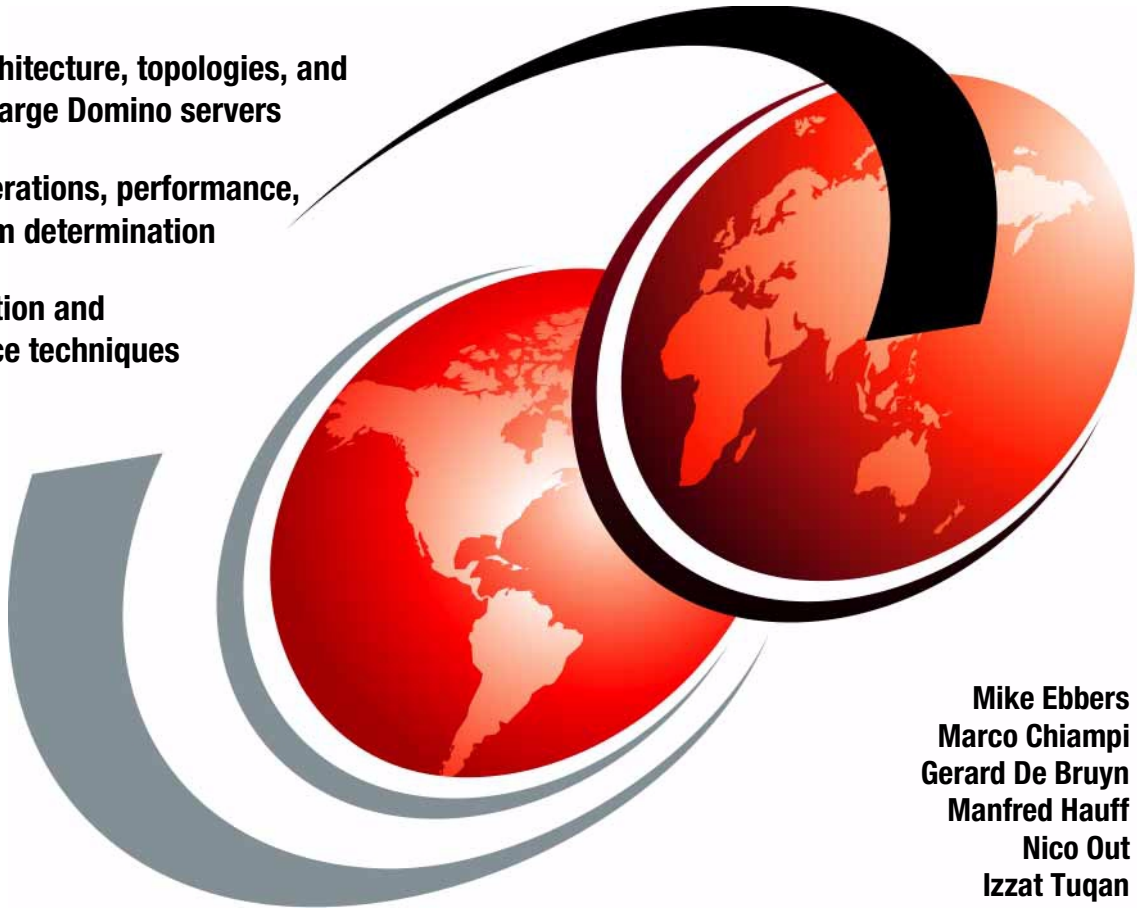
Lotus

IBM

Lotus Domino for S/390:

Running a Large Domino System

- Sample architecture, topologies, and design for large Domino servers
- Tips for operations, performance, and problem determination
- Administration and maintenance techniques



Mike Ebbers
Marco Chiampi
Gerard De Bruyn
Manfred Hauff
Nico Out
Izzat Tuqan

Redbooks

ibm.com/redbooks

SG24-5984-00

International Technical Support Organization

Lotus Domino for S/390: Administering a Large Domino System

February 2001

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special notices” on page 189.

First Edition (February 2001)

This edition applies to Domino for S/390 V5R5.04a, Program Number 5655-B86 for use with the OS/390 V2R8 Operating System.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This IBM Redbook describes how to run a large Domino for S/390 production system from the perspective of both S/390 and Notes administration. It documents the experience of running the world's largest Domino/Notes e-mail installation at IBM Poughkeepsie, as well as the experiences of other consultants and customers from around the world. You will find answers to questions commonly asked by customers, with solutions that are valid for any Domino platform.

Part 1 of this book describes the architecture, topologies and design of a large Domino system, focusing on the organizational aspects and challenges of running Domino on OS/390.

Part 2 reveals the secrets of operating, administering and monitoring a large Domino system on OS/390. It describes backup and recovery possibilities, problem determination, maintenance of software, as well as hints and tips. Sample shell scripts, Notes API programs, and LotusScript agents are also provided to assist you with administration.

This book is intended for Domino administrators, OS/390 system programmers, and operators responsible for installing, configuring, operating, and administering a Domino system on OS/390.

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 on Domino for S/390. He has worked at IBM for 27 years. Before joining the ITSO six years ago, Mike worked in IBM US education as a curriculum developer and instructor.

Marco Chiampi is an IT Specialist at a large bank in Brazil. He has 9 years of experience in the UNIX field. His areas of expertise include Domino, TCP/IP networks, EDI and e-commerce.

Gerard De Bruyn is a Domino Administrator at the ABN-AMRO Bank in the Netherlands. He has 12 years of experience in PC LAN networks, programming and support, and 4 years of experience in the Domino

Administration field. He has worked at the ABN AMRO Bank for 4 years. His areas of expertise include Networking Architecture, Client Support, NT and Novell Server support.

Nico Out is an IT designer at the ABN-AMRO Bank in the Netherlands. He has 25 years of experience with OS/390, and 4 years of experience with Notes Application and Systems Administration. He has worked at IBM for 9 years. His areas of expertise include OS/390 System Programming, Notes application development, and Notes system administration

Izzat Tuqan is an OS/390 Operations Analyst at the ABN-AMRO Bank in the Netherlands. He has 10 years of experience in the OS/390 field. He holds a degree in Dutch law from Leiden University. His areas of expertise include OS/390 Operations Support and ITIL.

Manfred Hauff is an IT Architect working for IBM ITS in Mainz, Germany. He has 12 years of experience in supporting OS/390 software and has worked for 2 years in Lotus Domino S/390 EMEA second level support. His areas of expertise include OS/390 SCP, TSO and ISPF, UNIX System Services, e-business, and Web applications.

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

Derek Bambauer
Lotus Professional Services, Cambridge, MA

Joseph Bostian
IBM Server Group, IBM Poughkeepsie

Cheryl Christopher
MHVPL Storage Management, IBM Poughkeepsie

John Duffy Jr.
S/390 Storage & Server Solutions, Waltham MA

Barbara Filippi
IBM Lotus Integration Center, Dallas

Jim Golding
MVS Support, IBM Poughkeepsie

Sreehari Haridevara
OS/390 Lotus Notes Administration,IBM Poughkeepsie

Paul-Robert Hering
ITS MVS Software Support, IBM Germany

Mark Holmans
Suncorp Metway Ltd, Australia

Bill Kilduff
Lotus Professional Services, New York

Mike Lang
Technical Specialist - Domino/390, IBM Dallas

Gordon Palin
Domino/390 and e-Business on S/390, Toronto

Bucky Pope
Lotus Notes Performance, IBM Watson Reseach, Hawthorne NY

Elsie Ramos
OS/390 Lotus Notes Administration, IBM Poughkeepsie

Ray Schaff
Brian Sullivan
Domino for S/390 Development, IBM Poughkeepsie

Hans van Eif
AT&T Global Network Services, Netherlands

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 215 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Part 1. Architecture, topology, and design

Part 1 describes the architecture, topologies, and design of a large Domino system, focusing on the organizational aspects and challenges of running Domino on OS/390.

Chapter 1. Introduction

This chapter defines the scope of this redbook: what it is about, what is in the book, and what is not in the book. We also introduce the components of a Domino server.

1.1 What this book is about

This redbook is about administering a large Domino domain on OS/390. It describes the best practices used in some of the largest real-life Domino installations to run Domino smoothly on a large OS/390 system,

1.2 What is in this book

This redbook tells you how to design and administer a large Domino domain for S/390. It contains information for S/390 operators, system programmers, and Notes administrators on how to create, enlarge, and maintain your servers. We include “best practices” advice, as well as answers for commonly-asked administration questions.

1.3 What is not in this book

Topics covered in other redbooks are not included in this redbook. We refer you to the other redbooks for more details.

1.4 Server architecture

The current release of Lotus Domino is Release 5. Figure 1 on page 4, shows the functions that R5 provides.

For a more in-depth view of the Domino server architecture, see *Lotus Domino for S/390 Release 5: Problem Determination Guide*. SG24-2083, Chapter 2.

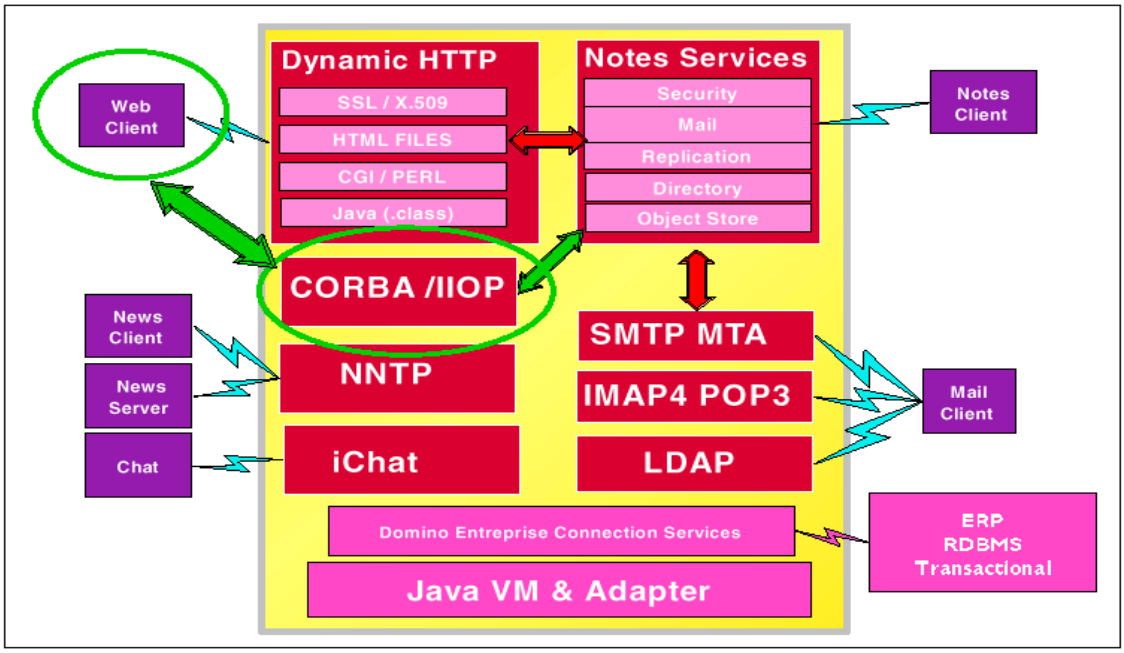


Figure 1. Function of Domino Release 5

The main component of Domino is Notes services. It is a client-server system based on its own remote procedure call mechanism called Notes Remote Procedure Call (NRPC). Lotus Domino servers and Notes clients run on many platforms, from workstations to mainframes, but the basic structure of the Notes services function is the same. Any platform differences are transparent to the user and the application programmer.

1.5 Organizational responsibilities

Domino on the S/390 platforms brings together different environments, which were until now far apart. The S/390 environment, the Domino environment, and the UNIX environment, all with a history of their own. It is, therefore, absolutely essential for a successful implementation of Domino on S/390, that the people in these environments communicate and establish a mutual understanding of each others' backgrounds and methods of work.

1.5.1 Introduction to the backgrounds

This chapter addresses the background of each environment. Then we list areas where environments might overlap, which could cause conflicting

interests between the organizational groups. Finally, we provide recommendations on how to solve these conflicts.

1.5.1.1 OS/390 background

The OS/390 environment is highly automated. In most cases it is monitored from one central point, called the operations department.

The main objectives of the operations department are to keep the system up and running, monitor performance, and react to any incidents which might hamper the availability and performance of the system. In addition, OS/390 operators also start, stop, and restart tasks running in the system, and they restart (IPL) the entire system if needed.

Users are concerned about availability of the system, maintenance windows (downtime), and performance requirements. So they often work out a Service Level Agreement (SLA) with the operations department. Automation tools, such as System Automation for OS/390, stop and start the tasks after an IPL or after a period of housekeeping. In case of an abend (abnormal ending of a task), an automatic restart procedure can be invoked.

OS/390 uses system managed storage (SMS) as a highly automated storage management tool. SMS uses software programs to manage security, placement, migration, backup, recovery, restoration, and deletion of data sets. It is often set up such that an operator cannot deviate from its definitions.

A job scheduler task, such as Tivoli OPC, is very often used to run jobs at a predetermined time or when certain predefined conditions are met. This can be used in cooperation with System Automation for OS/390 to stop a started task before running housekeeping jobs, then starting the task again.

1.5.1.2 Domino background

On most distributed implementations of Domino, the operating platform (for instance, NT or AIX) is usually dedicated to the Domino server. Because of this, OS level access is often restricted to Domino administrators. Service Level Agreements (SLA) are, therefore, made by the Domino administration group, since they control the service end-to-end. The Domino administrator usually goes to the operating system of the server to accomplish the following tasks:

1. Perform maintenance on the OS, such as examining the log files or changing the IP address.
2. Access a database in local mode to modify the ACL, in case it becomes corrupted or was misconfigured (such as excluding anyone from having manager access).

3. Examine the contents of the notes.ini file.
4. In case of a server crash, examine the last messages on the Domino console for further investigation into the cause of the crash.
5. Reboot the Domino server after a crash.
6. Reboot the Domino server as a preventive measure, based upon monitoring information, before an actual crash takes place.
7. Reboot the Domino server to let a changed notes.ini parameter take effect (one that is only read during startup of the Domino server).
8. Reboot the server at OS level to clean up all Domino processes and free up all resources allocated by Domino after a crash.
9. Run Domino maintenance utilities, such as compact and fixup, in an offline mode.
10. Restore databases from tape backup.

Note: Tasks 5 and 6 can also be performed by an automated monitoring utility.

1.5.1.3 UNIX background

OS/390 provides not only traditional MVS services, but also a rich set of UNIX services which conform to open system specifications such as POSIX and XPG4. This allows existing UNIX applications to be ported to MVS and new ones to be developed on MVS. UNIX applications, such as Domino, therefore benefit from System/390 strengths, such as performance, DASD capacity, system availability, and security.

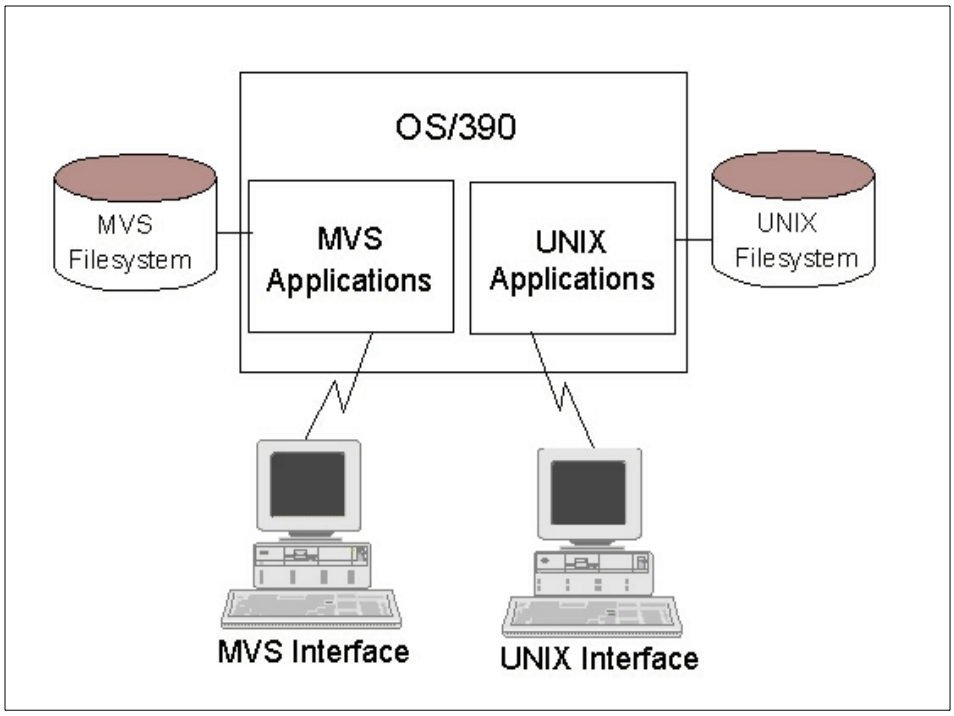


Figure 2. OS/390 and UNIX

Apart from UNIX services, which also include a UNIX-like hierarchical file system, OS/390 comes with a UNIX front-end that provides users with the UNIX 'look and feel', a collection of UNIX tools and utilities, and a versatile debugger.

UNIX System Services (USS) overview

OS/390 contains the UNIX System Services features as part of its base control program. UNIX System Services has an XPG4 Base Profile branding, allows UNIX applications to be executed on System/390.

UNIX System Services supports the recent set of standards being developed by the X/Open Company Limited and known as X/Open Portability Guides(XPG). The incorporation of these standards in USS offers:

- Program portability across multiple vendor operating systems
- A hierarchical file system (HFS) in an OS/390 environment
- Exchangeability between MVS data sets and the UNIX-style byte stream oriented file system

- A Language Environment (LE) which provides program interaction from MVS and UNIX programs
- Enhanced UNIX processes control for better system resource utilization and control
- A UNIX-like user interface
- Application threads
- New commands and capability for interacting with TSO/E
- Enhanced UNIX security features provided by RACF controls

X/Open is an independent, non-profit organization dedicated to making the idea of open systems a reality for the computer industry and users of information technology. X/Open currently involves major companies using technology, software vendor companies and system vendors through its associate membership program.

The X/Open Corporation Ltd. has defined a set of UNIX standards with the XPG4 Base specification and the XPG4.2 Single UNIX specification. Included in XPG4 are POSIX.1 and POSIX.2 standards. With OS/390, both the XPG4 Base specifications and 90% of the XPG4.2 Single specifications, have been integrated into OS/390, which now provides more than 1100 UNIX functions on a S/390 platform.

POSIX is a set of standards defined by the Institute of Electrical and Electronics Engineers (IEEE) on behalf of the American National Standards Institute (ANSI). POSIX defines a common interface across operating systems.

In the world of open systems, an enterprise using systems that conform to the standard interface can exchange or reuse applications, data, and skills on the conforming operating system of any manufacturer.

1.5.2 Conflicting interests

From the backgrounds mentioned above, you can see that when Domino is installed on a S/390, the areas where the S/390 and Domino administration groups have overlapping activities are:

- Emergency stopping and starting of the Domino server
- Scheduled stopping and starting of the Domino server
- Backup / Restore activities
- Monitoring and maintaining of the hardware and OS
- Setting up of SLAs
- OS level (UNIX) access

- Monitoring of application (Domino) behavior
- Application (Domino) maintenance

To avoid a situation where both groups claim they need to perform one or more of these activities, careful consideration needs to be made when deciding which activity is most applicable to which group. Before assigning responsibilities, the following questions should be considered.

- Can the operation be completed entirely from the Domino environment? If so, the Domino administrators are the most likely candidates for the operation.
- Can the operation be completed entirely from the S/390 environment? If so the S/390 administrators are the most likely candidates for the operation.
- Which group does the user community see as owner of the system. This is the group whom the user contacts in case of failure. Since a chain is only as strong as its weakest link, we recommend that all groups be consulted or at least notified before any “user noticeable” action is taken.

Based upon the previous considerations, assigning the tasks can be done as follows:

- **Emergency stopping and starting of the Domino server**
  This should be a joint effort between the Domino and S/390 admin groups. In case a Domino server is not functioning properly, the Domino group investigates the problem and notifies the S/390 group. If the Domino group identifies that the Domino server requires a restart, the S/390 group should be the ones to issue the commands from the MVS console to stop and start the server. Even though there are several ways to accomplish this outside the MVS console and let the Domino administrator issue the restart, you should consider these facts:
  - In the case of a crash, normal shutdown procedures often do not work. So the server might not release all resources, or not come down at all.
  - In determining whether Domino has ended normally and released all the address spaces, you must have the right skills and access level for OS/390. These are also required to fix any complications after a crash.
  - To determine the cause of a crash, often a dump is required for root cause analysis by IBM. This can only be generated from the MVS console.

  Obviously, good cooperation between the two groups is of vital importance, particularly in this situation. Communication should be as direct as possible to ensure a quick restoration of service.

- **Scheduled stopping and starting of the Domino server**
  If this is a requirement on an as-needed basis, such as a configuration change, then the Domino group issues a request with the S/390 group to have this done at an agreed time. When this is a requirement from the S/390 group, for instance to perform housekeeping, the S/390 group will have to consult the Domino group regarding if and when this can be done, and then the S/390 group can perform the restart. If this is a requirement on a regularly scheduled basis, such as an offline fixup of the directory, then this should be scheduled from the Domino environment by the use of a program document. Following are some guidelines to follow:
  - The program document should call a UNIX script that handles the procedure.
  - As already discussed in item 1, the preferred way to stop the Domino server is from MVS. Therefore, to bring the server down the script should make an MVS call to stop the Domino server. This can done, for example, by using the tool `oeconsole`. You would type `oeconsole 'S STOPDOMINO'` where STOPDOMINO is the name of your STC for stopping the Domino server. The tool `oeconsole` is available from the following Web site:

    `http://www.s390.ibm.com/unix`
  - To start the server again, the script can call the `rc.notes` UNIX script.

  By initiating the scheduled server restart from Domino through a program document, you can also develop scripts that perform the same function on platforms other than S/390. This way, the Domino administrators keep the overview of what is scheduled for all servers, regardless of the platform they run on.
- **Backup / Restore activities**
  Since this operation can be done entirely from the S/390 environment, it is recommended that the operations group be assigned to this activity.
- **Monitoring and maintaining the hardware and OS**
  Since this operation can be done entirely from the S/390 environment, it is recommended that the OS/390 system group be assigned to it. For any changes to the OS, such as the installation of new PTFs, the OS/390 group should inform the Domino group before making any changes, as this can also affect the Domino server.
- **Setting up SLAs**
  In the case of a mixed environment, the user community is usually unaware of the type of server on which their mail file or application resides. Even in an environment where Domino runs exclusively on the S/390, the user community looks at Domino as a whole, including the

non-S/390 related issues. Therefore, it is recommended that the Domino group creates the SLAs with the user community. For aspects that are not performed by the Domino group, (such as backup and restore), these should be made in cooperation with the S/390 group, but for the end-user visible activities, just the Domino group should be involved.

- **OS level (UNIX) access**
  Obviously, the S/390 group has OS level access. They are also the ones who can grant or deny access to certain individuals from the Domino group. Domino administrators require access to the Domino Data directory to be able to:
  - check or change the contents of the `notes.ini` file
  - change the ACL of databases outside the Domino server
  - remove corrupted full-text indexes
  - move databases around outside the Domino server

  It is recommended that only the individuals who perform these tasks be granted access to the Domino data directory. These individuals must have UNIX skills.

