricted 120
Server_Session_Timeout 148
ServertasksATx 107
service
class 30, 46, 165
rates 47
service class 30, 46
serviceability 5
services, testing 159
set configuration command 90
SEZAINST sample library 76
shared CPs 36
shared HFS 65
shared library space 35
Shared Library Support 7
shared memory 90
SHARED_DPOOLSIZE 90
sharing 115
show cluster 120
show stat command 94
show statistic 111, 121, 144
sample 181, 193
show statistic replica 121
show tasks 103, 121
simple tests 3
single engine 105, 108
sizing 2
Domino 158
sizing your system 158
SMF 6, 47, 61, 149
type 108 record 91, 128
type 30 record 48, 62
type 42 record 62
type 92 record 50, 62
SMF 108
server load 149
SMF records
types 61
SMF type 108 record
enabling 128
field mappings 128
SMF30AIS 49
SMF30CPS 49
SMF30CPT 49
SMF30EXN 49
SMF30JBN 49
SMF30OFR 49
SMF30OFW 49
SMF30OPR 49
SMF30OPW 49
SMF30OSR 49
SMF30OSW 49
SMF30PGM 49
SMFPRMxx 50, 128
SNAP/SHOT capacity planning tool 159
socket descriptor 26
soft deletes 99
space
DASD 115, 157
unused 108
splitting the network 83
Spool Display and Search Facility (SDSF) 76
SQA 82
SRB 47
SSCH 49
Start Subchannel 49
started task 127
statistics 143
and events log 143
and reporting database 143
command 144
database 143
Domino 7, 50, 143
processing 144
monitoring 143
network 84
platform 6, 50, 144
gathering 52
TCP/IP 80
UNIX 62
statrep.nsf 143
storage
64-bit 33
cost 118
extended private 32
have enough 24
IMAP clients 153
iNotes Web Access clients 154
Notes clients 151
POP3 clients 152
processor 150
use 39, 48
virtual 32
Storwatch Enterprise Storage Server Expert 57
stress testing 159
summary report, RMF 39, 57
symmetric multiprocessors (SMPs) 2
sync daemon interval 64
csync interval 67
SYNCVAL 48
SYS1.PARMLIB 24, 28
BPXPRMxx 150
Sysplex 29
SYSRPTS 46
System Data Mover 123
System Management Facilities (SMF) 38, 47
T

tasks
  Domino  25, 88
  non-peak times
    adminp  103
    billing  103
    stats  103
    update  103
  replicator  105
  scheduled  88, 104, 107
    catalog  103
    compact  88
    fixup  88
    index  88
    updall  88
    update  88
  show  103
  TCB (see also task control block)  47

TCP/IP  28, 75–76
  AF_NET sockets  81
  buffer  79
  dispatching priority  81
  monitor  76
  multiple copies  84
  priority  81
  profile  77
  region size  76, 81
  same stack  115
  session timeout  30
  stack  81
  statistics  80
  tuning  81

TCPIPSTATISTICS  76, 80

team  3
  Domino  3
  teamwork  3
  TechXPress  158
  Telnet  81
  telnet  81
  testing
    performance  146
    services  159
    simple  3
    stress  159
  thread  25, 91, 103
    HTTP server  111
    multiple  28
    physical  91
  threadpool  153, 162
  time
    CPU  146, 149
    peak  149
    response  168
    time zones  150
    time-out  30
    timing out  24
  tips, OS/390 UNIX Services  24
  Tivoli Analyzer for Lotus Domino  7, 52
  Tivoli Data Protect for Domino  100
  Tivoli Data Protection tool  109
  to do improvements  11
  tools
    analysis  52
    total cost of ownership (TCO)  8
    TRACERTE command  76–77
    traffic, network  104, 158
  transaction
    maximum  88
    rate  146, 148
  transaction log  66, 74, 94
    benefits  94
    DBIID  96
    disk allocation  101
    enabling  96
    extent files  100
    lock manager  98
    performance test results  101
    recommendation  101
    view elements  98
  transaction logging
    setup  96
  transaction rate  148
  trash
    view  99
  tuning  1, 3
    BPX parameters  89
    DASD subsystem  55
    Domino  87–88
    HFS  64
    I/O  56
    information  81
    network  75
    OS/390  24
    parameters  89
    TCP/IP  81
    zFS  68
    type 30 SMF record  62
    type 42 SMF record  62
    type 92 SMF record  62

U
  UIC  157

UNIX
  file  50
  history  2
  statistics  62
  System Services  2
  unmount  28
  unread marks
    disabling  109
  unreferenced interval count (UIC)  45
  unused space  108
  updall  88, 106–107
  update  88, 106
  update task  106
  Update_Suppression_Limit  108
  Update_Suppression_Time  108
  upgrading  119
  upper case  27
  useful Domino statistics  143
user 68
  active 15 minutes 118
user connections 26
user file cache 26
user ID 68
users
  active 148, 157
  connected 118, 148, 151
  maximum 118
  number of 143, 146, 148, 156
  per DPAR 118
  registered 118, 148
utilization 40
utilization, channel 56, 58

V
velocity 30, 46, 168
Verity KeyView filter 7
view logging 98
view, database 106
VIPA 115
virtual
  storage 32
  thread 26
    ID 26
virtual IP addressing (VIPA) 84
virtual storage 93
  DFSMS 151
vnode cache 71

W
wasted space 108
Web
  browser 88
  improvements 18
  server 110–111
    parameters 111
Web Connector 6
Webmail 67
WebSphere plug-in 6
weight, LPAR 44
working set 32
workload
  average 149
  Domino 1
  mail 32
  manager
    goal mode 46
    peak 150
    RMF report 46
workload balancing 119
Workload Manager 30

X
X.509
  certificate 6
XPLink 31
Lotus Domino 6 for Z/OS: Performance Tuning and Capacity Planning
Lotus Domino 6 for z/OS: Performance Tuning and Capacity Planning

What’s new in Domino 6

Performance tuning on z/OS systems

Performance management and capacity planning

This IBM redbook helps z/OS system programmers and Domino administrators to monitor and tune Domino 6 for z/OS. It identifies factors that affect performance and provides recommendations to tune the configuration and parameter settings for optimal performance. Tips are provided for the operating system, the disk I/O subsystem, the network, and the Domino product. In addition, it includes guidelines for capacity planning. A background in z/OS and experience with Domino is assumed.

This version, updated for Domino 6, includes tips on the zFS file system and new monitoring tools and techniques.