- **Monitoring application (Domino) behavior**
  The S/390 group should monitor the Domino server the same way they monitor any other started task. They can take advantage of additional tools which are provided with Domino, such as Domino Console support. When they see something going wrong, they should consult the Domino group before taking action.
  The Domino group should monitor the server the same way they would monitor Domino on any other platform. Regardless of who restarts the Domino server after a problem, the Domino group should inform the S/390 group before taking action. Any issues which are inside the Domino environment, such as ACL changes, replication problems, and mail routing problems, can and should, be handled by the Domino group alone.

- **Application (Domino) maintenance**
  Where maintenance does not require a server restart or upgrading of the Domino code, this can be done by the Domino group. If it does involve a server restart or an upgrade of the Domino code, it is still the responsibility of the Domino group, but the S/390 group must be informed before any changes are made. This is especially true when upgrading the Domino code, as this could also require some maintenance (PTFs) to be installed on parts of the OS/390 operating system.

1.5.3 General recommendations

A written agreement should be drawn up between the groups, specifying exactly who can do what. This agreement should not be static, but evaluated from time to time. Changes may be required if things work a little differently from what you planned.

Exchange knowledge between the groups. The Domino group should have an adequate understanding of what S/390 is all about. The S/390 group should have an adequate understanding of what Domino administration is all about.

The physical locations of the Domino and S/390 groups should not be too far apart, to ensure there is good communication between the two groups. Usually communication improves when people actually know each other and are not talking to an anonymous voice at the other end of a phone line.

To avoid conflict of interest, both groups should be part of the same department. This can reduce issues about budget, personnel, and so forth.

Synchronize the change processes. The two groups should make use of the same resources for entering and approving changes to their environment. When this proves impossible, both groups should at least give each other access to their individual change processes and make them part of the approval cycle.

1.6 Service level agreements

A service level agreement (SLA) is a written contract between the service provider and the customer (user). It documents criteria for the delivered service, such as availability and performance. Although in most organizations running OS/390, an SLA for the system as a whole is agreed upon, it might be wise to consider a separate SLA just for Domino. The reason is that Domino is often considered to be a very important product for which good performance is key. Also, Domino crosses boundaries in the production environment, which makes it unclear what criteria of which SLA are valid for Domino. Moreover, to the customer it should not make any difference on what platform Domino is running. Depending on your organization, a separate section in the OS/390 SLA concerning Domino can be agreed upon.

1.6.1 Domino SLA

In this section we state the subjects to cover in a Domino-only SLA with the end user. However, the details are very much dependent on your

organization, and general guidelines are difficult to establish. Note that only parameters which can be measured should be incorporated in an SLA.

Subjects to cover might include:

- Domino server availability (for example, 98% available on a 24/7 basis)
- Domino cluster availability (for example, 99,5% available on a 24/7 basis)
- Application availability (for example, 97%, cannot be higher than server/cluster availability).
- Mail delivery time (for example, 95% within 2 hours)
- Mail file restore time (for example, 90% of request within 2 hours)
- ID creation time (for example, within 24 hours)
- Problem response time (for example, 90% fix within 8 hours)
- Help desk wait time (for example, 95% less than 30 seconds)
- Directory lookup time (for example, 5 minutes)
- Password reset time (for example, 95% less than 2 hours)
- Group creation time (for example, 90% less than 2 hours)
- Mail usage (see 3.7.1, “SLA for usage” on page 57)

1.6.2 OS/390 SLA

This section describes the subjects that might be covered in an OS/390 SLA in relation to Domino running on the OS/390 system:

- CPU utilization for Domino
- Availability of OS/390 itself and Domino
- Backup and recovery time
- When to perform housekeeping
- How long the housekeeping takes
- What is included in the housekeeping
- Performance
- Problem solving time
- Maintenance (this might be included in an SLA because keeping up to date with your maintenance level is very important when running Domino in an OS/390 environment)
- Organizational responsibilities

Chapter 2. Topologies

The Domino design that we describe in this section is for large companies that have multiple administration groups managing their Notes environment. Figure 3 shows an example of how different regions might be defined.

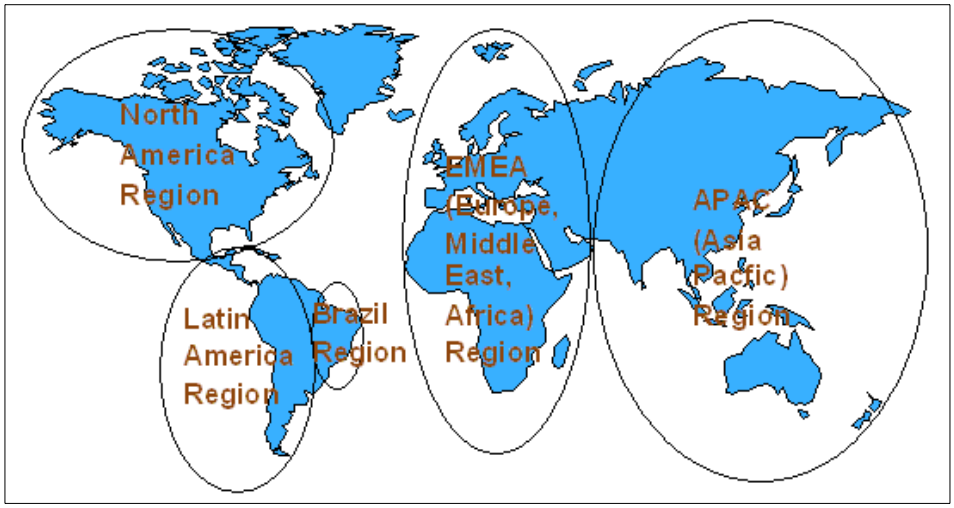


Figure 3. Regions

This design is based on the following situation:

- Each Domino administration group manages their own region.
- All regions use the same Domino domain for production.
- For communication with other Domino domains, there is a firewall domain.

The core infrastructure of the production domain described in this chapter consists of four major architectural elements.

- Domino directory infrastructure

  The directory infrastructure facilitates the availability of messaging and application functionality throughout the production and firewall domains. The directory infrastructure also provides information necessary to communicate over the Internet.

- Administration infrastructure

  Facilitates the management of the directory infrastructure.

- Domino mail routing infrastructure

  Facilitates the flow of messages within the production and firewall domains and the flow of messages between the domains. The message routing infrastructure also facilitates the flow of messages between the Domino domains and the Internet.

- Domino application infrastructure

  Facilitates the replication of applications.

The four elements just described are supported through the use of different functional types of Domino servers. They have been designed to support current and future business requirements. There are two types of servers:

- **Hub servers**: These servers are not accessible by the end user, only by Domino administrators. They are dedicated to tasks such as mail routing, replication, or administrative functions.
- **Spoke servers**: End users access this type of server. Often these servers also perform specialized functions such as mail or applications.

The server types can be defined as follows:

- AdminP server
- Regional administration server
- Directory hub server
- District directory server
- Routing hub server
- Regional application hub server
- Mail spoke server
- Application spoke server
- Combination mail and application server

A detailed description of these server types is in 2.1.1, “Directory infrastructure servers” on page 20.

2.1 Directory topology

To protect your production domain, you can install a separate firewall domain that handles all external communication to other Domino domains. Your production domain will have only a cross-certification with this external domain. All external parties that need to communicate with your production domain will only be cross-certified with the firewall domain.

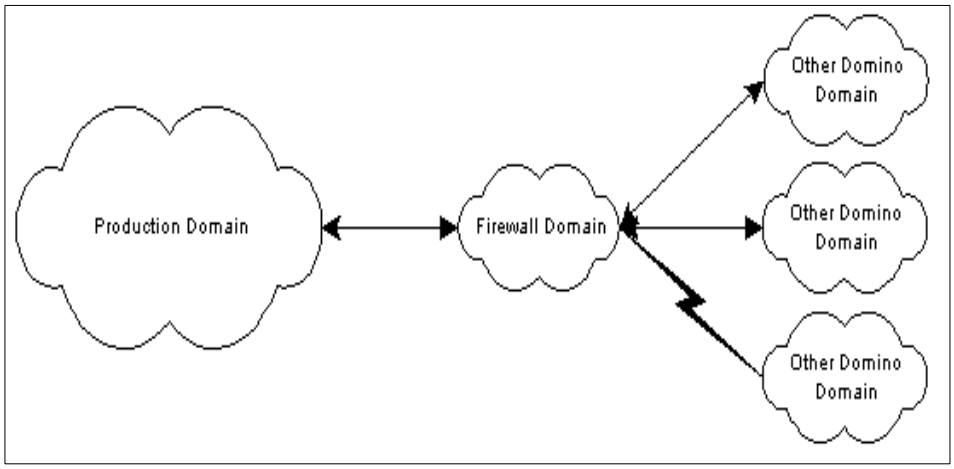


Figure 4. Domino domain structure.

The primary directory of the production domain is called the master directory. The master directory is core to the operation, security, and management of any Domino system. The directory infrastructure has, therefore, been designed to take advantage of all high availability, resiliency and scaling capabilities available on the Domino platform. The directory infrastructure goals are to provide directories which support the following functions:

- User authentication of proprietary Notes clients and Intranet-based open clients.
- Direct mail addressing to all mail users using keyword lookup or selection from a dialog box.
- Mail directory services to all supported clients, both Intranet-based and Notes.
- A minimal portable directory to provide mobile users with all users and groups.
- Scalability to well over 100,000 mail users with acceptable performance and stability.
- Resource reservation capabilities which meet the needs of the user community.
- Enterprise calendaring and scheduling that meet the needs of the user community.
- Resilient and redundant services that handle equipment and network failures.

- Distribution of HTTP password updates worldwide in a timely fashion, to provide the user community with uniform HTTP access.
- Enabling of Internet mail addressing and a valid return address in Internet format.

In order to deliver functions such as service availability (24 hours x 7 days), high performance delivery of directory functions, and a portable directory for mobile users, the directory infrastructure makes use of the following Domino R5 features.

- Domino clustering
- Domino directory assistance
- Domino directory catalog

By using these features, the service is also able to deliver Intranet directory services. However, it is not able to deliver 24x7 support using only Domino features. The service availability for Intranet clients is provided by standard networking fail-over solutions such as Network Dispatcher and multiple DNS entries.

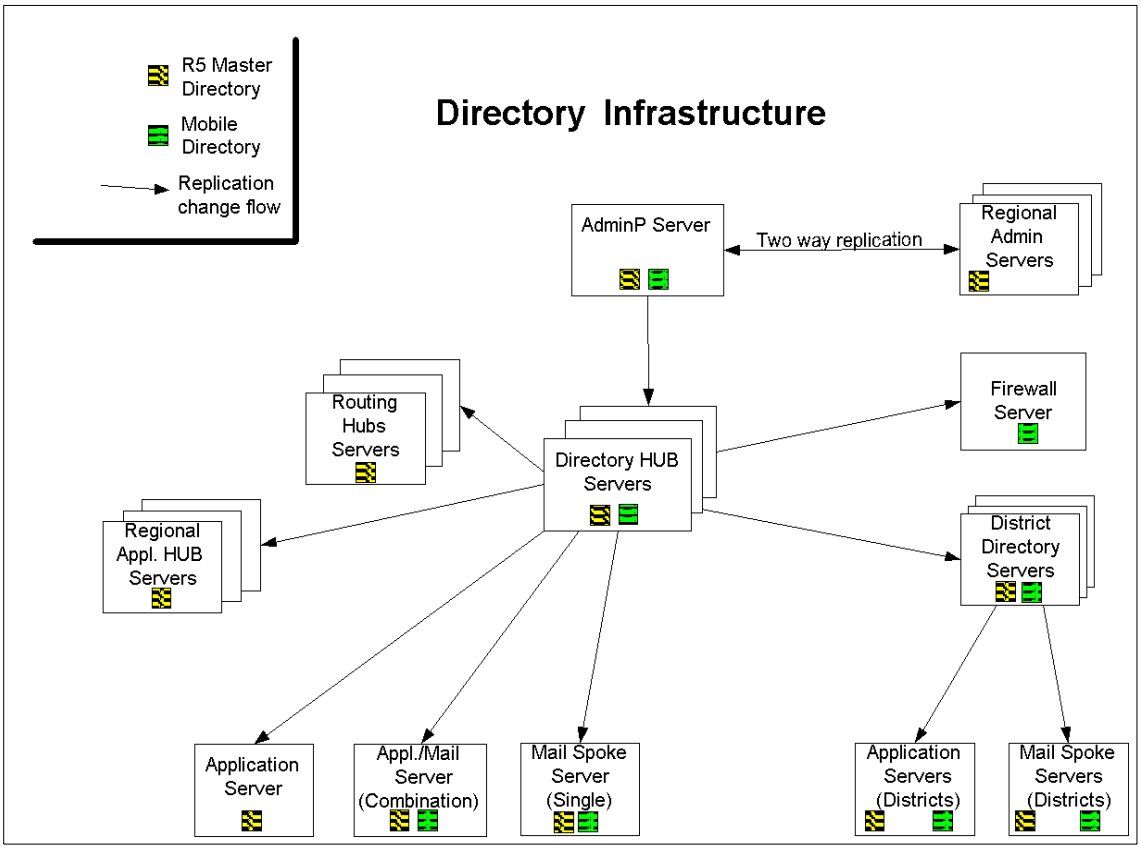


Figure 5. Directory infrastructure

The directory infrastructure design uses multiple directories to achieve the required functionality. As shown in Figure 5, these directories are split up into the following categories:

- The master directory
- The mobile directory
- The firewall directory

The servers involved in maintaining and distributing the directory are:

- **The AdminP server**: This is the primary source of the master directory and the primary replication hub for the master directory and all other directories. On this server all automated changes to the directory, such as AdminP requests, are carried out.

- **Regional administration servers**: These servers are used for the day-to-day administration of the regional users. All manual changes to the directory from these servers are replicated directly to the AdminP server. In case this is not applicable to your organization, you can remove this layer and use only the AdminP server.
- **The directory hub servers:** These servers are the secondary source of the directories and replicate changes received from the AdminP server throughout the production domain. Depending on how many spoke servers you have and how they are distributed, you might want to eliminate this layer and replicate directly with the AdminP server.
- **The district directory servers**: These servers can be used in locations where there is a large user community with multiple servers, but with poor network connections to the main center where the directory hub resides. The function of the district directory server is identical to a directory hub, except that it services a limited number of servers in a selected area.

2.1.1 Directory infrastructure servers

This section describes each of the directory infrastructure servers in more detail.

2.1.1.1 AdminP server

The AdminP server is the administration server of the master directory and hosts the following directories:

- Master directory
- Mobile directory

The AdminP server facilitates the replication of the master directory with the regional administration servers. Also, the AdminP server generates the mobile directory (using the Directory Catalog process), which it pushes to the directory hub servers along with all other directories that it shares.

Access to the AdminP server should be restricted to a team of directory coordinators.

As part of the administrative infrastructure, the AdminP server:

- Performs all global, automated administration procedures on the master directory
- Uses agents to update HTTP passwords within the master directory
- Generates and maintains the mobile directory

To ensure all changes are propagated quickly throughout the production environment, the directory hub servers and regional administrative servers replicate every 15 minutes with the AdminP server.

2.1.1.2 Regional administration server

This server type is only applicable when your Domino administration is done at several different locations. If this is not the case, you might not use this server, but use only the AdminP server instead. If you decide not to use the regional administration server, take into account the extra load you will put on the AdminP server.

The regional administration servers are the central administration point for all manual directory updates carried out in a region and are located at the regional administration site. These servers host the master directory. All directory updates are replicated back to the AdminP server and then to the rest of the production domain.

Regional administration servers generate the AdminP requests for each region, which are replicated back to the AdminP server along with the directory. All regional-based administration tooling is located on the regional administration servers.

The following tasks are managed on a regional administration server.

- Group management, such as updating group membership.
- Directory management, such as updating existing server and configuration documents.

2.1.1.3 Directory hub servers

This server type is only applicable when you have a large number of spoke servers connected over LAN/WAN network links. If the number of spoke servers is low, you can use the AdminP server directly. When you are also using regional administration servers, you should consider placing one or more directory hub servers in each administration site. The directory hub servers provide the following services:

- Pull all directories hosted by the AdminP server
- Provide a replication host for all directories
- Provide district directory services for servers located on the regional wide area network

Directory hub servers host the following directories:

- Master directory

- Mobile directory

2.1.1.4 District directory servers

District directory servers are located wherever there are three or more servers grouped on a local LAN or WAN link and there is limited bandwidth available between the directory hub and the LAN in this district.

District directory servers host the following directories:

- Master directory
- Mobile directory

District directory servers are the primary source of replication for these directories at the local level. In small districts, district directory server services can be provided on a single existing server.

2.1.2 Directories

This section describes the two main directories used in a Domino directory infrastructure.

2.1.2.1 Master directory

This is the main directory, containing all documents in the production domain. The master directory is the core directory, and is used to support:

- Domain security
- User authentication
- Mail routing
- Server configuration
- Replication

The master directory is the primary location for all production domain administration and is used as the source directory for the mobile directory.

2.1.2.2 Mobile directory

This is a compressed Domino directory that is provided using the Domino Directory Catalog process. This directory is designed to provide laptop users with consistent *mail user* and *mail group* addressing when connected and disconnected from the network. The mail user and mail group information is generated from the master directory. By utilizing a mobile directory, the following objectives are achieved:

- The mobile (laptop) user community has a comprehensive addressing directory that can be maintained on their laptop device.

- The directory footprint required to host the entire production directory is reduced.
- Indexer effort on the client is reduced.

2.1.3 Directory replication

The replication of directories throughout the production environment was designed using a top-down approach. Top-down replication is used to aid in controlling the content and integrity of the master directory.

Because all changes to the master directory are performed on the AdminP server or regional administration servers, the AdminP server is the primary replication source of the master directory. Updates made to the master directory on any server, apart from the AdminP or regional administration servers, will not be replicated to the directory master servers or beyond, due to the restricted ACL settings on the master directory.

To support the AdminP administration infrastructure, ADMIN4.NSF is included in the master directory schedule. To facilitate this, the replication is push-pull for all connections and the directory ACL ensures the top-down propagation model.

Since this schedule does not allow directory changes to be made on the local server, there is a need for HTTP password resets to be replicated down the tree. To distribute the password reset in a timely fashion, you have two options:

- Increase the replication interval of the directory to a response time which is acceptable to the users (for example, 20 minutes). Be aware that this might cause a significant impact to the indexer if you have a large directory. We recommend that you pilot this increased schedule first to see if there are any significant performance problems.
- Use the directory hubs and district directory servers as authentication servers using LDAP for HTTP users. This would still require a frequent update between the AdminP server and the directory hub and district directory servers, but the spoke servers would no longer require a fast replication schedule. However, this method also has implications such as:
  - The directory hub and district directory servers, which are considered to be infra-servers, now also have to be accessible for end users.
  - The directory hub and district directory servers have to load the HTTP and LDAP server tasks.
  - In locations which do not have their own district directory server, all HTTP users need to go over a network link to the directory hub. This

might result in an overload of the network link, in which case, additional district directory servers need to be installed.

2.2 Administration infrastructure

The administrative infrastructure is key to managing the infrastructure and is vital for maintaining and managing a secure, stable, and consistent infrastructure.

The design for an administration infrastructure contains a single Domino AdminP server and multiple regional administration servers. Global, automated administration is carried out on the AdminP server and regional administration, while manual tasks are carried out on each regional administration server.

The administrative infrastructure is designed to:

- Move as many directory updates as possible to the master directory on the AdminP server. This ensures a single source of updates on a global scale.
- Provide a stable and secure infrastructure for the continued use of the administration process for automating master directory updates.
- Provide an administrative infrastructure that is accessible only by the Operations and directory coordination teams. This provides each administrative region with the infrastructure necessary to manage manual updates to the master directory (those that cannot be done through the administration process).
- Provide directory tooling which ensures that all updates to the master directory are reflected in the mobile directory.

2.3 Message routing infrastructure

The production domain mail routing architecture is based solely on Notes Named Network (NNN) routing. This method provides high levels of reliability, giving seamless failover routing and allows fine control of mail sizes using the R5 mail controls.

The goal of the message routing infrastructure design is to provide the following benefits:

- Give control over the size of mail that is routed.
- Allow individual locations to have larger internal mail sizing than that imposed on the Global Area Network (GAN)

- Route all international mail through mail routing hubs in the regional centers.
- Raise the visibility of single server errors by concentrating the traffic that generates the errors onto the mail hubs instead of from each individual location server. This requires that the Notes Named Networks coincide with the physical network structure.
- Route mail to the Internet from all of the regional mail hubs, using the R5 SMTP routing facility built into the native R5 router.

Figure 6 on page 26 shows how the various NNNs relate, and how communication with other Domino domains and the internet is configured.

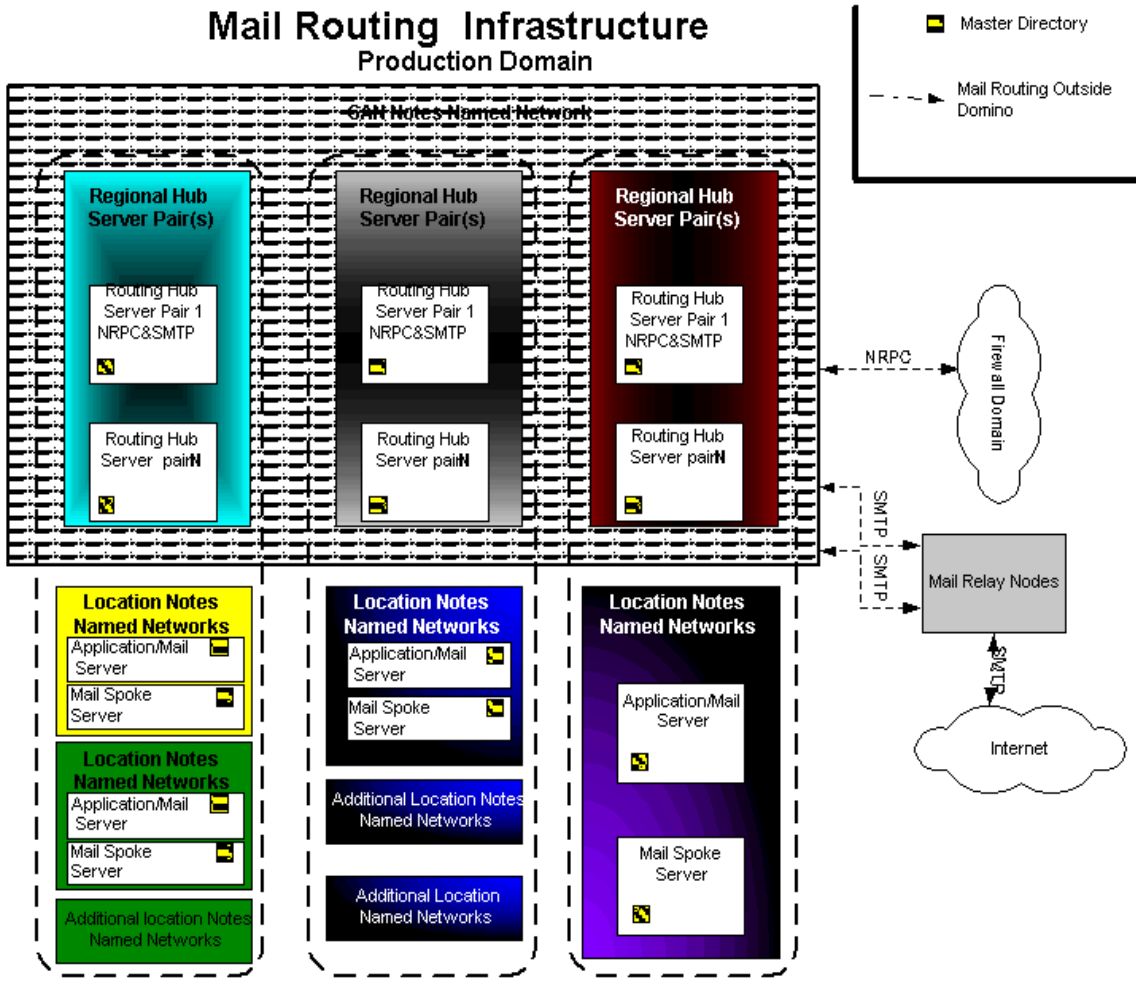


Figure 6. Mail routing infrastructure

2.3.1 Regional mail routing hub servers

Regional mail routing hub servers, servicing both SMTP and NRPC, are grouped into pairs for high availability. Each pair functions as a primary and a backup for the Notes Named Networks which use them.

The hub servers can be configured to reject all mail that exceeds a maximum mail size.

Each regional mail hub takes advantage of the R5 multiple mail.box feature. Initially this is set to three mail.box files to facilitate greater routing

throughput. The number of mail.boxes per server can be refined, based on monitoring, to improve performance characteristics for each hub pair.

The number of mail transfer threads to be configured is based on the number of servers that the hub is expected to serve. The minimum number of transfer threads is 10. This can be extended up to the number of servers that the hub serves to a maximum of 50.

The number of consecutive transfer threads is to be set to the minimum number of transfer threads.

2.3.2 Connections and named networks

This design supports two types of Notes Named Networks, one at the corporate or global level, the other at the location level. A location is defined by the local network configuration. All servers on the same LAN can be grouped into one location. Whenever long distance network links have to be used, it is advisable to define a different location. This can be a building, a city, or a country.

Each location is assigned its own unique Notes Named Network. By mapping your logical NNN structure to the physical network links, any failures in the physical network are then easier to locate.

Depending on the volume of traffic to be managed, there may be more than one pair of mail hub servers per regional center. Each server in a regional mail hub pair is a member of the associated Global Notes Named Network as well as its assigned location Notes Named Networks.

Regional mail hub server network ports have a maximum of 50 location Notes Named Networks assigned to the network port. To accomplish this, the server document design in the master directory has to be altered to contain a multi-valued named network field. The following procedure describes how this can be accomplished.

1. Open the **Server\Servers** form of the directory with a Domino Designer Client.
2. Select **Ports ->Notes Network Ports** as shown in Figure 7 on page 28.

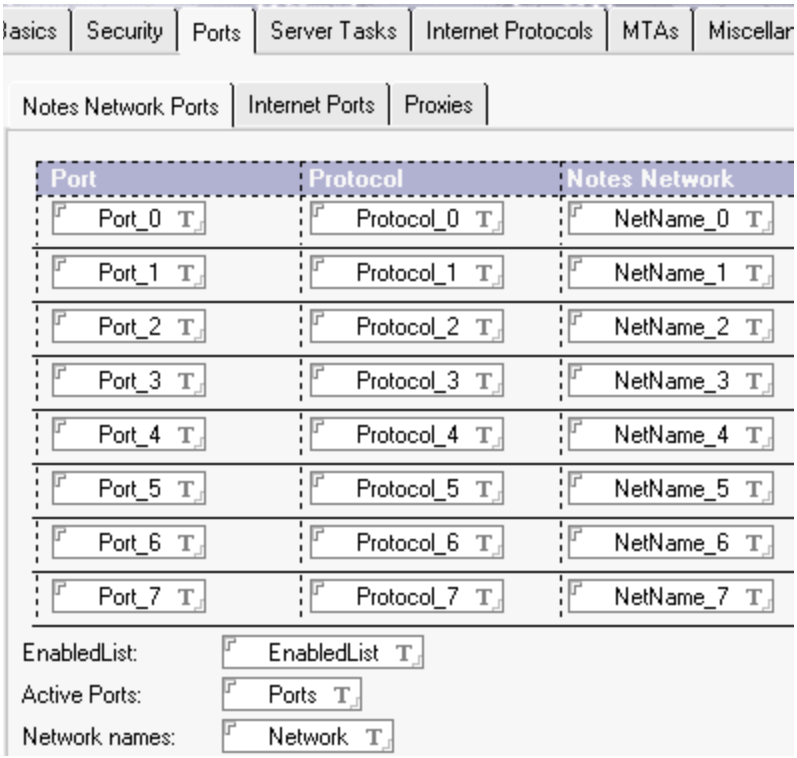


Figure 7. Network Ports tab

3. Click each of the **NetName_0** through **NetName_7** fields in turn, and select **Allow multiple values** for each field, as shown in Figure 8.

Figure 8. Field properties

4. From the beanie tab, enter the following **Multi-value** options:

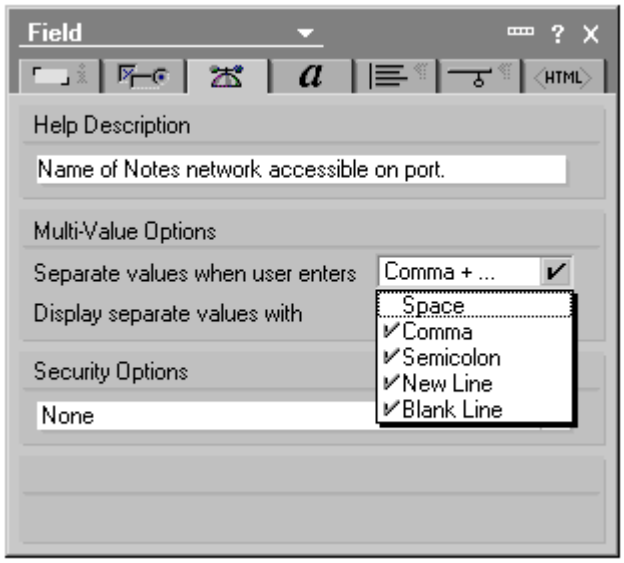


Figure 9. Multi-value options window

5. The **Network** field is a computed value based on the other fields and is already able to support multiple values. Change the **value** formula for this field to equal:

   ```
   @TRIM(@IF(Enabled_0="1";NetName_0;""):
   @IF(Enabled_1="1";NetName_1;""): @IF(Enabled_2="1";NetName_2;""):
   @IF(Enabled_3="1";NetName_3;""): @IF(Enabled_4="1";NetName_4;""):
   @IF(Enabled_5="1";NetName_5;""): @IF(Enabled_6="1";NetName_6;""):
   @IF(Enabled_7="1";NetName_7;""))
   ```

Location Notes Named Networks assigned to network ports are to be balanced to spread the routing workload equally across the ports. Location Notes Named Networks will be spread evenly over the hub pairs, providing good load balancing.

2.3.3 Internal messaging routing

All internal mail can be transferred using the NRPC protocol. The SMTP protocol will only have to be routed by the regional mail hubs. The SMTP routing on the regional hubs is to support messaging in to and out of the production Domino domain.

2.3.4 External messaging routing

All messages sent to other Notes domains go through the firewall domain as NRPC messages. All messages sent out to a non-Notes destination, for example the Internet, are converted by the mail routing hubs into SMTP messages.

2.4 Application infrastructure

The application infrastructure has been designed to provide a robust server infrastructure with different servers offering different specialized services in order to maximize performance and reliability.

2.4.1 Application infrastructure server types

Following is a hierarchy of application servers offering different application infrastructure services (in order from lowest to highest):

- **Application spoke servers**

  These are the servers accessed by end users in order to use applications. An application spoke server is a server dedicated to hosting applications. It does not provide mail hosting services (although it does support mail routing for mail-enabled applications).

  A variation on the application spoke server is the application/mail spoke server, which provides both application hosting and mail hosting. This type of server is typically deployed for user groups that are too small to justify a separate application spoke server and mail spoke server.

  For the purposes of the application infrastructure, the configuration of the application aspects of application/mail spoke servers is identical to that of dedicated application spoke servers. Therefore, in this chapter, references to application spoke servers include application/mail spoke servers unless otherwise specified.

  Directory hub servers provide directory updates to application servers. If the application server is placed in a district, then the directory updates will come from a district directory server. See Figure 5 on page 19, for more information.

- **District application servers**

  These are located wherever there are two or more application servers grouped on a local LAN or WAN link and there is limited bandwidth available between the directory hub and the LAN in this district.

District application servers act as application replication hubs within the district. If a district only has one application spoke server in it, then there is no need for a district application server.

District application servers are generally not available for use by end users. Instead, they are dedicated to replicating application data and designs between the application spoke servers in the district and the application hub server for their region. See the description in Figure 10, for more information.

District directory servers provide directory updates to district application servers, also shown in Figure 10.

- **Application hub servers**

  These servers act as replication hubs for application data and design propagation across regions. They also replicate with equivalent servers in the firewall domain. End users do not normally access these servers.

  District directory servers provide directory updates to application hub servers, as shown in Figure 5 on page 19.

  Application hub servers are generally not available for use by end users and normally only support Notes clients for access by members of the operations team.

You can cluster servers in pairs to provide load balancing and failover where required.

2.4.2 Replication topology

In the replication topology scheme shown in Figure 10 on page 32, push-pull replication is initiated by the spoke servers. This delivers the following advantages:

- The application hub server is able to service more spoke servers, because it only has to facilitate the replication.
- There is no single point of failure in case the replicator task hangs.
- As the replication hub servers are now the target servers for replication, you can take advantage of clustering to increase the availability.

On the downside, to check the log for replication errors, you will now have to go to each individual spoke server instead of just the application hub server. This can be overcome by using third party tooling to collect the log information and store it in a central location.

The replication topology between the regional replication hub servers is done using a round robin method, ensuring that the updates will come from two places.

However, this is based on the assumption that there is a physical network link between the application hub servers that are available. If this is not the case, you should create a replication schedule that fits your physical network topology.

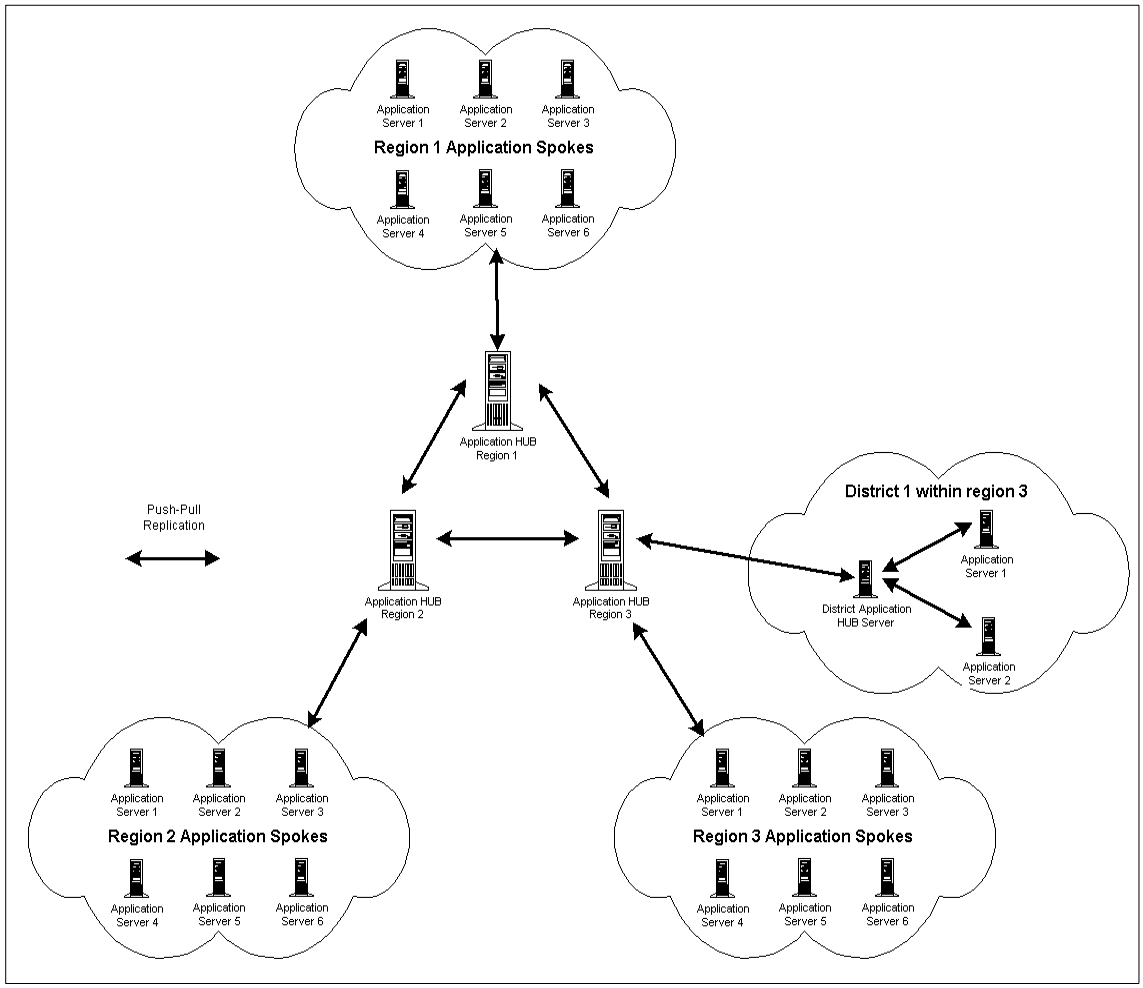


Figure 10. Replication topology

Within a district, the application spokes initiate replication to the district application hub. The district application hub then initiates the replication to the regional replication hub. This method saves bandwidth but requires an

additional infra server. Each time you use a district replication hub, you should carefully consider whether having an extra infra server is worth the effort. Alternatives to a district application hub are:

- Consolidating the application spokes within a district into one server
- Increasing the bandwidth between the regional application hub and the district

2.4.3 Server failover and load balancing

Where it is practical and cost-efficient to do so, application servers are clustered in order to provide failover and load balancing. The minimum cluster size is two servers.

Clustering is implemented using a number of complementary tools in order to support Notes and Web browser clients.

2.4.3.1 Clustering for Notes clients

For Notes clients, server clustering is implemented with Domino clustering technology. Clustered servers implement optimized cluster replication, often using separate NICs and intra-cluster LANs (compared to their other network tasks) for cluster replication and management tasks.

Clustered servers are usually set up to be exact copies of each other in terms of hardware specification, OS version, Domino version, and server contents. This allows for a clean failover from one server to another in the cluster in the event of the first server failing. It also allows optimum load balancing, as it does not usually matter which cluster server a user is connected to, normally allowing a user to be connected to the least loaded server.

However, in some sites it may be appropriate to use asymmetric clustering using different specifications of hardware within the cluster because user connectivity to some servers in the cluster may be inferior to others.

2.4.3.2 Load balancing thresholds

The detailed design document for clustering specifies the server utilization (load) threshold at which the next user attaching to a server is switched to another server in the same cluster.

The threshold varies from cluster to cluster depending on the hardware specification of the servers within a cluster, the number of servers in a cluster, and the connectivity between users and the different cluster members. Guideline thresholds are provided for known cluster types based on the above parameters. However, it is expected that in some instances the

thresholds will be modified by operations personnel to account for specific usage patterns.

2.4.4 Application replication

The replication infrastructure relies on the application server hierarchy, described in the previous section, in order to replicate data/design changes between servers in different parts of the hierarchy.

The standard replicator task running on each server deals with the actual mechanics of replication. The number of replicator threads running on a server depends on the number of processors (CPUs) incorporated in the server hardware. Replication performance is fine-tuned through the use of connection documents, together with replication priorities and replication schedules.

Connection documents are created in order to allow each server to initiate a replication connection with its parent in the application server hierarchy, or, if it is an application master server, then with its peers. There are no connection documents allowing a server to initiate a replication connection with its children in the server hierarchy. Each connection document contains information on the priorities of databases to be replicated and when replication should be scheduled.

The three levels of replication priority are defined as:

- **High**: This level requires high frequency replication.
- **Medium**: This level requires medium frequency replication.
- **Low**: This level requires low frequency replication.

Application databases are allocated one of these priorities by their replication settings dialog, depending on the frequency with which they need to replicate. It is anticipated that the majority of databases will be set to medium priority.

Chapter 3. Architecture and system design

This chapter explains the following architecture and system design topics.

- DPARs and LPARs
- Network connections
- HFS use and architecture
- Using database links, directory links and symbolic links
- Transaction logging
- Using multiple time zones
- Using mail policies and quotas

3.1 DPARs and LPARs

This section explains Domino partitions (DPARs) and S/390 logical partitions (LPARS). For more information, see *Lotus Domino for S/390 Release 5: Installation, Customization and Administration*, SG24-2083, and *Lotus Domino for S/390 Release 5: Performance Tuning and Capacity Planning*, SG24-5149.

3.1.1 Domino partitions

Partitioned servers share the same binary files (version of Domino code), but have individual data directories, which allows for independent configuration of resources.

Following are several reasons for implementing partitioned servers.

- Segregation of server tasks on a single machine

  For example, a mail server and application server can have individually tuned Domino tasks, but run on the same hardware.

- Segregation of independent user communities

  For example, you might want to host multiple Domino domains or Domino-based Web sites on a single machine.

- Independent operation of services and resources

  Partitioned Domino servers operate independently and do not have any effect on each other. This allows an administrator the flexibility to treat each partitioned server as though it were a separate system.

- To reduce operating system and hardware limiting factors that are valid within an LPAR

The previous limit of six supported partitioned servers on one system has been removed in Domino Release 5. System resource was the limiting factor. However, although you can now run more servers, the amount of CPU needed to run more servers does have an additional increase, for each added server.

See *Lotus Domino for S/390 Release 5: Installation, Customization and Administration*, SG24-2083 (edition 02) for a longer discussion on partitioning and the CPU effects that it has.

3.1.2 Logical partitions

For those readers who understand S/390 logical partitioning, it may be helpful to understand some of the differences between Domino partitioning (DPARs) and logical partitioning (LPARs).

With logical partitioning, you can run up to 15 different operating systems on a single physical S/390 processor. These operating systems can be any combination of the operating systems and release levels available on S/390. We could, for example, run an OS/390 Release V2R8 system, and an OS/390 Release V2R10 system on the same processor. Therefore:

- We could run multiple OS/390 operating systems, each with a single Domino server, on a single S/390 processor. This would provide better workload isolation, because now the operating systems are not shared.
- With separate OS/390 operating systems, the Domino servers can be at different software levels. This would enable a test or development environment to run in parallel with a production system.
- You could use a combination of Domino partitioning and logical partitioning to run multiple OS/390 systems with multiple Domino servers on each one.

On a S/390, you could also use hardware partitioning on a multiprocessor or VM/ESA to provide the same types of function.

For more information on S/390 logical partitioning, see *S/390 Processor Resource/Systems Manager Planning Guide*, GA22-7236.

3.2 Network connections

Domino server uses only TCP/IP as a network protocol. So during a Domino project, network planning is very important. Different Domino servers running in one large S/390 server present some advantages when compared to implementing many smaller servers:

- Network traffic from one server to the other will not go to the network. If you are using an adapter such as the Open Systems Adapter (OSA), then link traffic will use IUCV, which is an internal host layer.
- Multiple DPARs in the same LPAR can use less server memory because Domino modules will be loaded into memory just once, and are then shared for all Domino partitions.
- When the workload in one LPAR is too high, the server tries to use other LPAR resources (if you are using the Workload Manager). Thus, you can use idle servers if others become jammed.

Centralizing your Domino servers in one host simplifies administration procedures, but it changes your network traffic. When you work with multiple servers, your users probably access a local server through a LAN. But the users are not the only source for bandwidth consumption. Another is the server's communication for replications, for example. If users are spread out, there will be server-to-server traffic over a WAN. Now, when you have one large server, all of the users that are spread out will reach the server through the WAN. So, an important consideration is your WAN. Another issue is the S/390 LAN adapter, because all user traffic will use it. If not planned properly, it can become a bottleneck.

Figure 11 on page 38 shows network traffic differences with many Domino servers and one large server.

Section 4.3.4, “Network” on page 87, provides a formula for calculating the necessary bandwidth for your server, based on the estimated number of users. You can use this information to decide the appropriate LAN type to connect your server.

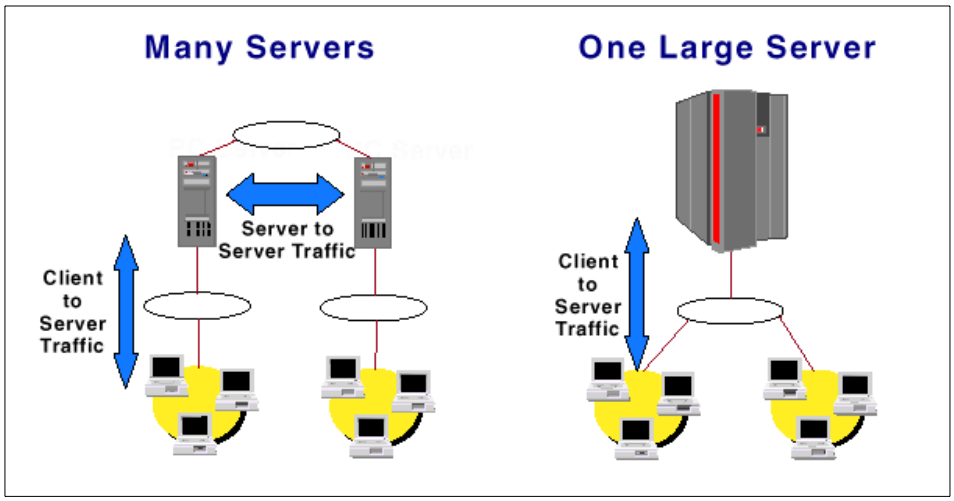


Figure 11. Different network traffic when using one or multiple servers.

Depending on the type of LAN—Ethernet, Fast Ethernet, ATM, Token-Ring, FDDI, Gigabit Ethernet—that your server is attached to, you need to choose a LAN adapter for the server. Some of the adapters you can use are:

- The OSA feature of the S/390 server
- A 37x5 communications controller
- An IBM 3172 Model 3 controller
- An IBM 2216 Nways Multi-access Connector
- CIP card

The best adapter choice depends on your LAN and on your S/390 host model.

Figure 12 on page 39, shows a S/390 server connected to a LAN using an OSA adapter.

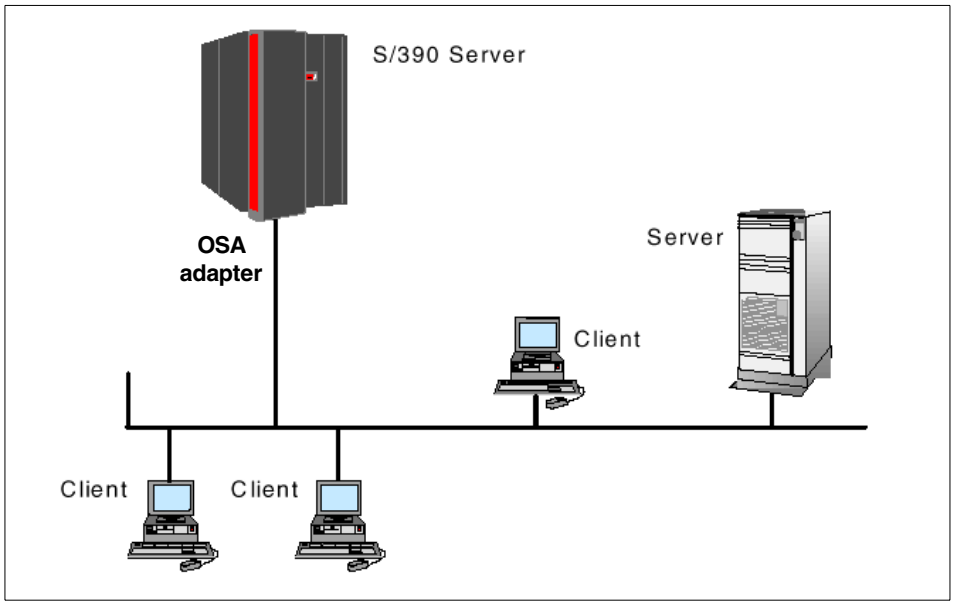


Figure 12. S/390 server with an OSA adapter

OSA cards have some features that we must point out.

- It is configured using an OSA Address Table (OAT), which allows you to share it among different LPARs. Outbound datagrams are sent to the network, except if the destination IP address corresponds to a TCP/IP stack sharing the OSA adapter. For such an LPAR-to-LPAR connection, the datagrams are sent directly to the other LPAR without using the network. Now, if there are two Domino servers, each of them in one of these LPARs, communication among them will not even go to the network.
- You can configure OSA cards using a GUI interface in an OS/2 workstation connected to the host, or you can use a TSO REXX interface. No matter which interface you choose, OSA configuration is a non-disruptive process. The device and channel will need to be taken offline, but no IPL is required.

For detailed information on installing and configuring OSA adapters, see *OSA-2 Implementation Guide for OS/390*, SG24-4770-01, and *S/390 OSA-Express Gigabit Ethernet Implementation Guide,* SG24-5443.

Another consideration is how critical your Domino server is. Using two LAN adapters for each LPAR can protect your server from not being able to reach the network.This can be achieved using virtual IP addressing (VIPA). With

VIPA, you create a virtual adapter that is used by the server for incoming and outgoing network traffic. VIPA redirects this traffic to one primary physical adapter. If this device fails, VIPA redirects traffic to a second adapter, achieving redundancy. Figure 13 shows a Notes client recovering from a connection failure to a Domino server in a S/390 host using VIPA.

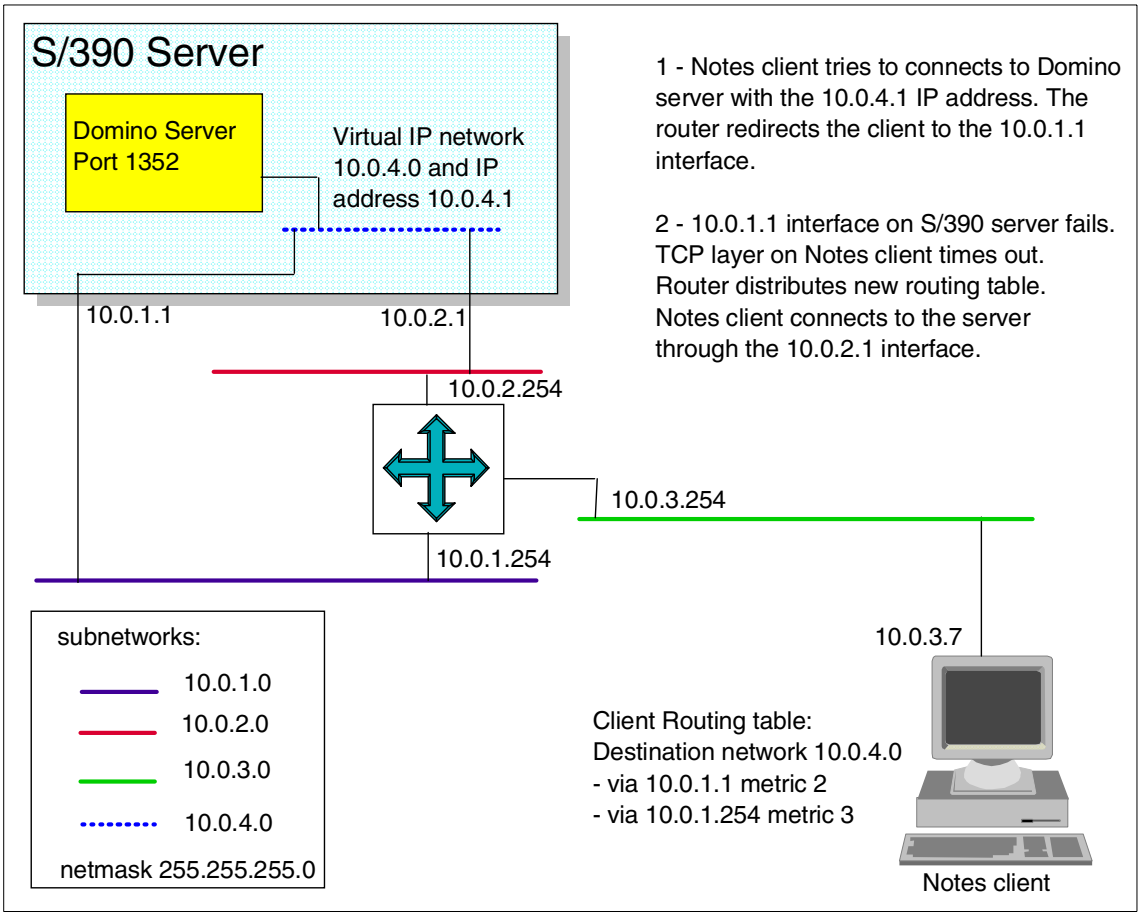


Figure 13. Inbound connection recovery with VIPA

Implementing VIPA requires some knowledge of TCP/IP, OS/390, and routing protocols. You can find detailed information in *Configuring TCP/IP and VIPA in OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing*, SG24-5227. APARs *II11710, II11711, and II11712* should also be read for general information about TCP/IP.

From a Domino perspective, the virtual adapter IP address is used for configuration. But one issue in using VIPA is that it needs one of three routing protocols to work automatically:

- RIP 1
- RIP 2
- OSPF

If your IP network routes are static, VIPA will not be able to change from the primary adapter to the secondary without manual changes to the routing table.

Besides providing adapter redundancy, you can use VIPA for connecting different DPARs. You would use a single TCP/IP port and a single virtual IP address for the users while using various real IPs for each Domino.

Following are other alternatives for running different DPARs in the same LPAR.

- Use different IP stacks, one for each DPAR.
- Use different IP addresses, one for each DPAR, but the same IP stack.

  By using different IPs, you can isolate applications. Different servers can bind to different IPs. So clients can reach the servers through different networks.

  If you are planning to use a network backup utility, such as Tivoli Storage Manager (TSM) or Tivoli Data Protection (TDP), you should use a network backup utility for Domino client access, a different backup utility for server-to-server procedures such as replication, and yet another for backup purposes. These backup tools allow you to run backup routines on active servers. Having only one network could be a bottleneck and cause poor system performance. You can find more information about TSM and TDP in Chapter 5.2.6, “Tivoli Storage Manager (TSM)” on page 101.

  The IP address to which the Domino server will bind is configured during installation, when you provide the host name. But, when you use more than one IP stack, you must configure your server to use the appropriate one. Set the value of the _BPXK_SETIBMOPT_TRANSPORT variable with the stack name you want to bind the server. This variable must be put in the .profile file of the user under which the server will be running.

  In Figure 14 on page 42, the server called Domino 1 uses TCP/IP stack A. This stack uses the Int x device, connected to the 10.0.1.0 subnet. Domino 1 network traffic is isolated from Domino 2 traffic, which uses a different TCP/IP stack, device and subnet.

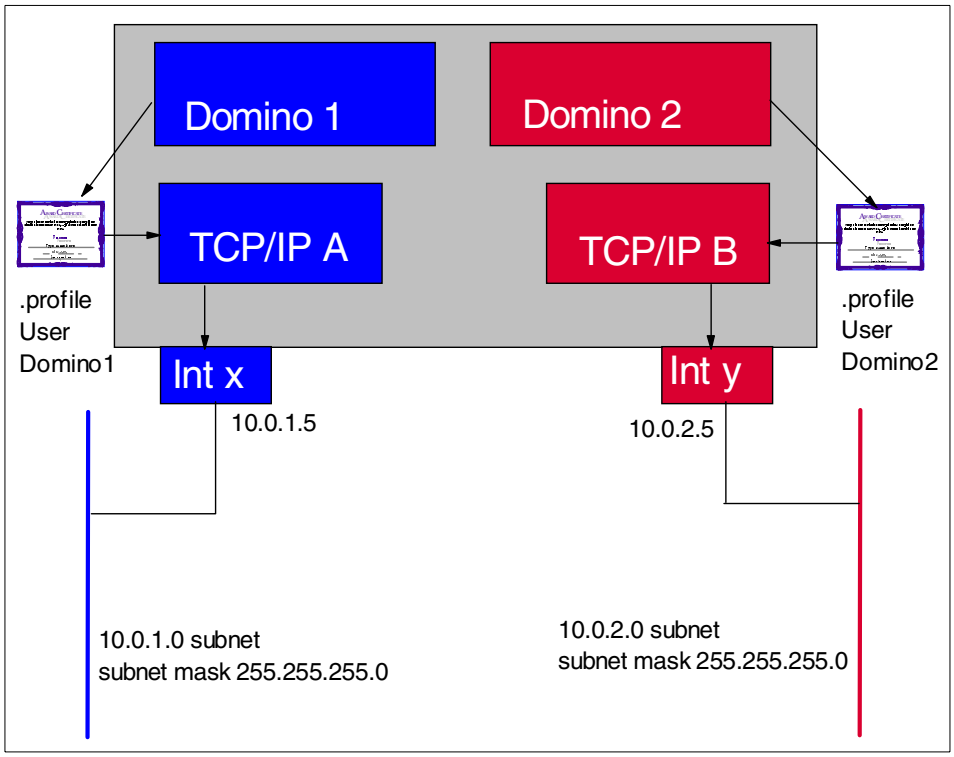


Figure 14. Restricting Domino to bind to a single TCP/IP address

Note: If you need to isolate applications and you also need device redundancy, you can set a combination of VIPA and different IPs.

- Use the Domino port mapper.

  The port mapper allows you to have different DPARs using the same IP address and, from the user perspective, the same TCP/IP port. In fact, with port mapper, each different DPAR has its own port number. One server, with default port 1352, will work as the port mapper server. Figure 15 shows three servers using a single IP address with port mapper. All of the clients access this port mapper server and will be redirected to the appropriate server.

  However, this approach is not used much, because it is far more complex, more vulnerable, and less efficient. You need one dedicated server to work as the port mapper; if it fails, new sessions cannot connect. Existing sessions do remain connected. It is even possible that a client, with server information in cache, can reconnect to his server while the port mapper is

down. But, because of these uncertainties, port mapping is not recommended.

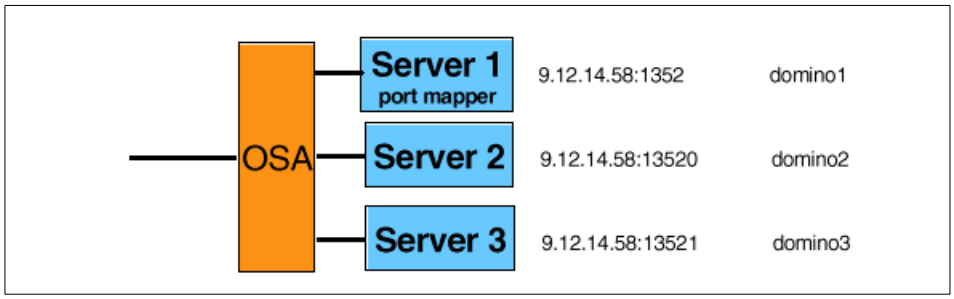


Figure 15. Different servers using one IP address and different ports

One more thing to be aware of about TCP/IP and Domino is that the HOSTS.LOCAL MVS file and /etc/hosts HFS file cannot be used at the same time. If you use the HOSTS.LOCAL file with all of your hosts entries, creating a /etc/hosts with just one entry will make Domino recognize only that one entry in /etc/hosts. So, decide which one you will use. Take care when moving from one to the other to keep all entries up-to-date.

In Figure 16 on page 44, there is an example of one host with two LPARs, LP1 and LP2. In each LPAR there are three DPARs, two for mail and one for applications. For each of these DPARs, there is a cluster DPAR in the other LPAR, shown in the same color, DMAIL01 and DMAIL02, for example. There are two adapters in each LPAR, and the DPARs use VIPA addresses. Two different networks are used in order to provide redundancy and isolate clients from backup traffic, for example. This design provides good server redundancy.

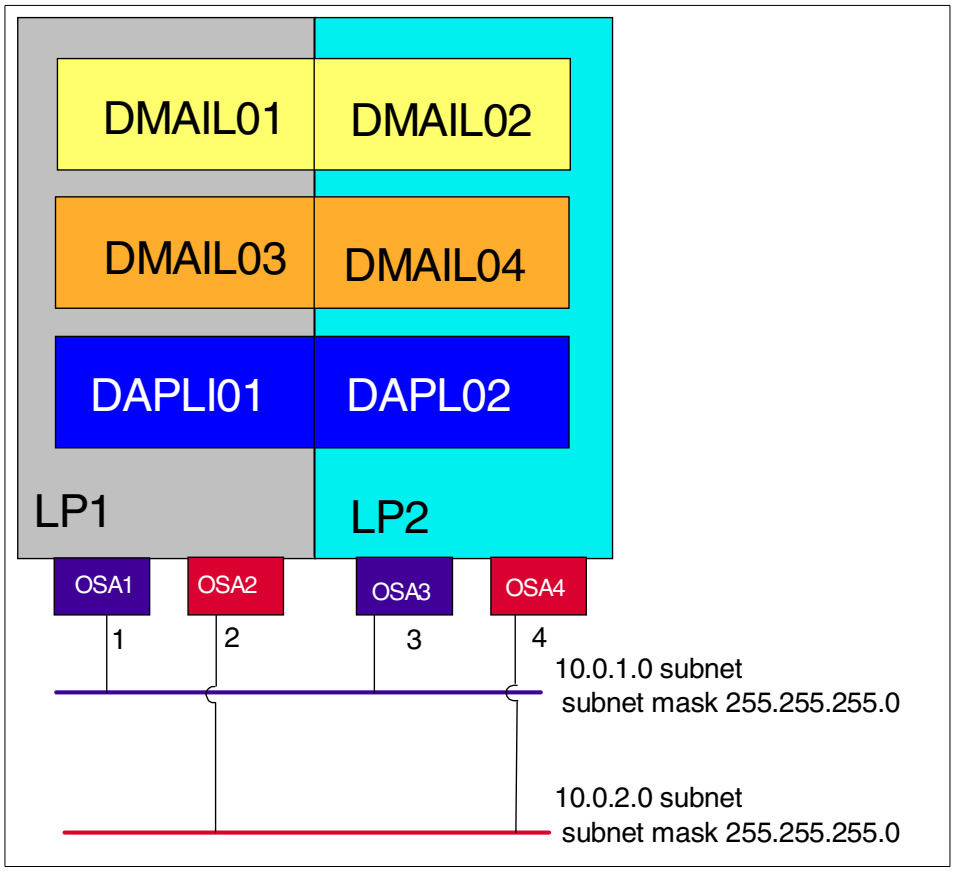


Figure 16. Domino design with server, network and adapter redundancy

3.3 HFS use and architecture

Running a large Domino system on OS/390 requires careful usage of the hierarchical file system (HFS). This section provides an overview of HFS and lists the enhancements provided in DFSMS/MVS Version 1 Release 5, and in OS/390 Version 2 Release 9.

3.3.1 HFS overview

An HFS data set is an MVS data set that contains a POSIX-compliant hierarchical file system, which is a collection of files and directories organized in a hierarchical structure that can be accessed using the OS/390 UNIX System Services (USS).

The file systems within HFS data sets have a tree structure. This is a root directory with various subdirectories that contain files. The files within an HFS data set are identified by their path and file names.

HFS is used by UNIX System Services and its applications such as ftp, NFS, WebSphere Application Server for OS/390, Domino for S/390, and Component Broker.

HFS data sets were introduced in MVS/ESA with DFSMS/MVS Version 1 Release 2, together with support for UNIX System Services. The support for HFS data sets was rewritten for DFSMS/MVS Version 1 Release 5 to increase the performance of an HFS. Figure 17 illustrates the relationship between DFSMS/MVS HFS services and OS/390 USS.

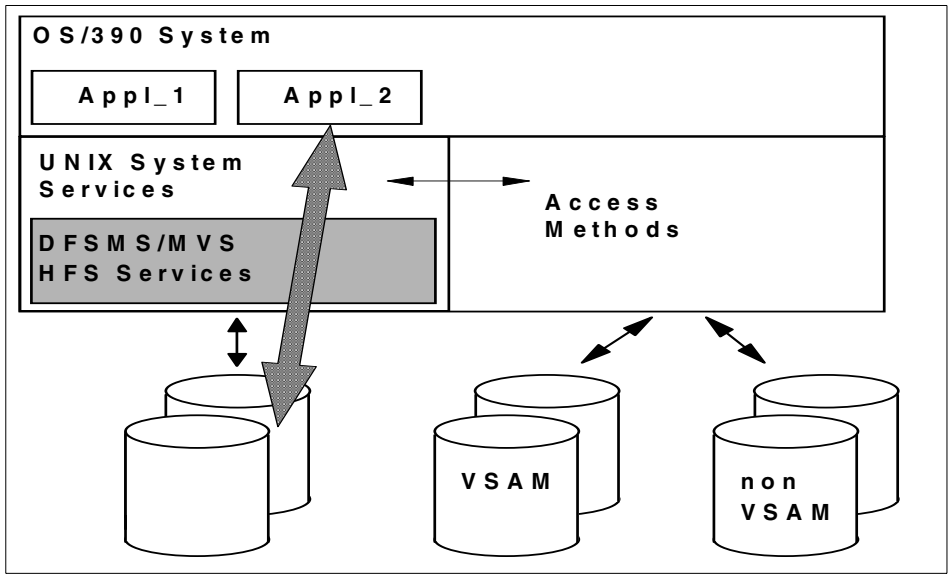


Figure 17. HFS overview

DFSMS/MVS Version 1 Release 5 is a base element of OS/390 Version 2 Release 7.

Following are the basic characteristics of an HFS data set.

- DFSMS/MVS provides access to the files of data within an HFS data set.
- DFSMS/MVS provides storage management capabilities, such as backup/recovery and migration/recall for HFS data sets.
- HFS data sets can expand into multiple volumes (this support is provided in DFSMS/MVS 1.5). The single volume restriction for HFS data sets has

been removed. HFS data sets can now span up to 59 volumes, with up to 255 total extents for all volumes, and up to 123 extents per volume.

- HFS data sets no longer must be SMS-managed. Now it is possible to allocate non-SMS managed HFS data sets.
- HFS data sets can be shared in read/write mode between systems in a Sysplex (support has been provided since OS/390 2.9).
- HFS data sets (not HFS files) have the following requirements and restrictions:
  - Non-SMS managed HFS data sets are supported but must be cataloged and reside on a single volume.
  - They cannot be processed by standard open, end-of-volume, or access methods; POSIX system calls must be used instead.
  - They are supported by standard DADSM create, rename, and scratch functions.
  - They are supported by DFSMShsm for dump/restore and migrate/recall. DFSMSdss is used as the data mover.
  - They are not supported by IEBCOPY or the DFSMSdss COPY function.
  - Partial release for HFS data sets is not supported, since the DADSM PARTREL function no longer supports HFS data sets.

Note

Tivoli Storage Manager can be used to back up, recover, archive, or retrieve individual files or groups of files held in an HFS data set.

3.3.2 UNIX file system overview

This topic provides background information about the UNIX file system, the *hierarchical file system*.

3.3.2.1 UNIX file system structure

The hierarchical file system consists of:

- Hierarchical file system (HFS) files, which contain data or programs. A file containing a load module or shell script or REXX program is called an executable file. Files are stored in directories.
- Directories, which contain files, other directories, or both. Directories are arranged hierarchically, in a structure that resembles an upside-down tree,

with the root directory at the top and the branches at the bottom. The root is the first directory for the file system at the peak of the tree and is designated by a slash (/). Figure 18 is an example of a UNIX root directory, containing a typical set of subdirectories.

- Additional local or remote file systems, which are mounted on directories of the root file system or of additional file systems.

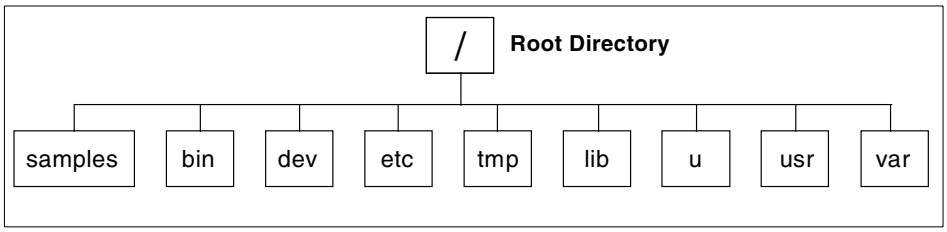


Figure 18. Example of the UNIX file system directory structure

To the OS/390 system, the file hierarchy is a collection of HFS data sets. Each HFS data set is a mountable file system. The root file system is the first file system mounted. Subsequent file systems can be logically mounted on a directory within the root file system or on a directory within any mounted file system.

Except for the direction of the slashes, the hierarchical file system is similar to a Disk Operating System (DOS), Windows, or an OS/2 file system.

When Domino for S/390 is installed according to the installation procedures, the three separate HFS data sets shown in Figure 19 on page 48 are built, with the logical file structure shown in Figure 20 on page 49.

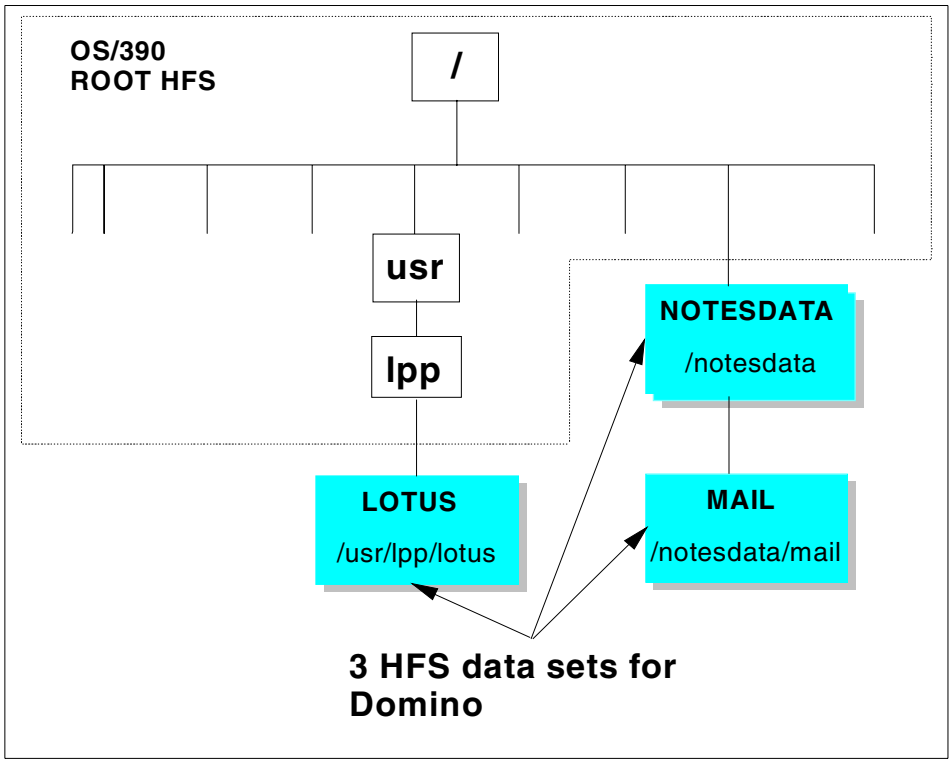


Figure 19. HFS data sets structure of Domino for S/390 installation

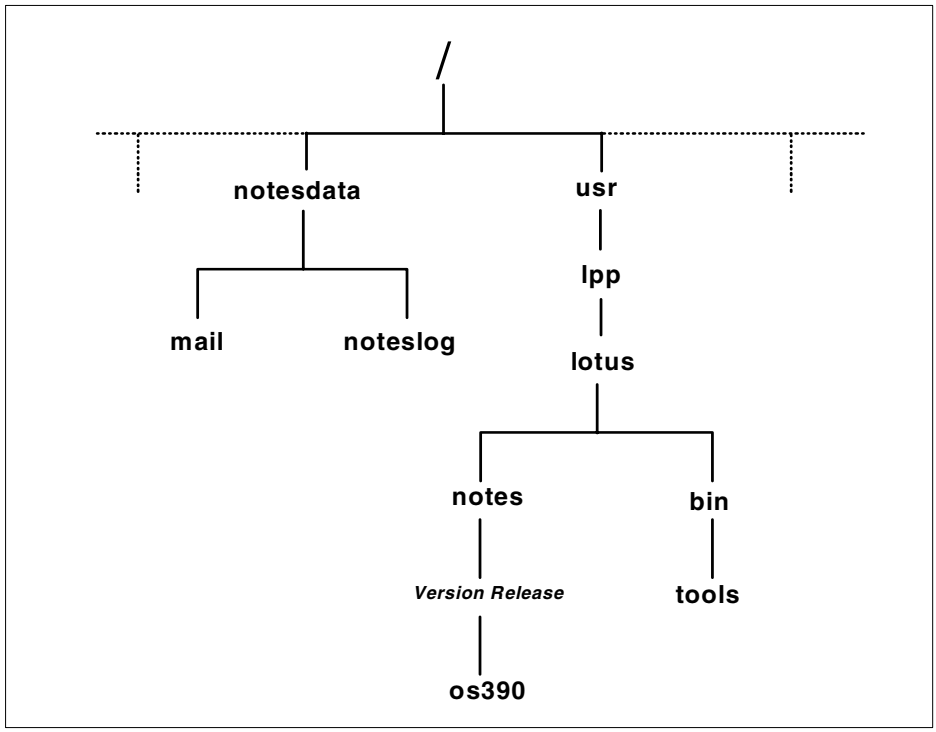


Figure 20. Logical file structure for Domino for S/390

3.3.2.2 Recommended HFS structure and management

Our recommendations for using a Domino HFS structure are:

- One more HFS data set should be allocated on the mount point /notesdata/noteslog, because files under /notesdata/noteslog tend to increase in size. If these log records make the HFS data set full, other files under /notesdata cannot be updated, and this can stop the Domino server activities.
- In general, the mail HFS data set should be separated into several data sets, and the directory structures should be changed according to the recommendations in Domino for S/390. Methods for separating this file system are not discussed in this redbook, but following are some recommendations from a storage management standpoint.

Lotus HFS data set

The Lotus HFS data set, which is mounted on /user/lpp/lotus, consists of several executables and shell scripts related to the Domino server function, and some tools. This data set is an aggregation of read-intensive files, but it

is not totally static; for example, some updates take place when a new server partition is added. If we lose this data set, the Domino server will not run. We recommend making a copy of the Lotus HFS data set, using another data set name. If the Lotus HFS data set is damaged, you can use the backup copy. This file system is considered *static data.*

Note: For a definition of file system data categories (including static, discardable, and critical), see 5.1, “Understanding your backup and recovery needs” on page 93.

Notesdata HFS data set
The Notesdata HFS data set, which is mounted on /notesdata, has various important files used to initialize and manage the entire Domino server. Some of them are updated dynamically. This file system is considered *non-critical data.*

Noteslog HFS data set
The Noteslog HFS data set, which is mounted on /notesdata/noteslog, consists of various log files. These log files continue to increase in size, so the space management to delete the increasing files can be performed automatically by the *cron daemon*, which can periodically run a shell script to delete files. This file system is considered *discardable data.* If these log records are used as statistics data, this file system is considered *non-critical data.*

Mail HFS data set
The mail HFS data set has the following characteristics:

- In general, this HFS data set would be separated into several data sets after the Domino installation.
- This HFS data set consists of many Notes client databases (files, from an HFS standpoint), and they will sometimes need a large amount of storage space.
- Most update I/Os to the Domino system are concentrated in this data set.

This file system can be considered *critical* or *non-critical,* depending on your requirements.

See 4.3.2, “Server” on page 84, for an example of a possible Domino data structure. For more information on HFS internals and usage, refer to *Hierarchical File System Usage Guide*, SG24-5482.

3.4 Using database links, directory links and symbolic links

In an existing Domino infrastructure, standards for the Domino mail/application directory structures are typically well-defined. Therefore, when migrating to S/390, it is not surprising that Domino administrators almost always want to preserve their existing production Domino directory structure. It is easier to maintain and there are fewer procedural changes, so keeping this structure minimizes changes to the server and Notes clients.

See 3.3, “HFS use and architecture” on page 44, on what a S/390 infrastructure could look like.

3.4.1 Organizing databases with directory and database links

The intended use of link files is for organization. For example, you may move a single database or entire subdirectory to another HFS, yet still want to have it appear in the Domino data directory.

Directory and database links are simple text files with a .DIR or .NSF file extension located in the Domino data directory. You can create these files using a text editor or through the server administration panel. A link file contains a full path that points to another subdirectory or a specific database. When the Notes client seeks out available directories and databases, these link files appear as regular directories or databases. Each link file can specify only one directory or database; however, the link can also include a list of groups or users that have access to the link.

Note: While you can control user access with these files, this added functionality is NOT a substitute for proper ACL implementation.

To organize databases on a server you can:

- Store databases in the Domino data directory. This is the default.
- Create subdirectories of the Domino data directory to store groups of related databases.
- Create directory folders and store databases outside of the Domino data directory, then create links to the database from the Domino data directory.

3.4.1.1 Directory links

You can store databases in a directory outside the Domino data directory to take advantage of disk space available on other file systems. Then you can create a link that is in the Domino data directory and points to that directory.

3.4.1.2 Database links

You can create a database outside of the Domino data directory and create a database link to it from the Domino data directory. A database link appears in the Domino data directory as a database icon followed by the name of the linked database. When a request is issued to open the database, Domino redirects the request to the database specified in the link. Moving a file from one directory to another is simply a matter of updating its database link; the original Domino directory structure is maintained and no changes are required on the server or client sides.

For a small number of databases, database links are a very good solution. However, if an installation has hundreds or thousands of files, the administrative overhead to manually maintain these links is much greater, although UNIX scripts could be written to create and manage them. One thing to keep in mind is that Domino caches databases by default at close time if no users or processes are using those databases. Doing so reduces the open times for databases if they are accessed again within a certain window of time. Caching makes it very dangerous to move databases while the server is running, whether the move is initiated through an OS command, UNIX script or administration client because the cache could be pointing to the disk space before the move. To guarantee full server database integrity, we recommend that you move files and create the database links while the server is down. It should be noted that the backup/recovery products, such as Tivoli Storage Protection, do not back up database links. TDP will back up databases pointed to by database links; however, the links themselves must be backed up through the Tivoli Storage Manager's backup-archive client.

3.4.2 Using UNIX System Services symbolic links

A symbolic link is a file that contains the pathname for another file; that pathname can be relative or absolute. If a symbolic link contains a relative pathname, the pathname is relative to the directory containing the symbolic link. You can create a symbolic link to a file or directory. Additionally, you can create a symbolic link across mounted file systems, which you cannot do with a hard link.

To create a symbolic link to a file, you can use ISHELL or the following command:

```
ln -s old new
```

New is the name of the file containing the reference to the file named *old*.

Notes administrators may prefer the use of database links, whereas S/390 and UNIX System Services people prefer the use of symbolic links. There

should be no difference or performance impact when using one method or the other. Every installation should decide which way they prefer.

Our recommendation here is to spread a large mail directory across smaller mail directories, allocate an HFS for each of them, and mount them to the Domino data directory instead of using one of the previous mentioned methods.

For more information on this, see 3.3.2.2, “Recommended HFS structure and management” on page 49.

3.5 Transaction logging

Transaction logging is a new data storage technology that enhances Domino R5 server availability, reliability and efficiency of data recovery and backup operations. Used in combination with R5's new backup capabilities, transaction logging enables you to deploy fully recoverable servers at a lower cost of operation.

3.5.1 Definition

Transaction logging is based on ARIES (Algorithm for Recovery and Isolation Exploiting Semantics), developed by Almaden Computer Science at IBM and implemented in Domino by a joint team of developers and researchers from IBM and Iris Associates. Lotus Domino R5 is the first semi-structured data store that implements transaction logging. The ARIES algorithm has been in production since 1988, and versions are used not only in Domino R5, but also in IBM DB2 mainframe and workstation products, IBM MQSeries, and many other products.

The transaction log is one common log for all transactions in Domino databases. Each transaction is a series of changes done in a database to complete one indivisible service. Some examples of transactions are creating a new document, adding text, and saving the document. This log may consist of multiple log files, called extents. Transaction logging is only valid for databases in the Domino R5 format.

3.5.2 How does it work?

When transaction logging is enabled, the Domino R5 server captures all of the changes that users make to databases and writes them to a transaction log, rather than immediately writing them directly to the various databases. This allows Domino to defer write operations to databases during times of

high server activity. When resources are available, or at periodic intervals, logged transactions are written to disk in batch.

Figure 21 shows the transaction logging writing process.

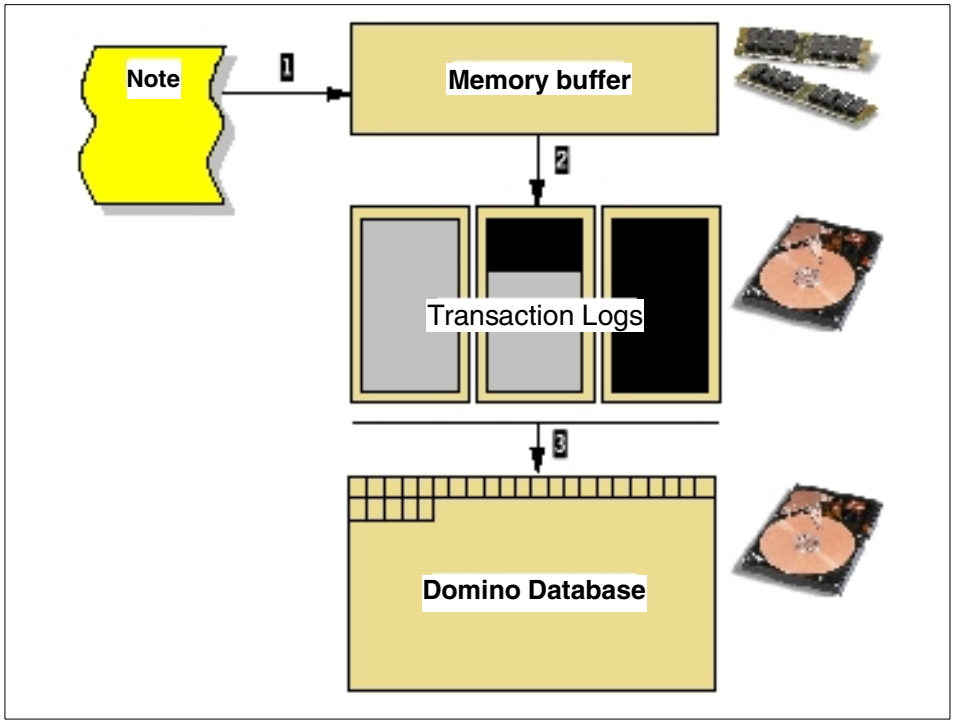


Figure 21. Transaction logging writing process

Let us say that a user writes a document to a database. The transaction logging writing process flows as follows:

1. When the new document is saved, Domino saves it to a memory buffer.
2. When the memory buffer is needed for other data, the contents are posted to the transaction log.
3. At some later time, determined by the system, the logged event is written to the actual Domino database.

Should a failure occur in the interim, the transaction log is used to recover the affected databases. Depending on the nature of the failure, a restored copy of the database might also be required.

When Domino uses transaction log data to restore or recover databases, it performs these basic steps:

1. Analyzes the contents of the transaction log
2. Performs all unrecorded changes for committed transactions (redo)
3. Removes all recorded changes for uncommitted transactions (undo)

This leaves the affected database in a clean and consistent state that reflects all operations performed on it. No generalized “state checking” of databases is required, so recovery times are much shorter.

Domino R5 provides a backup and recovery API, which explores transaction logging characteristics and provides pre-image data buffers for products such as Tivoli Data Protection.

If you need a specific point-in-time restore, the most recent backup before that time is used for the restore. Then, if the database is logged, transactions committed prior to the specified point in time can be applied.

The exact procedures involved in database restoration and recovery depend on the backup utility used. But whatever your production environment looks like, server restart will be faster with transaction logging enabled.

Transaction logging does not provide a way to recover individual documents. If you need this, then you will have to recover the database containing the documents, to an alternate name. Then, the Notes client can be used to copy the desired documents from the restored database.

You can find more information on how transaction logging works in *Lotus Domino for S/390 Release 5: Problem Determination Guide*, SG24-5599.

3.6 Using multiple time zones

Some companies may have users in different time zones accessing one Domino system. The time parameter can be set in several places, making it possible for a company to do this.

3.6.1 OS/390 system clock

The OS/390 system clock tells OS/390 what time it is. It is set to GMT (Greenwich Mean Time, also called Universal Time Coordinated (UTC)). The local time is set in the clock statement of the IEASYSxx member in the SYS1.PARMLIB.

Running a multisystem sysplex requires utilizing the sysplex timer (9037), which is an external time reference. This timer is also set to GMT and, in SYS1.PARMLIB(IEASYSxx), is set to local time using the clock statement.

3.6.2 Time in USS

The time zone must also be set in USS to set the correct time for applications running under USS and USS itself. This can be done in several different files. In /etc/init.options, the TZ parameter is used to set the time for all processes that are active during USS kernel initialization. This is mandatory. So is the TZ parameter in /etc/profile, which specifies the time zone for all shell users. A user can override this in their .profile file in their HOME directory, if they wish. Optionally, the TZ parameter can be set in /etc.rc, which will set the time for jobs started in this script file at initialization.

Verifying the time set can be done by issuing the following commands:

- `date` (local time)
- `date -c` (machine time)
- `date -u` (GMT time)

The time set in OS/390 is independent from the time set in USS, but we recommend keeping the OS/390 time and the USS time the same.

3.6.2.1 Domino time zone parameter

Keep in mind that there is also a time zone parameter in the notes.ini file in Domino. As of Release 5, this parameter has no effect when running Domino under OS/390. Domino obtains the time from the TZ parameter in /etc/profile. Setting this parameter will have only a temporary effect, until the next recycle of the Domino server.

3.6.3 Multiple time zones

If you want to run different time zones from one location, you dedicate at least one LPAR to each time zone. In that LPAR, the OS/390 clock and the USS time should be the same.

3.7 Using mail policies and quotas

Most users of an e-mail system are not aware of the system resources involved in delivering and storing their e-mail. When users are unaware of this, their mail file sizes will continue to grow. Users often regard e-mail as an archive, or sometimes even as an extended LAN storage space. From an

administrative point of view, there are several reasons why the sizes of mail files, and the e-mail messages, need to be restricted.

- The indexer task requires more resources to index a large mail file than it does for a small one.
- Unlimited mail files require very intensive monitoring of disk space, as the usage varies greatly between users. This makes it difficult to predict how much disk space a certain number of users require, so updates need to be done frequently.
- Sending large e-mails over limited bandwidth networks can cause delays for the delivery of other e-mails, or for other services sharing the same connection.
- Large mail files are more difficult to move to a different server over a limited bandwidth network connection.

To overcome these problems, a service level agreement (SLA) with the user community needs to be drawn up, together with a set of tools to enforce the SLA.

3.7.1 SLA for usage

When drawing up an SLA with the user community about the usage of e-mails, the following items need to be addressed:

- Maximum size of the mail file. This is the size of the active mail file. To simplify administration, it is important to set one standard that applies everywhere within your organization.
- Alternatives need to be offered for archiving mail (sometimes due to legal obligations) and sharing of mail (mail-in databases).
- To make users aware of the extra resources they require, billing of the above mentioned alternatives needs to be in place. However, keep in mind that although you can set very advanced billing parameters, such as MB sent/received or the number of reads of the mail file, it is best to keep it simple. For instance, you can charge per 100 MB of required archiving space regardless of wether it is used or not.
- Maximum message size per destination. Before you can set this limit, you need to make sure you are able to enforce it at the destination you require. This is highly dependent upon your mail routing topology.
- Message sizes that will be reduced to low priority and sent outside of the office hours time frame. Again, you need to check you mail routing topology to see where you can make this applicable.

- Also, there needs to be an alternative for these message size restrictions, in case users need to move large files.
- Usage of agents. When users create personal agents in their mail file, they can often cause problems with mail routing such as mail routing loops. End users are usually not qualified to write agents. You should, therefore, limit the use of agents to a limited number of predefined agents in their mail file.

3.7.2 Tools to enforce the SLA

It is essential that you can enforce everything that is drawn up in the SLA.

To help you do this, there are several settings in the server document that you can use, as well as setting quotas on the mail files themselves. For more information about how to implement these mail policies and their alternatives, see 9.1, “Mail policy implementation” on page 141.

Part 2. Administration and operations

Part 2 reveals the secrets of operating, administering, and monitoring a large Domino system on OS/390. It describes backup and recovery possibilities, problem determination, maintenance of software, as well as hints and tips. Sample shell scripts, Notes API programs, and LotusScript agents are also provided, to assist you with administration.

Chapter 4. Performance

This chapter gives you information on how to tune Domino and OS/390 to get the most out of the servers in your large Domino domain. The information in this chapter consolidates part of the detailed guidance provided by the Redbook *Lotus Domino for S/390 Release 5: Installation, Customization and Administration,* SG24-2083*, Lotus Domino for S/390 Release 5: Performance Tuning & Capacity Planning*, SG24-5149, and *Lotus Domino for S/390 Release 5: Problem Determination Guide*, SG24-5599.

For performance problems, see 6.2, “Performance” on page 123.

When looking at performance, you also need to look at capacity and tuning, by monitoring the resources used in your system. The following picture shows the flow of the initial sizing, the monitoring, and either the tuning actions to be undertaken or the adding of additional resource to adjust the initial sizing.

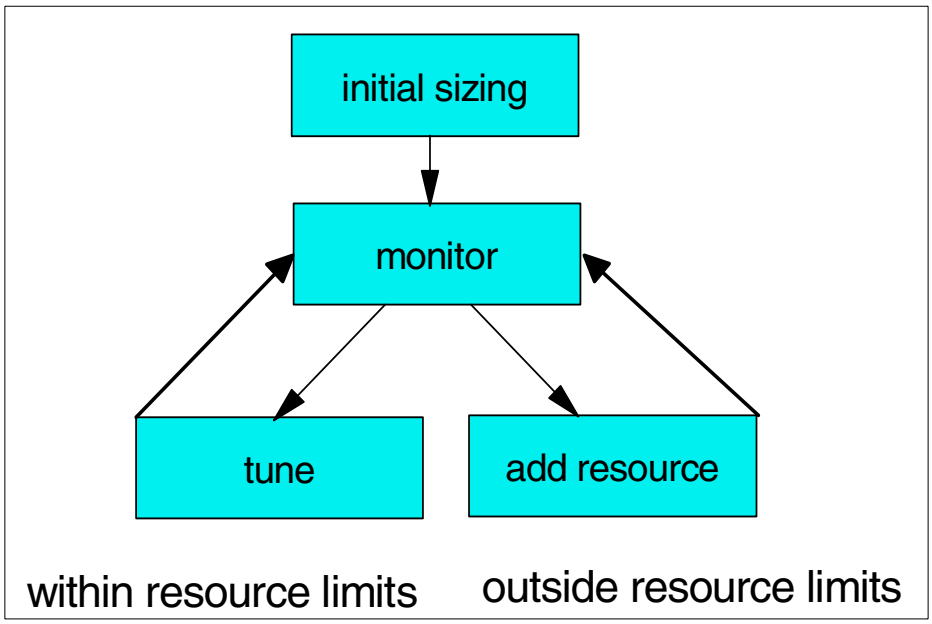


Figure 22. Process diagram of monitoring and tuning your domino system

This chapter contains the following sections:

- Trends—the important indicators to watch for. When do you need more CPU, when do you need more DASD space, or when do you need more servers?

- Monitoring techniques and tools—what tools are available and how are these used to monitor the trends?
- Hints and tips—what can you do?

4.1 Trends

To analyze trends, you must be able to recognize the characteristics of the workloads on your servers. If you have a server that serves mail for Notes clients and Web clients, as well as provides Notes application services, it may be difficult to determine what causes this server to use resources excessively.

Therefore, separate your workload by type, as shown in Table 1.

Table 1. Relating system usage to workload type

Workload Type	System Usage
Mail	I/O subsystem
Groupware	CPU, I/O subsystem, network
Web applications	CPU, network

Run separate servers for mail and applications. You may even be required to split mail servers because of the type of clients that are supported, such as Notes clients and Web clients. Also, various types of Web e-mail services, such as POP3 or IMAP, may have different resource usage characterizations, different trends to analyze, and most important, different thresholds to manage.

Some trends to analyze include:

- Number of registered users
- Number of active users
- Transaction rate generated by the users
- Transaction response times
- Average CPU time used per transaction
- Average memory needed per user
- Amount of data you have, including the activity
- Amount of network traffic

Use the guidelines provided in this redbook and in *Lotus Domino for S/390 Release 5: Performance Tuning & Capacity Planning*, SG24-5149, to make the initial estimate of your workload.

At IBM Watson Research, William “Bucky” Pope has investigated the behavior of Notes users for capacity planning.

He has concluded that transactions are a much better trend to monitor than the number of users. He has written several papers on this subject, which were presented at CMG conferences and SHARE/GUIDE. We have listed some of Mr. Pope’s papers in the additional material provided with this redbook. (See Appendix D, “Using the additional material” on page 193.)

They include:

- *Planning Domino E-mail Servers using Notes Transactions* presented for CMG
- *A Planning Model for Lotus Notes Applications* presented at CMG
- *Planning Realistic Performance Test for Domino E-mail Servers using Notes Transactions*, published by Bucky Pope for the 11th software testing symposium in Sydney, Australia on October 1998.

These documents explain the use of Notes transactions to model the workload and capacity planning for a Domino server. Especially for Notes clients, the workload for a server has become generated from the processes that run on a user’s desktop computer, rather than initiated directly by user activity. Also, a server should have a mix of heavy, average, and light work being generated by user’s desktops, regardless of the users, as they may even be in meetings when their desktop generates server activity.

Start by collecting data for trending your workload, and keep this data for trending the changes in your workload as users learn and use the capabilities of Notes. Use either Table 2, or Table 9 on page 177, as a guide.

Table 2. Worksheet for trends to monitor by type of work

service type/ usage	Notes Mail	Notes Application	HTTP	HTTP POP3	HTTP IMAP	UsageRef to
number of users		X	X	X	X	
percent active		X	X	X	X	“Activity Trends tool” on page 71
number active users						“Activity Trends tool” on page 71

service type/ usage	Notes Mail	Notes Application	HTTP	HTTP POP3	HTTP IMAP	UsageRef to
Transaction rate						“Activity Trends tool” on page 71
Transaction Response time						See “RMF postprocessing” on page 65
CPU						See “RMF postprocessing” on page 65
DASD space						“Activity Trends tool” on page 71
I/O activity						
Memory						
Network bandwidth						“Activity Trends tool” on page 71 and See “RMF postprocessing” on page 65

The last column of this worksheet refers to sections in this document where you can find more information.

4.2 Monitoring techniques and tools

Both Domino and OS/390 provide statistical data on the behavior, usage, and load of the operating system and the Domino servers that run on it. Use information from both sources to determine whether to tune your workload or to add resources to accommodate the workload.

Two sections to read for the best tools to gather your trend data, are the “RMF postprocessing” on page 65 and the “Activity Trends tool” on page 71.

4.2.1 RMF postprocessing

Use RMF postprocessing to extract SMF record types 70-79 and 108, and to create reports for CPU, Storage, DASD, I/O, paging, WLM, OMVS, HFS, Domino, and cache. For more on how to do this, see these resources:

- *Lotus Domino for S/390 Release 5: Performance Tuning & Capacity Planning*, SG24-5149, Chapter 3, DASD subsystem tuning and Chapter 7, Domino Monitoring
- *RMF Report Analysis,* SC28-1950

Starting with APAR OW44845, the RMF postprocessor Lotus Domino server report accepts SMF record type 108. It provides feedback on Domino server load, as well as the number and types of messages that the server handled. For more information, visit the following Web site:

`http://www.s390.ibm.com/os390/bkserv/r10pdf/rmf.html`

Check the reports to make sure there is no paging.

4.2.1.1 The RMF Domino report

This report consists of the following two parts.

- Lotus Domino server summary (shown in Figure 23)

  The summary contains one line for each server that is part of the report.

- Lotus Domino server details (shown in Figure 24 on page 66)

  This part consists of 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)

L O T U S D O M I N O S E R V E R S U M M A R Y PAGE 1

OS/390 SYSTEM ID LN21 DATE 01/30/2000 INTERVAL 30.00.000
RPT VERSION 02.10.00 TIME 18.40.00

SERVER NAME	AVAILABLE HHH.MM.SS	USERS CONNECTED	USERS ACTIVE	TASKS	TRANSACTION RATE	ASYNC READS	I/O RATE WRITES	MAIL RATE DELIVERED	MAIL RATE SENT	SMTP RATE READS	SMTP RATE WRITES
SUT1/COCPOK	000.30.00	2036	18	2136	62.50	137.0	101.9	4.42	0.46	0.00	0.00
BLUED1/BIGBLUE	000.30.00	5034	32	3532	119.31	207.4	199.3	9.14	1.04	0.00	0.00

Figure 23. Lotus Domino server report: Summary

```
                                L O T U S   D O M I N O   S E R V E R   D E T A I L S
                                                                                                                                  PAGE  2
          OS/390                  SYSTEM ID LN21            DATE 01/30/2000           INTERVAL 05.00.000
                                  RPT VERSION 02.10.00      TIME 18.40.00

NAME: SUT1/COCPOK

--- USER ACTIVITY ---    ------ TASKS ------    ------------- MESSAGES -------------    --- ACCESS RATES ---    --- DATABASE CACHE ---
MAX                0    MAX            2148    MAILBOXES                      1    AS I/O READ    139.3    STATUS              OK
CONNECTED       2034    CURRENT        2148                     COUNT  RATE AVG SIZE  AS I/O WRITE   99.41    MAX ENTRIES        516
ACTIVE            18    MAX UPDATES       0    MAIL DELIVERED    1274  4.25        3  POP3 READ       0.00    CURRENT ENTRIES    216
 WITHIN  1 MIN   361    MAX REPLICS       0    MAIL SENT          211  0.70        1  IMAP READ       0.00    HIGH WATER MARK    774
 WITHIN  3 MIN  1098    COUNT REPLICS     0    SMTP RECEIVED        0  0.00        0  HTTP READ       0.00    INITIAL DB OPENS   408
 WITHIN  5 MIN  1287                           SMTP SENT            0  0.00        0  HTTP WRITE      0.00    REJECTIONS           0
 WITHIN 15 MIN  1814                                                                                          HITS               379
 WITHIN 30 MIN  2001
                        - VIRTUAL THREADS -    - PHYSICAL THREADS -                   --- AVAILABILITY ---    -- NSF BUFFER POOL ---
                        MAX            2034    MAX              76                    THRESHOLD          0    MAX              44032
                        CURRENT        2034    CURRENT          27                    INDEX             98    CURRENT              0
                                               TOTAL           100

-------------------------------------------------------------------------------------------------------------------------------------
                                                TRANSACTION ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------------

MAXIMUM NUMBER OF CONCURRENT SESSIONS:  NO LIMIT

------------------------ TOP-10 BY COUNT ------------------------    ------------------------ TOP-10 BY R/T TOTAL -------------------
TYPE NAME                     COUNT %TOTAL  RATE    R/T      R/T    TYPE NAME                     COUNT %TOTAL  RATE    R/T      R/T
                                                      AVG    TOTAL                                                          AVG    TOTAL

     TOTAL                    19560  100.0 65.20  22193  434090K         TOTAL                    19560  100.0 65.20  22193  434090K

   6 OPEN_NOTE_RQST            3670  18.76 12.23  23617 86674928       59 NIFOPENNOTE_RQST         2175  11.12  7.25  79485  172880K
  59 NIFOPENNOTE_RQST          2175  11.12  7.25  79485  172880K       8 UPDATE_NOTE_RQST_ALT     1266   6.47  4.22  68560 86796448
   8 UPDATE_NOTE_RQST_ALT      1266   6.47  4.22  68560 86796448       6 OPEN_NOTE_RQST           3670  18.76 12.23  23617 86674928
  55 READ_ENTRIES_RQST         1225   6.26  4.08  275.2   337168     142 START_SERVER_RQST         580   2.97  1.93   149K 86470752
  49 CLOSE_COLLECTION_RQST     1185   6.06  3.95  11.75    13926      51 OPEN_COLLECTION_RQST     1151   5.88  3.84  413.5   475961
  48 CLOSE_DB_RQST_ALT         1183   6.05  3.94  18.44    21815      55 READ_ENTRIES_RQST        1225   6.26  4.08  275.2   337168
   1 OPEN_DB_RQST              1163   5.95  3.88  26.06    30306      15 SEARCHSTART_RQST          135   0.69  0.45   2174   293431
  51 OPEN_COLLECTION_RQST      1151   5.88  3.84  413.5   475961     128 UPDATE_FOLDER_RQST        623   3.19  2.08  76.58    47707
  34 READ_OBJECT_RQST          1030   5.27  3.43   3.53     3635       1 OPEN_DB_RQST             1163   5.95  3.88  26.06    30306
 111 SERVER_TIME_LITE_RQST      860   4.40  2.87   1.20     1030      48 CLOSE_DB_RQST_ALT        1183   6.05  3.94  18.44    21815

-------------------------------------------------------------------------------------------------------------------------------------
                                                PORT ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------------
MAXIMUM NUMBER OF CONCURRENT SESSIONS:   65535
SESSION TIMEOUT:                         63744

NAME     -- SESSIONS IN --  -- SESSIONS OUT --   SENT  RECEIVED
           COUNT  RATE        COUNT  RATE        (KB)      (KB)

TCPIP        580  1.93           266  0.89      117815      2620
```

Figure 24. Lotus Domino server report: Details

4.2.2 Daily continuous monitoring: SDSF

Use SDSF for daily monitoring of the systems where your Domino servers are running. Figure 25 on page 67 shows an example of an SDSF display active panel.

4.2.2.1 SDSF Display Active panel

```
SDSF DA ML96  ML96      PAG    0 SIO  7819 CPU  72/ 47  LINE 1-16 (121)
COMMAND INPUT ===>                                            SCROLL ===> PAG
PREFIX=**  DEST=(ALL)  OWNER=*  SORT=CPU%/D
NP   JOBNAME  STEPNAME PROCSTEP JOBID    OWNER    C POS DP REAL PAGING    SIO
     ML96SER  STEP1             STC10121 ML96NP     NS  FB 7645   0.00 563.88
     ML96SM1  STEP1             STC10153 ML96NP     IN  FB 9592   0.00 2188.8
     ADSMM6   ADSMM6   SERVER   STC11417 ADSM       NS  E7 5232   0.00 427.63
     ML96UP1  STEP1             STC10146 ML96NP     IN  FB 1886   0.00 319.05
     OMVS     OMVS     OMVS                         NS  FF  66T   0.00   0.00
     ML96AM1  STEP1             STC10144 ML96NP     IN  FB 3135   0.00  82.59
     OTCP     OTCP     TCPIP    STC10112 OTCP       NS  FE 1772   0.00   0.00
     ML96RO1  STEP1             STC10124 ML96NP     NS  FB  12T   0.00 186.56
     RMFGAT   RMFGAT   IEFPROC  STC11759 RMFGAT     NS  FE 1817   0.00   0.00
     ML83UP1  STEP1             STC10130 ML83NP     IN  FB 1468   0.00 184.47
     ML83NP9  STEP1             STC11784 ML83NP     IN  FB  736   0.00 347.55
     ADSM83A  ADSM83A  *OMVSEX  STC10119 ADSM       IN  E7  780   0.00 405.83
     ADSM96A  ADSM96A  *OMVSEX  STC10789 ADSM       IN  E7  959   0.00 382.35
     ADSM96G  ADSM96G  *OMVSEX  STC10796 ADSM       IN  E7 1356   0.00 372.29
     ADSM96J  ADSM96J  *OMVSEX  STC10803 ADSM       IN  E7 1123   0.00 362.23
     ADSM96F  ADSM96F  *OMVSEX  STC10794 ADSM       IN  E7 1130   0.00 308.56
```

Figure 25. SDSF DA panel

Review display active (DA) continuously for tasks that may be using a larger amount of CPU than normal. By monitoring this information on a consistent basis, you will develop skills and information to be able to determine what “normal” usage is.

4.2.3 RMF Monitor III

Use RMF Monitor III, and possibly Omegamon, for daily continuous monitoring of the systems where your Domino servers are running. This section contains panels from RMF Monitor III and points to the items to watch for. Figure 26, shows the RMF Monitor III System information panel.

```
 RMF 2.7.0  System Information            Line 1 of 30
 Command ===>                                          Scroll ===> CSR

 Samples: 100     System: ML96  Date: 10/09/00  Time: 13.56.40  Range: 100   Sec

    -------------- 9672 Version 00 Model ZZ7       ------------ Policy: PRISHFT
    Processor(s) Online:   7          Vector Processors:  0     Date:   08/23/00
    Average CPU Util:     49%         Appl% / EAppl%:  44/ 44   Time:   20.28.02
    Avg. MVS CPU Util:    71%         Partition Name:  ML96
 Group     T WFL --Users--  RESP TRANS -AVG USG-   -Average Number Delayed For -
              %   TOT  ACT  Time  /SEC PROC  DEV   PROC  DEV STOR SUBS OPER  ENQ

 *SYSTEM      49  156    7        0.03  3.3  2.1    3.0  1.5  0.0  0.1  1.0  0.0
 *TSO         83    3    0        0.03  0.0  0.0    0.0  0.0  0.0  0.0  0.0  0.0
 *BATCH        0    0    0        0.00  0.0  0.0    0.0  0.0  0.0  0.0  0.0  0.0
 *STC          45  111    3        0.00  0.9  2.0    1.0  1.5  0.0  0.1  1.0  0.0
 *ASCH         0    0    0        0.00  0.0  0.0    0.0  0.0  0.0  0.0  0.0  0.0
 *OMVS         54   42    4        0.00  2.4  0.0    2.0  0.0  0.0  0.0  0.0  0.0
 *ENCLAVE      0  N/A             N/A  0.0  N/A    0.0  N/A  0.0  N/A  N/A  N/A
 PRIMESTC  W  32   14    2  .000  0.00  0.2  1.0    0.9  0.6  0.0  0.0  1.0  0.0
 STCLOW    S  32   14    2  .000  0.00  0.2  1.0    0.9  0.6  0.0  0.0  1.0  0.0
 PRIMETSO  W  83    3    0  1.99  0.04  0.0  0.0    0.0  0.0  0.0  0.0  0.0  0.0
```

Figure 26. RMF Monitor III system information

Review RMF for any significant delays, especially in processor, storage, and devices. Examine these delays where necessary.

Note: It is not recommended to have the RMFGAT task (Monitor III) active at all times, because it consumes CPU resource. You may want to consider starting this task only when it is needed, and stopping the RMFGAT task as soon as you no longer need it.

4.2.3.1 Processor delays

```
RMF 2.7.0  Processor Delays                Line 1 of 58
Command ===>                                          Scroll ===> CSR

Samples: 100     System: ML96  Date: 10/09/00  Time: 13.56.40  Range: 100   Se

            Service  DLY USG  Appl EAppl ----------- Holding Job(s) ----------
Jobname   C Class     %   %     %     %  %  Name      %  Name      %  Name
ML96SM1   O NOTESSRV  40  54  47.5  47.5 39 ML96SER   25 ML96SM1   17 OMVS
ML96SER   O NOTESSRV  38  99 161.3 161.3 37 ML96SER   24 ML96SM1   13 OMVS
ADSMM6    S STCLOW    34  12   7.3   7.3 33 ML96SER   21 ML96SM1   12 OTCP
ML96RO1   O NOTESSRV  24  20  17.2  17.2 24 ML96SER   13 ML96SM1   12 ML96RO1
ML96UP1   O NOTESSRV  15  12   8.4   8.4 15 ML96SER   10 ML96SM1    6 OMVS
ML83SER   O NOTESSRV  11  19  13.7  13.7 10 ML96SER    8 ML96SM1    5 AUTONET
ML83UP1   O NOTESSRV   8  12   6.1   6.1  8 ML96SER    5 ML96SM1    4 OMVS
ML96SER   O NOTESSRV   8   2   0.9   0.9  8 ML96SER    4 ML96RO1    4 ML96SM1
ADSM96B   S STCLOW     8   2   1.1   1.1  7 ML96SER    4 ML96SM1    3 OTCP
ADSM96E   S STCLOW     8   0   0.4   0.4  8 ML96SER    6 OTCP       5 ML96SM1
ML83RO1   O NOTESSRV   7   7   3.7   3.7  7 ML96SER    3 ML96SM1    3 OMVS
ADSM83A   S STCLOW     6   1   0.7   0.7  5 OTCP       5 ML96SER    3 ML96SM1
ADSM96H   S STCLOW     6   0   0.5   0.5  6 ML96SER    3 ML83SER    2 OTCP
ADSM96A   S STCLOW     5   0   0.5   0.5  4 ML96SER    3 OTCP       3 ML96SM1
```

Figure 27. RMF Monitor III processor delays

Use the Processor Delays panel, shown in Figure 27, to determine that the Domino server tasks, such as ML96SER, are the primary jobs holding up any other jobs on the system. This may not necessarily mean that there is a problem, but it indicates that the server tasks are getting the most resource, as they should.

```
RMF 2.7.0  Channel Path Activity           Line 1 of 86
Command ===>                                          Scroll ===> CSR

Samples: 100     System: ML96  Date: 10/09/00  Time: 13.56.40  Range: 100   Sec

Channel Path      Utilization(%)     Read(MB/Sec)  Write(MB/Sec)
ID  Type SHR    Part  Total    Bus    Part  Total    Part  Total

04  CNC_S Y     0.00   0.01
08  CNC_S Y     0.01   1.67
09  CNC_S Y     1.07   2.95
0A  CNC_S       0.00   3.57
0C  OSA         0.00   0.00
10  CNC_S Y     0.00   0.00
11  CNC_S Y     0.00   0.01
12  CNC_S Y     0.00   0.00
13  CNC_S Y     0.00   0.00
14  CNC_P Y     0.00   0.00
17  CVC         0.00   0.11
18  CNC_S Y     0.00   0.00
1A  CNC_P Y     0.00   0.00
1B  CNC_S Y     0.00   0.00
```

Figure 28. RMF Monitor III Channel Path Activity

Use the RMF Monitor III Channel Path Activity panel, shown in Figure 28, to monitor for significant increases in CHPID utilization, which could lead to a channel bottleneck or a contention problem.

```
RMF 2.7.0  I/O Queuing Activity           Line 1 of 113
Command ===>                                          Scroll ===> CSR

Samples: 100     System: ML96  Date: 10/09/00  Time: 13.56.40  Range: 100   Se

                                        Cont  Del Q   %All   CHPID   %DP    %CU
Chan Paths Control Units        LCU     Rate  Lngth  Ch Busy Taken   Busy   Busy

3C         4300                 001D    0.0   0.00   0.02    8.85   0.00   0.00
3E         4300                 001D                         9.29   0.00   0.00
8E         4300                 001D                         8.72   0.00   0.00
8F         4300                 001D                         9.00   0.00   0.00
BC         4300                 001D                         9.39   0.00   0.00
BE         4300                 001D                         9.44   0.00   0.00
D6         4300                 001D                         9.03   0.00   0.00
D7         4300                 001D                         9.21   0.00   0.00
3C         4301                 001E    0.0   0.00   0.02    8.58   0.00   0.00
3E         4301                 001E                         8.68   0.00   0.00
8E         4301                 001E                         8.57   0.00   0.00
8F         4301                 001E                         9.01   0.00   0.00
BC         4301                 001E                         9.29   0.00   0.00
BE         4301                 001E                         9.23   0.00   0.00
```

Figure 29. RMF Monitor III I/O queuing activity

Use the RMF Monitor III I/O Queuing Activity panel, shown in Figure 29, to monitor for significant increases in contention rates (Cont Rate), %DP Busy, or **%**CU Busy to ensure that you are not overloading a control unit or a channel.

```
RMF 2.7.0  Common Storage               Line 1 of 158
Command ===>                                          Scroll ===> CSR

Samples: 100     System: ML96  Date: 10/09/00  Time: 13.56.40  Range: 100   Sec

                                  ---- Percent ----   ------- Amount --------
System Information                CSA ECSA SQA ESQA     CSA  ECSA   SQA  ESQA
 IPL Definitions                                     4884K  121M 1236K  158M
 Peak Allocation Values            20   48  66   31   955K   58M  821K   49M
 Average CSA to SQA Conversion      0    0               0     0
 Average Use Summary               20   48  66   31   953K   58M  820K   49M
 Available at End of Range         80   52  34   69  3931K   63M  416K  109M

              Service        ELAP  -- Percent Used -   ----- Amount Used -----
Jobname  Act C Class    ASID  Time  CSA ECSA SQA ESQA     CSA  ECSA   SQA  ESQA
%MVS                          11   12  45   26   528K   14M  557K   42M
%REMAIN                        0    0   0    0  12576 76176   128  6640
*MASTER*     S SYSTEM      1   5.6D  5   29  16    3   248K   35M 201K 4229K
VTAM         S SYSSTC    116   5.6D  0    3   0    0  10952 4117K     0   528
OTCP         S SYSSTC    125   5.6D  0    2   0    0   136 2155K     0  9464
NETVNET      S SYSSTC    124   5.6D  0    0   2    0  6872   99K 21272   832
AUTONET      S SYSSTC     25   5.6D  0    0   1    0  4032  246K 16728   848
```

Figure 30. RMF Monitor III common storage

Use the RMF Monitor III Common Storage panel, shown in Figure 30, to periodically review the CSA/ECSA and SQA/ESQA figures. Make sure that they are not being fully used. As with any other OS/390 system, depletion of this storage can lead to an operating system crash that requires an IPL.

```
RMF 2.7.0  Device Resource Delays        Line 1 of 262
Command ===>                                          Scroll ===> CSR

Samples: 100     System: ML96  Date: 10/09/00  Time: 13.56.40  Range: 100   Sec

Volume S/   Act  Resp  ACT CON DSC PND %,  DEV/CU             Service  USG DLY
  /Num  MX  Rate Time   %   %   %  Reasons Type     Jobname  C Class    %   %

SK4612 S    51.4 .008    9   2   7 PND    0 33903    OMVS     S SYSTEM    10  33
  4612   4                                  2105
SK4616 S     654 .002   35  30   1 PND    4 33903    OMVS     S SYSTEM    89  29
  4616   4                                  2105
Z57520      19.5 .051   98   5  80 PND   13 3490L    ADSMM6   S STCLOW    71  23
  0F48
Z52331      23.2 .042   97   6  78 PND   13 3490L    ADSMM6   S STCLOW    72  22
  0F21
SK4614 S    87.5 .003    6   2   3 PND    1 33903    OMVS     S SYSTEM    13  18
  4614   4                                  2105
Z52389       6.4 .151   97   2  89 PND    6 3490L    ADSMM6   S STCLOW    76  16
  0F45
Z53509      11.2 .090  101   3  88 PND   10 3490L    ADSMM6   S STCLOW    79  15
  0F41
```

Figure 31. RMF Monitor III device resource delays

Use the Device Resource Delays panel, shown in Figure 31, to review the activity rate and response time of the various volumes on the system to make sure there are no bottlenecks with a specific HFS.

4.2.4 Activity Trends tool

The Activity Trends tools is a sample Notes application that can be used to gather trends data using a Notes database. The Trends application is a prototype of a function that could be added to a future release of Domino. The application as included with this redbook is “alpha” code with no warranty expressed or implied. Figure 32 provides an explanation of this trend data-gathering tool.

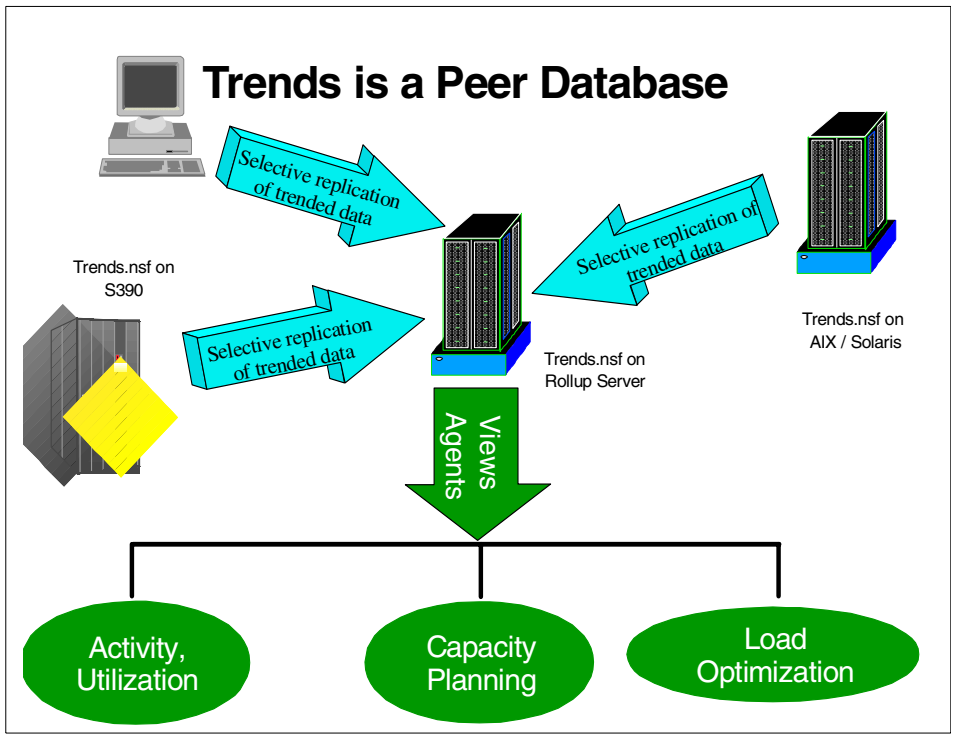


Figure 32. Overview of Activity Trends collection

The purpose of the Activity Trends database is to record and report statistics which portray the activity of users (clients) against the databases on the Lotus Notes server where this database resides. This data aids in the administration of Lotus Notes servers by:

- Providing awareness of server load to the administrators in charge.
- Providing trends and statistics to intelligent algorithms which can be fully automated, or providing computer-aided load balancing of a set of servers.

The primary source of the activity data collected is the standard Lotus Notes Log database (log.nsf). Specifically, two sets of documents found in the Notes Log are used:

Activity Log documents which provide database activity statistics regarding reads, writes, and uses, as well as providing basic database information such as path and disk space statistics.

Session Log documents which detail individual user (client) sessions with the server, provide the transactions, bytes sent to/from the server, elapsed time, list of any databases connected to, along with the connect time to each.

This database collects data in the following categories:

Trends Represents an ever-growing number of data observations which have been summarized to provide a statistical trend. Each trended data item is a weighted average of the observations with more recent data having more influence. The trended data mean is accompanied by the necessary data components to enable historical trending (without maintaining all the detail data points), specifically: total of all observations, weighted average of the squares, and standard deviation of the observations. For comparison, the most recent observations are maintained.

Daily To allow detail analysis of individual data observations, a configurable number of daily observation details are maintained. These daily details are individually dated versions of the same data that is aggregated to form the trends data.

History The server trends are maintained historically so that long-range trends for the server can be viewed more readily.

A useful graphing facility is built into this database in the form of a Java agent which can be invoked from a view button in any of the server trends views. Since it is a Java agent, it requires a R4.6 Notes client, or newer, to invoke this graphing facility.

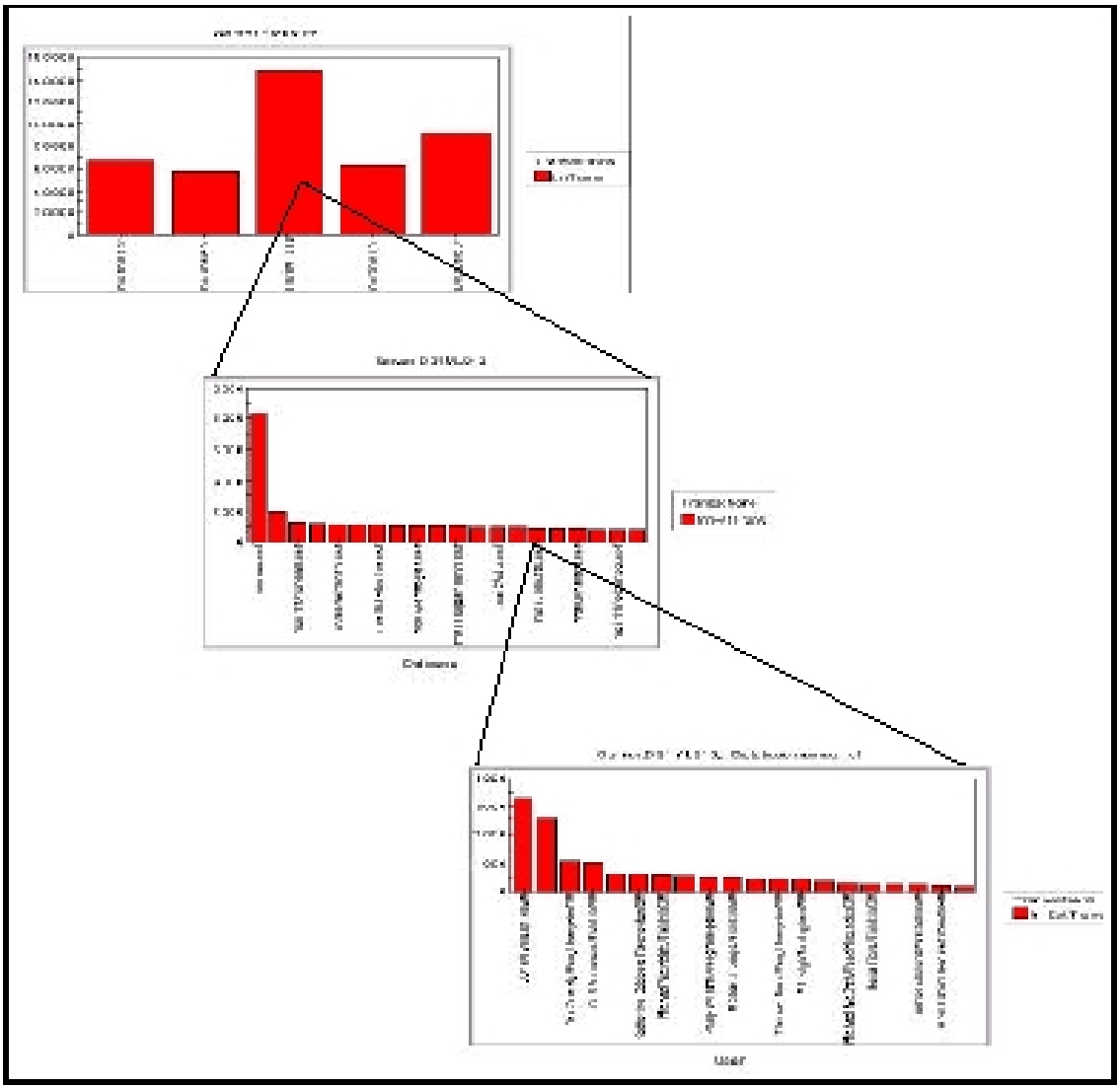


Figure 33. Trends: Java agent to graph trends

The graph trends agent presents key portions of the data in the trends database in graphical form. When graph trends is started, it creates a chart summarizing the statistics for servers in the view, as shown in Figure 33. Different fields in the server data can be viewed using the data selection buttons on the left of the chart. Commonly-used data views, such as *transactions* and *reads and writes* have predefined buttons. By selecting the data view marked *custom*, any of the fields in the trends database document

can be charted by choosing them in the selection list immediately below the custom button. This custom option can be a little slow the first time it is activated.

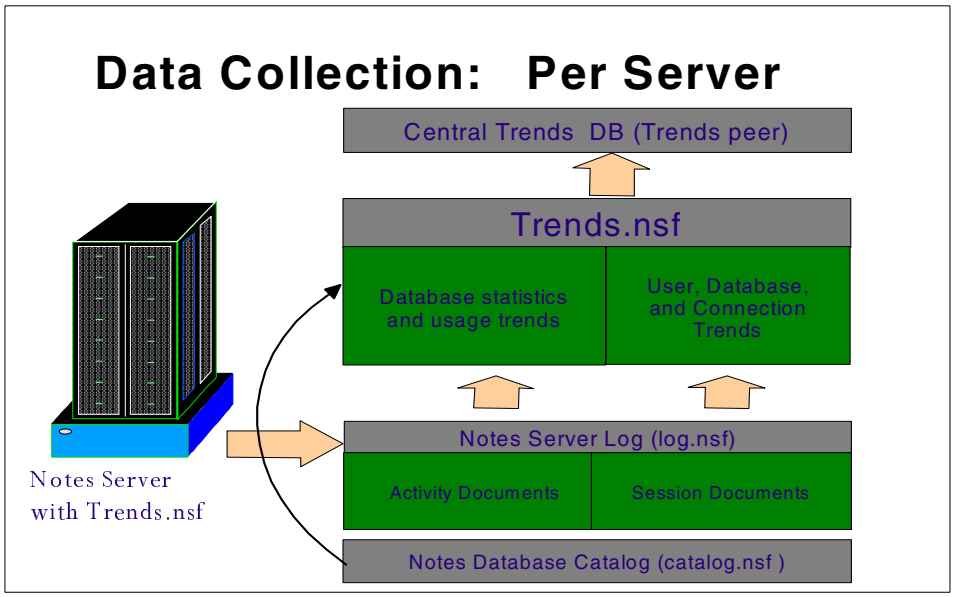


Figure 34. Trends data collection per server

Figure 34 shows how this database is intended to run on each server of interest in a Lotus Notes domain, for collecting data locally. Each of the Activity Trends databases can be replicas of each other, so that the trended data can be collected centrally with Lotus Notes (selective) replication. This allows an administrator or automated agent to have a focused collective view of a group of servers so that load balancing analysis can be performed. Also included in this database is a Java agent that provides Notes administrators with a graph of the aggregate set of server trend data, giving a better understanding of activity and load on these servers. This agent also performs load/space balancing across a set of selected servers. It includes the ability to add servers to, or remove servers from, the load/space balancing computations.

Trends is available as a sample in *Lotus Notes and Domino 5 Scalable Network Design: Web Server Network Infrastructure, SR24-9218.*

Note: The function of the Trends database may be extended to the HTTP task, and Trends may be included in a future release of Domino.

Server Trends

Server Name: SC67ADM01/H/Itso **Notes Build:**
Server Platform: **Notes Version:**

Server Database Activity Summary

Database Trend History: 4 observations, last observation on 10/09/2000

	Trended Mean	Last Observation	St Dev	CoV
Databases:	38	33	2	0.05
Active Databases:	8	4	2	0.22
DiskSpace:	99,784,904	76,212,736	8725945	0.09
DiskUsage:	95,352,953	71,979,264	8564789	0.09
Percent Used:	95.6%	94.4%		
Database Uses:	39	62	7	0.18
Database Reads:	156	476	195	1.24
Database Writes:	72	108	166	2.32
Database Read/Write Ratio:	2.2	4.4		

Figure 35. Trends sample database activity summary

Figure 35 shows an example of the output from a Trends database activity summary report.

Server User Session Summary

Sessions Trend History: 5 observations, last observation on 10/09/2000

	Trended Mean	Last Observation	St Dev
Unique Users:	3	7	1
• Prime Users:	4	6	1
Sessions:	44	18	77
• Prime Sessions:	26	13	38
• Sessions with No Connections:	0	4	1
• Prime Sessions with No Connections:	1	4	1
Session Seconds:	16,212	6,337	6441
• Prime Session Seconds:	15,981	4,013	6837
• Sessions with No Connections :	1	4	1
Server In Use %:	32.6%	10.1%	
• Prime Server In Use %:	56.6%	15.2%	
Concurrent Use Peak:	1.1	1.2	0
Concurrent Use Average:	0.2	0.1	0
• Prime Concurrent Use Average:	0.4	0.1	0
Transactions:	2,786	1,267	4560
• Prime Transactions:	1,941	1,087	2717
• Sessions with No Connections:	2	16	6
Transaction Rate Peak:	68	52	122
Transaction Rate Average:	2	1	3
• Prime Transaction Rate Average:	3	2	4
Connections:	644	23	1200
• Prime Connections:	354	17	566
User Reads:	121	28	245
• Prime User Reads:	121	25	245
User Writes:	145	19	289
• Prime User Writes:	145	19	289
Sent To Server:	2,384,900	2,891,027	4373349
• Prime Sent To Server:	2,312,267	2,877,097	4244764
• Sessions with No Connections:	423	3,238	953
Sent From Server:	3,113,133	4,367,278	4746912
• Prime Sent From Server:	2,888,922	3,341,870	4515135
• Sessions with No Connections:	34,495	266,780	76687
Network I/O Rate Peak:	4,879	13,460	9833
Network I/O Rate Average:	63	84	100
• Prime Network I/O Rate Average:	127	157	204

Figure 36. Trends sample user session summary

Figure 36 shows an example of the output from a Trends user session summary report.

Configuration Navigator Graph Trends

Server Name	Databases	Active Databases	Disk Space KB	% DS Used	R/W Ratio	Last Observation
SC67ADM01	38	8	97,446	96	2.2	10/09/2000
SC67M001	51	9	171,576	95	6.4	10/09/2000

Figure 37. Trends sample database activity per server

Figure 37 shows an example of the output from a Trends activity per server report.

Server Name	Users	Sessions	Sessions NC	Connects	Transactions	In Use %	Bytes To Server	Bytes From Server	Last Observation
SC67ADM01	2	30	3	251	2,949	16.3%	2,919,171	8,901,688	10/09/2000
SC67M001	3	44	0	644	2,786	32.6%	2,384,900	3,113,133	10/09/2000

Figure 38. Trends sample user activity per server

Figure 38 shows an example of the output from a Trends user activity per server report.

Connection Trends

Server: SC67M001/M/ltso **User Name:** Manfred Hauff/DE/ltso
Last Active: 10/09/2000 **Pathname:** mail\MHauff.nsf

Sessions Trend History: 5 observations, last observation on 10/09/2000

	All Sessions		Prime Shift Sessions	
	Trended Mean	Last Observation	Trended Mean	Last Observation
Connections:	1	3	1	2
Connect Seconds:	16408	5136	16262	3025
Reads:	3	6	3	3
Writes:	0	0	0	0
Transactions:	55	118	54	96
Sent To Server:	5183	8547	5073	6949
Sent From Server:	252383	668649	249421	625683

Figure 39. Trends: Connection trends details

Figure 39 is an example of a Trends report showing connection details.

Database Pathname	Connects	Connect Seconds	Reads	Writes	Transactions	Bytes To Server	Bytes From Server	Last A
SC67ADM01	1,341	11,588	639	308	7,256	3,267,154	29,148,501	
Gerard de Bruyn	9	4,065	16	14	743	60,552	16,102,280	
Manfred Hauff	2	131	7	0	105	6,163	467,933	
Nico Out	28	7,319	226	283	2,261	2,843,937	6,578,331	
SC53A001	3	59	380	0	763	36,078	5,376,014	
SC67M001	1,298	13	10	11	3,385	320,423	623,943	
SC67M001	802	22,277	527	377	6,011	5,084,431	13,552,851	
Gerard de Bruyn	7	509	6	4	249	314,740	1,278,124	
Izzat Tuqan	2	1,729	3	0	76	5,488	824,653	
Manfred Hauff	4	16,421	3	0	66	6,288	257,343	
Max Santos	4	78	0	0	128	9,876	971,358	
Nico Out	32	3,532	381	359	3,151	4,476,849	9,727,109	
SC67ADM01	753	8	134	15	2,342	271,190	494,264	
	2,143	33,865	1,165	685	13,267	8,351,585	42,701,352	

Figure 40. Trends: Connects by user

Figure 40 is an example of a Trends report showing user connection details.

4.2.5 The OS/390 system log

Review SYSLOG for error messages regarding OMVS, TCP/IP, and OS/390 Domino console support (called DOMCON prior to R5.03).

Domino runs under UNIX System Services with several distinct tasks. OS/390 writes messages to SYSLOG during Domino server startup and termination, as well as when an error or exceptional condition occurs.

Because MVS messages scroll off the console quickly, operators need access to a TSO terminal with SDSF for viewing and searching through the system log. In a Parallel Sysplex environment, operators can view a merged log of all systems in the sysplex.

The SYSLOG data set shows basic information about the Domino server, such as:

- Domino startup and shutdown
- RACF messages
- Communications errors from TCP/IP, including those related to Domino users
- Abnormal termination of the Domino server or related tasks

From OS/390, view and control of the Domino server is limited. Commands to the Domino server are normally issued by a Domino system administrator. OS/390 has no view of what is happening inside Domino, unless you use

Domino Console support to filter and route Domino server messages to an operator console.

4.2.6 GRS contention

Make sure there is no global resource serialization (GRS) contention.

Contention for a given resource can cause the server to slow down or to hang. Contention may involve OS/390 resources, such as HFS files or disk addresses. Console commands of the GRS component of OS/390 may help to find contentions. The OS/390 operator command `d grs,c` displays existing latch contentions and you can take actions to free them.

4.2.7 OS/390 dumps

Make sure there are no dumps being performed for Notes-related tasks during prime time of your servers. Dumps halt activity, so if you do not need the dumps for determining the cause of a problem, turn dumping off during prime time. A command to do this is shown in Figure 41.

```
SLIP SET,C=U0034,ID=U034,A=NODUMP,AL=(0),JL=(OMVS),
DSPNAME=('OMVS'.SYSZBPX1,'OMVS'.SYSZBPX2),END
```

Figure 41. Turning off SVCDUMPs

4.2.8 Console log (notes.log)

The notes.log file is actually the Notes console log redirected to a text file. Its contents is quite verbose, well suited for debugging purposes. This is the place where you find server PANIC messages. They do not appear in the log.nsf file.

To search the server notes.log for errors, Telnet to the server and go to the Notesdata directory. After you change the path of the noteslog to where your notes.log file is, issue a `tail` command to look at a specific number of previous lines in the notes.log, as follows:

```
cd /domino4/notesdata/noteslog
tail -50 notes.log
```

You could also use `grep` to search a notes.log file for specific texts. Following are some examples you could try.

- Server panics or PANIC messages

  ```
  egrep -n -i "PANIC" notes.log
  ```

- Database corrupt messages

```
egrep -n -i "corrupt" notes.log
```

- General error messages

```
egrep -n -i "error" notes.log
```

4.2.9 Server log (log.nsf)

This is a log that Notes administrators generally look at using the Notes Administration client. It is the primary log for Notes troubleshooting.

Since log.nsf is a Notes database, it is much more organized than the notes.log file, because of the various views available in the database; however, we found that this log does not have as much information as the notes.log file. If the support center asks you to turn on special debug parameters in the NOTES.INI file, the trace information will show up in this log or the notes.log.

There are products such as IntelliWatch that scan the log file for the events that are monitored. Trend.nsf agents scan the log.nsf file for session and database information for trending.

To access log.nsf, do the following.

1. Log onto the Notes Administration client.
2. Click the **Server** tab.
3. Click the **Analysis** tab.
4. Expand the N**otes log** views to analyze documents.

4.2.10 Statistics and events

The Domino server provides services and tasks that create and report information about the Domino system. This information comes in two forms: statistics and events. Statistics show the status of processes running on the system. For example, the statistic “Free space on drive C” indicates the amount of free space available on drive C. Events are generated when something takes place on the system. For example, the event “Replicating files with *servername*” occurs when a file replicates with a specified server. The Domino server continuously updates statistics. To view system statistics at any time, use the `show stats` command at the server console. To obtain statistics to monitor the Domino system, use Collect task, which collects statistics and puts the information into the statistics database (STATREP.NSF).

To view statistic reports, you click **Server** -> **Analysis tab** -> **Statistics Reports** view in the Domino Administrator.

The Domino server also provides monitor documents that you use to configure statistic thresholds. When the Collect task collects a statistic and places it into the Statistics database (STATREP.NSF), it compares the statistic to the threshold configured in the monitor document. The first time the statistic reaches the specified threshold in the monitor document or in the Statistic Names document, an alarm report is generated. An alarm report is nothing more than a document informing you that a statistic has reached its threshold. To set up an alarm, you must specify a collection alarm interval in the Server Statistic Collection document. Alarms documents are automatically created in the **Server** -> **Analysis Tab** -> **Statistics Reports** -> **Alarms** view in the Domino Administrator.

Events happen continuously on the Domino system. To notify you about important system information, the Domino server has a notification facility called the Event task. The Event task sends notification about the event to a destination configured in the Event Notification document.

You can create statistics and events monitoring documents for your particular server and network needs by creating Statistic Monitors and Event Notification documents from the Statistics & Events view on the Configuration tab in the Domino Administrator. If you do not set up an Event Notification document, you will not be notified that an event has occurred.

4.2.11 Event and Collect tasks

The Event task and the Collect task are two of the server monitoring tasks that gather information about the Domino system.

The Event task monitors system activities when you start the Domino server. The Event task also creates the Statistics & Events database (EVENTS4.NSF) on the server, if it doesn't exist.

The Collect task collects and monitors statistics from the servers configured in the Server Statistic Collection document. To start the Collect task, load it on the server designated to collect statistics.

Statistics and Event monitoring now includes the provision of timed probes for the server, mail, and TCP/IP services. These can be used to periodically check whether the server and its services are available and responding to external requests.

4.2.12 IntelliWatch

IntelliWatch Pinnacle Enterprise is an integrated solution for Lotus Notes administrators seeking to automate all aspects of their Notes management.

The IntelliWatch Pinnacle Enterprise Product suite contains the latest versions of the following IntelliWatch products:

- IntelliWatch Monitor
- IntelliWatch Analyzer
- IntelliWatch Performance Manager
- IntelliWatch Trace

All components use the Pinnacle Express Console; therefore you will get the same console interface with each component. The console is a Java applet that runs in the Intelliwatch database on your servers. This means that you can connect to IntelliWatch Pinnacle Enterprise from anywhere with no client overhead, so long as it has a Java-enabled browser. Additionally, IntelliWatch Monitor contains the following products to accomplish its purpose:

- IntelliWatch Management Agent Database (MA)
- IntelliWatch Advanced ServerWatch
- IntelliWatch Utilities (send page, remote recycle, and command line)
- IntelliWatch Crash Detection
- IntelliWatch Paging Server

The IntelliWatch Pinnacle Enterprise suite of products is most effective when used together to provide a comprehensive Notes management solution in a mixed environment of servers. The product is available for AS/400, OS/390, Windows NT, Windows/2000, AIX, and Sun Solaris. It uses a set of server add-in tasks and Notes databases to define and monitor events in the managed servers. For crash detection, a process runs outside the Domino server. It also provides SNMP traps to communicate with SNMP-enabled management products such as Tivoli and HP OpenView. For more information, visit the following Web sites:

`http://www.intelliwatch.com`
`http://www.candle.com`

4.3 Hints and tips

This section lists some brief tips on tuning and capacity planning. For a more complete description, see *Lotus Domino for S/390 Release 5: Performance Tuning & Capacity Planning,* SC24-5149 and *Lotus Domino for S/390 Release 5: Problem Determination Guide,* SG24-5599.

4.3.1 S/390 platform enhancements

The S/390 server has received hardware and software enhancements during the past few years to enable improved Domino performance and capacity. Following are some ways to utilize those improvements.

4.3.1.1 Central storage

Provide each LPAR with plenty of memory (central storage). Use at least 1 GB, and make 2 GB available if possible. If you have a moderate to small installation, it is possible to use less central storage, but we don't recommend this. Be aware that a Domino HTTP environment needs more storage and is more I/O-intensive than a Notes RPC environment.

4.3.1.2 CPU

If the server is used in an environment with greater than 85% CPU utilization, consider dedicating CPUs to the LPAR supporting the server. This reduces overhead and ensures cycles are available when necessary. This may not be an option in all environments, but it will help provide consistent server behavior at high utilization levels.

4.3.1.3 Dedicate LPAR to Domino

Isolate Domino from other S/390 workloads by dedicating a S/390 Logical Partition (LPAR) to run Domino partitions. The tuning required for a Domino environment is often different from the optimal parameters for other S/390 workloads.

4.3.1.4 Run a small number of DPARs

Run a small number of Domino partitions. If you are migrating from a Release 4 environment, you may have a relatively large number of partitions to support a large number of users. For R5, consolidating to a small number of Domino partitions, with more users in each, will yield substantial savings in CPU and memory.

Note: Use a separate server user ID for each Domino partition.

4.3.1.5 Temporary files

Domino uses its caching capabilities intensively. Therefore, the use of a Temporary File System to improve the handling of temporary files may not have the desired effect. As the number and size of temporary files created by the Domino server may be large, your TFS could become full, perhaps even causing a system crash. Therefore, you should consider a large dedicated HFS for your temporary files.

4.3.2 Server

See Figure 43 on page 85, for a sample Domino directory structure for multiple Domino servers (mail and application), mapped into multiple HFS files.

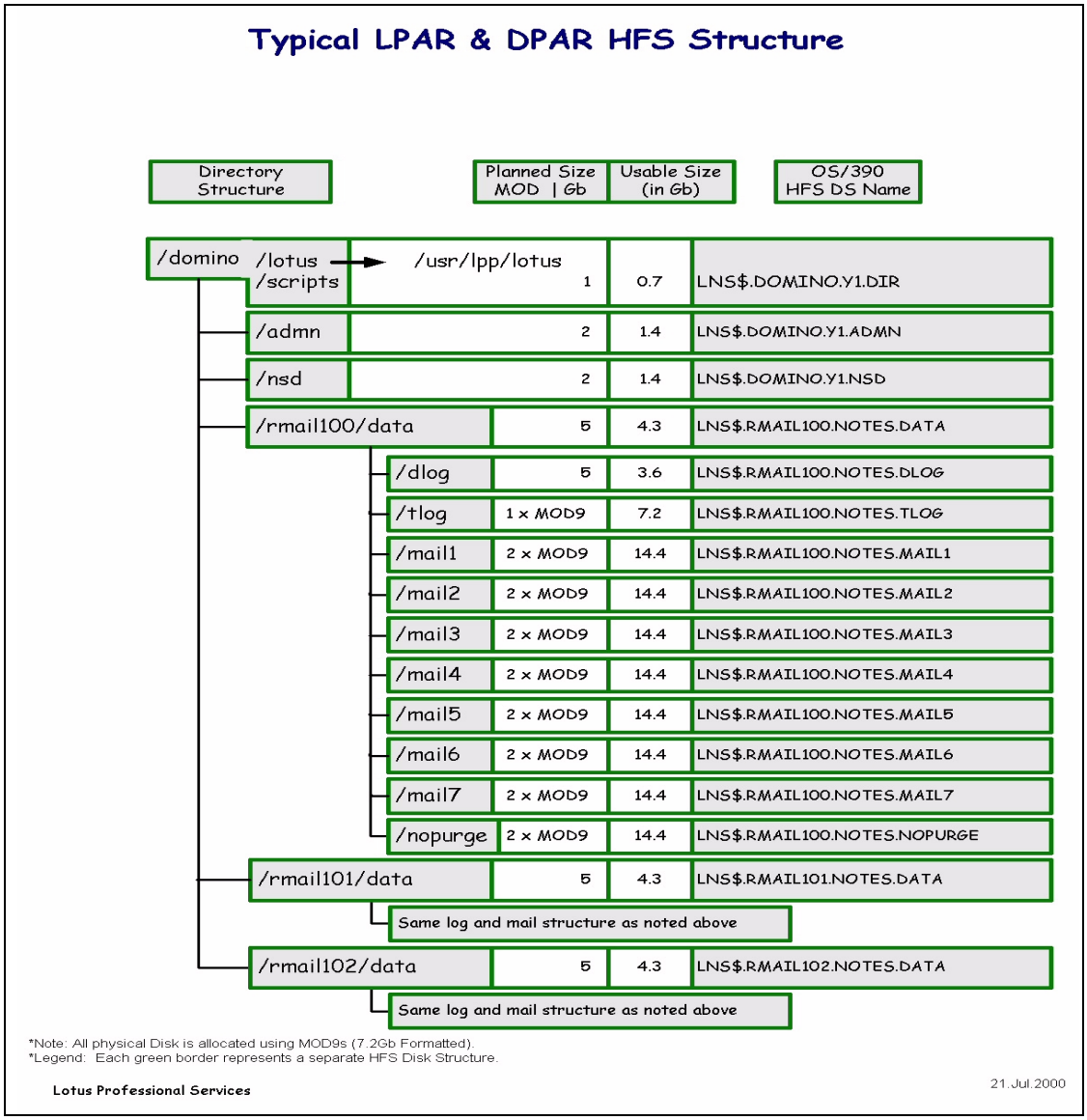


Figure 42. Typical HFS structure for mail servers

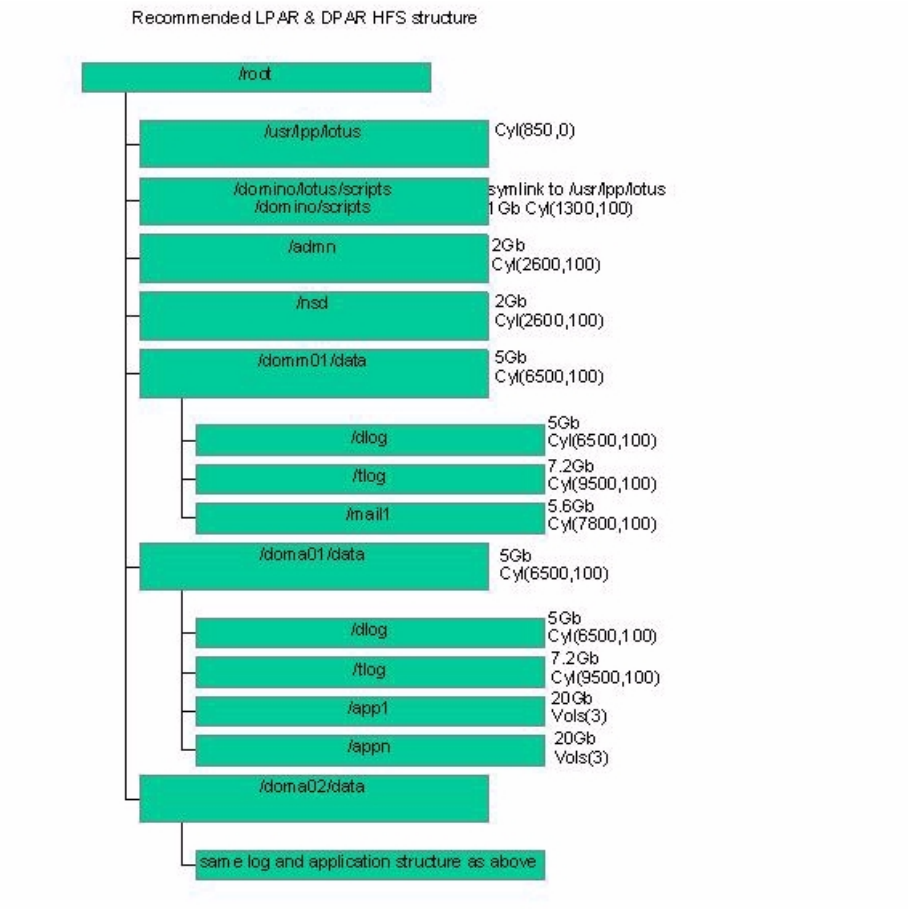


Figure 43. HFS structure for a Domino mail and application server

4.3.3 HFS

The hierarchical file system is relatively new, so here are some ways to use it effectively for performance.

4.3.3.1 Use separate file systems

In a Domino server, not all databases have equal size and activity. Both size and activity differences are reasons to use separate file systems. Particularly, the Domino directory (Name and Address Book), the directory catalog(s), and the log files (notes.log) should reside in separate HFS, using fast DASD. Create UNIX symbolic links in notesdata to point to the real locations of the files in the UNIX file system. Candidates for a separate file system are:

- Domino directory (Name and Address Book)
- Directory catalogs
- Transaction logs
- Log.nsf
- HTTP log file directory
- ViewRebuild directory
- Mail.box

You should monitor your database activity before deciding to use a separate HFS.

4.3.3.2 Spread your databases

Spread your databases over several file systems. If you have a large server with lots of databases, I/O bottlenecks can become an issue, either at the hardware level or within OS/390 itself. SMF records provide useful information for determining whether you have a problem. *Lotus Domino for S/390 Release 5: Installation, Customization and Administration*, SG24-2083 has a very good description of how to structure Notes directories for the best performance. Spreading also allows you to run server tasks, such as Compact, concurrently on multiple file systems.

4.3.3.3 Use full volume allocation

By dedicating a volume to an HFS, you will avoid interference from other applications accessing a shared volume, thereby making it easier to monitor activity on the volume.

4.3.3.4 Limit the number of volumes

Adding more volumes to an HFS will increase its capacity, but will have some side effects, such as:

- Backup will need all volumes dedicated for the duration of the backup. The more volumes there are in the HFS, the longer HFS is unavailable.
- Recovery of a damaged file system will need the same number of empty volumes as are in the multivolume file system.
- Accessing data in a multivolume HFS takes more time.

The backup duration, amount of spare volumes, and your I/O rate must be considered when using a multivolume HFS. If you do not have a need for multivolume HFS, stick to a single volume HFS. If you do need the multivolume HFS, consult software support for the latest guidelines.

4.3.3.5 Limit the number of extents in an HFS file

HFS data sets can expand to multiple extents. As with all MVS data sets, consolidate them to one extent using DFSMSdss. This reduces the time it takes to access files in the HFS.

4.3.3.6 Use the fastest volume available

For the best performance, use the fastest volume available. The IBM Enterprise Storage Server (ESS or Shark) is the first choice. Next is RVA, real 3390 model 3, and then real 3390 model 9.

4.3.3.7 File sharing

Domino does not support sharing an HFS across the sysplex. Make sure that a Domino HFS is mounted on only one LPAR to prevent HFS file corruption.

4.3.3.8 ABEND 0F4

A 0F4 ABEND can indicate data set corruption at the HFS level. There is no need to set a SLIP trap for this abend, but consider not suppressing it by DAE. When you report it to software support, provide the needed information and consider the following:

- The dump reason code is important
- SYSLOG and LOGREC are needed
- Keep the damaged HFS for analysis
- Always check on the latest DFSMS maintenance

On the Domino server, you can observe the following errors when there is a damaged HFS:

- Unknown OS Error
- NIF Detected Storage Corruption error -- this database cannot be read due to an invalid on-disk structure
- Database is corrupt -- cannot allocate space
- Full-text index is corrupt

A corrupted HFS should be recovered by `RESTORE` or by allocating a new HFS data set and using the **copytree** utility to recover as much of the HFS contents as possible.

If you observe an ABEND 0F4, then also check notes.log or log.nsf for other related errors that may have occurred.

4.3.4 Network

Here are some capacity planning tips for your network.

4.3.4.1 Bandwidth

When doing your capacity planning for bandwidth requirement, add the following for mail users:

- 4 Kbps per active Notes user
- 5 Kbps per active Web user
- 6 Kbps per active Notes user in countries that use double-byte characters

(Source: *Strategies for Reducing Notes Network Bandwidth Requirements* by John Lamb).

4.3.4.2 VIPA or CINET

If you are using multiple IP addresses for the partitions on a given LPAR instead of unique port numbers, use Virtual IP Addressing (VIPA) instead of Common INET (CINET). CINET can cause bottlenecks in high stress scenarios.

4.3.4.3 Network connections

Do not use a port mapper server. Give each DPAR its own IP address. When possible, use more than one network connection to a DPAR.

4.3.5 Notes.ini

The Domino server itself can be tuned for performance using parameters in the notes.ini file. Here are some suggestions.

4.3.5.1 SERVER_MAX_CONCURRENT_TRANS

Make sure that SERVER_MAX_CONCURRENT_TRANS is disabled in notes.ini. This can be accomplished by setting its value to -1 (the default). It is used to control the number of transactions that can take place concurrently in the server. On some platforms, this parameter can increase throughput because it reduces the load on the operating system. This is not necessary for OS/390, and disabling the mechanism increases CPU cycles available to the server.

4.3.5.2 Notes_SHARED_DPOOLSIZE

Check that Notes_SHARED_DPOOLSIZE is 8 megabytes. This is a UNIX environment variable that needs to be set before starting the server (see the Domino administration guide for details). The default value for Release 4 systems is 1 MB. For R5 it is 8 MB, so be sure that this variable is set to the default for R5 systems.

4.3.5.3 NSF_BUFFER_POOL_SIZE_MB

Use care when changing the value of the NSF_BUFFER_POOL_SIZE_MB variable in notes.ini. Setting this to a high value allows Domino to cache its data easily, but may cause OS/390 paging overhead if central and expanded storage are in short supply. Setting it too low may eliminate paging by OS/390 at the expense of Domino caching. The correct setting of this variable is completely dependent on the amount of central and expanded storage, along with the number of paging packs available. A good rule is to monitor SMF/RMF data for system paging, and Domino database buffer pool statistics, when changing or setting NSF_BUFFER_POOL_SIZE_MB.

4.3.5.4 SERVER_SESSION_TIMEOUT

The SERVER_SESSION_TIMEOUT variable has a default value of 4 hours. Setting this variable to a much lower value, such as 30 minutes (as shown in Figure 44 on page 90), allows OS/390 to re-use the threads that are idle and waiting for a user to reconnect. It also gives the Notes statistics variable *Server.Users.active* a better value, since the idle sessions are disconnected and, therefore, not included. The statistics can be obtained using the `SHow STatistics server.users.*` command as follows:

```
> sh stat server.users.*
  Server.Users.1MinPeak = 2
  Server.Users.1MinPeakTime = 09/25/2000 17:07:47 EDT
  Server.Users.5MinPeak = 4
  Server.Users.5MinPeakTime = 09/26/2000 11:38:03 EDT
  Server.Users.active = 0
  Server.Users.active15min = 2
  Server.Users.active1min = 2
  Server.Users.active30min = 2
  Server.Users.active3min = 2
  Server.Users.active5min = 2
  Server.Users.Peak = 6
  Server.Users.Peak.Time = 09/26/2000 10:01:33 EDT
>
```

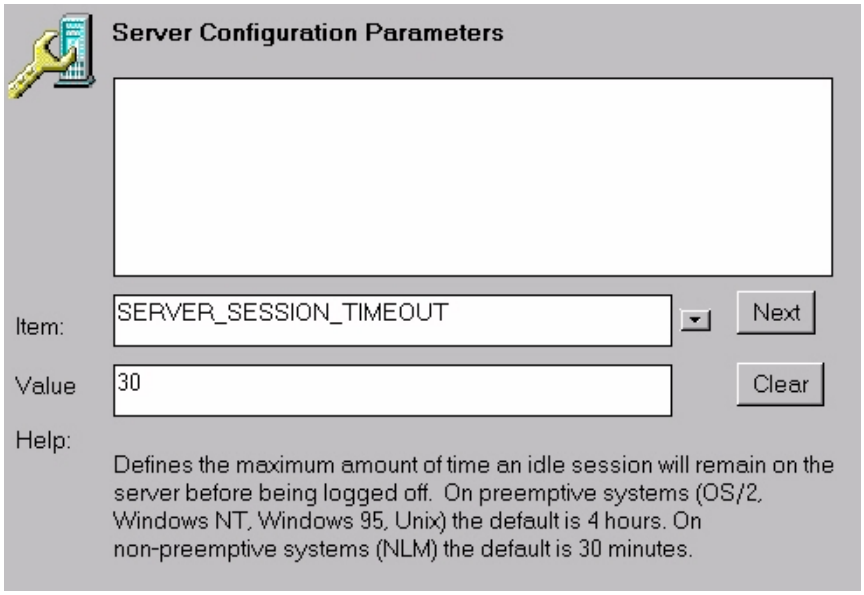


Figure 44. Notes.ini: server session timeout

4.3.5.5 SERVER_POOL_TASKS

The Domino server threadpool is not currently used by the Domino HTTP add-in server. If you are going to dedicate your Domino server to serving Web browsers, consider dropping the SERVER_POOL_TASKS setting in notes.ini back to 20. This will reduce some system overhead:

```
SERVER_POOL_TASK=20
```

4.3.5.6 VIEW_REBUILD_DIR

You should use separate file systems for the temporary space that Domino uses for rebuilding views. Be aware that the space you need to rebuild view indexes may be up to five times the size of your largest Notes database.

To set up a separate file system for view index rebuild, add the following NOTES.INI setting to point the temporary view index rebuild to the mount point of a separate HFS:

```
VIEW_REBUILD_DIR=/mountpoint
```

4.3.5.7 UPDATE_NO_FULLTEXT

On mail servers, you should not allow users to create a full-text index of their mail databases. Consider setting the *UPDATE_NO_FULLTEXT* in notes.ini to 1 to disable full-text indexing on a server:

```
UPDATE_NO_FULLTEXT=1
```

Figure 45 shows how to set this using the server configuration parameters.

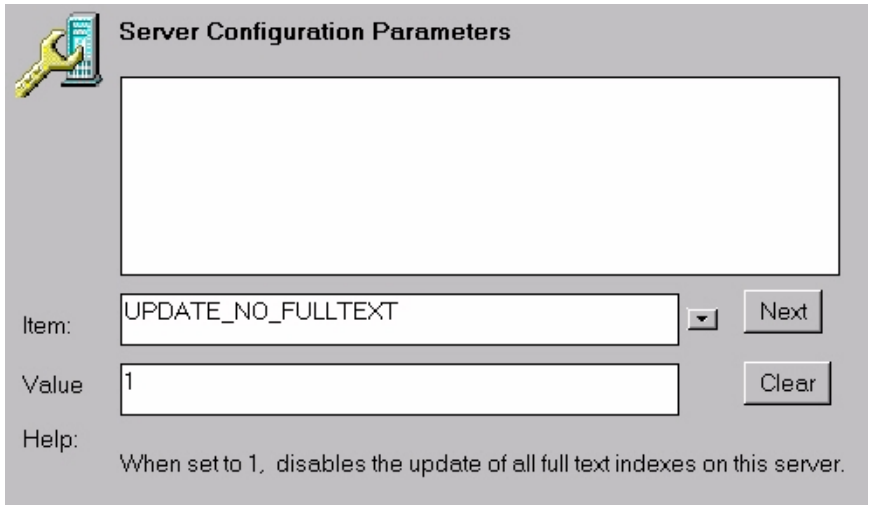


Figure 45. Notes.ini: disable full-text indexing

4.3.6 Capacity planning

What is the right number of Domino partitions to put in an LPAR?

When planning for a new server installation, or when migrating from a Domino R4 to an R5 environment, one of the first decisions to make is how many Domino partitions to use, and how to spread them out across LPARs for optimum performance. The answer is simple from a performance and scale standpoint—as few as possible.

In a traditional Notes R5 client mail environment, this will be two or three per LPAR, while partitions serving data to Web browsers should use one Domino partition per LPAR.

The limiting scale factor is the amount of memory (central storage) available to the LPAR. What ultimately determines the ratio of Domino partitions to LPARs is the portion of that available central storage that can be efficiently utilized by a Domino partition.

For example, measurements in the lab have shown that as the number of users on a given partition is scaled to larger numbers, contention for shared resources within the server causes the CPU usage per user to rise before all available central storage is used up. As a result, we recommend two Domino partitions so that this contention can be effectively avoided, and both CPU and central storage usage are as efficient as possible. In a production environment, you may find that your users place more stress on the server, making a third partition a viable option.

As for Web-based users, who access Domino resources using HTTP, we know that the memory requirements per users are substantially higher, making a single partition within the LPAR the best alternative. If you are scaling to high user workloads in either the Notes client or Web browser environments, it is important to understand that the amount of central storage is key; more is better.

When talking about the optimum number of partitions per LPAR, we get questions about this most often from users migrating from Domino R4 to R5. With R4, we recommended that the number of production users per Domino partition be limited to 500 to 700, because the design of the R4 server did not scale very well. This might mean six or more partitions on an LPAR, yielding a total of perhaps 3,000 users. With R5, we expect those 3,000 users to fit comfortably within one Domino partition, with the possibility of two such partitions running on the LPAR, depending on the type of hardware used, central storage available, and type of users supported.

While the number of partitions recommended per LPAR has dropped, the number of users supported on that LPAR has grown by a factor of two or more with Domino R5, thereby increasing efficiency and reducing overall administration overhead.

You may still ultimately choose to deploy more partitions in a single LPAR than we recommend here for several reasons (backup, isolation). However, with Domino R5, it is a choice rather than a necessity.

Chapter 5. Backup, recovery and transaction logging

When running Lotus Domino, like any application on OS/390, it is necessary to have a well-defined backup strategy in place to ensure data recovery. This chapter lists some OS/390 options for Domino. We do not document detailed implementation instructions for these options, but we present a high-level point of view, listing advantages and disadvantages as appropriate.

5.1 Understanding your backup and recovery needs

The characteristics of data in the Domino S/390 environment can be divided into four categories that influence backup requirements.

- **Discardable data**: Loss of this kind of data may be acceptable to you. The recovery process for an individual corrupted database might simply be to delete and recreate the database. To avoid re-creating a large number of databases, it might be simpler to take frequent volume dumps using a tool such as DFSMSdss. A simple DFSMSdss volume restore could quickly reload all the mailboxes with a previous version of mail data.
- **Static data**: The Lotus binary directory (/usr/lpp/lotus) is an example of data which is very static. It might seem like one backup after the installation would suffice for later recovery. However, the content of the directory is not totally static; for example, some updates take place when a new server partition is added. These updates are not well documented, so it is advisable to base your recovery strategy on full periodic backups. Since this data is contained in a single HFS, a DFSMSdss data set dump may be the option of choice.
- **Non-critical data**: The key here is to understand what is meant by non-critical data. Data is non-critical when business needs make it acceptable to recover the data to a given point in time in the not-too-distant past. This requirement could be accomplished through periodic backups and then a restore of the last backup when needed. Any updates that may have taken place after the backup would be lost, but this is acceptable by definition. Again, in certain instances, mail data might fall into this category.
- **Critical data**: This category involves databases where the business need dictates that the data must be recovered to its state immediately preceding the failure.The only way to accomplish this type of recovery is with periodic backups combined with some journal-based forward recovery system. The new transaction log in Lotus Domino R5 allows solutions in this area; see 5.2.6.4, “Tivoli Data Protection for Lotus Domino” on page 103.

A second consideration is the scope of the recovery. In the world of Lotus Domino data, there are three distinct recovery situations.

- **Full recovery**: Common disaster recovery situations, like failure of a disk subsystem. All databases contained in the subsystem must be restored to an accessible state.
- **Database level recovery**: A single database may have been corrupted. Recovery involves restoring the last backup of the database.
- **Document level recovery**: One or more documents in a database have been lost, either deleted or corrupted. Normal recovery in this situation involves restoring the backup of the database to a temporary location and then from it, copying the required documents to the active version of the database.

A third consideration is the ability to take backups without interrupting the service provided by the Lotus Domino server. In the case of 24x7 availability requirements, all backup operations must run concurrently with the server. The OS/390 environment has software products that allow this to happen.

5.2 Different backup techniques

The data on the Domino server can be backed up in several ways:

- Notes-based database replication
- UNIX archive commands
- Traditional MVS backup utilities
- RVA-based backup options
- Tivoli Storage Manager (TSM) and Tivoli Data Protection (TDP)
- USS Backup-Archive client

5.2.1 Notes-based database replication

Notes provides standard facilities to replicate databases. The replicas can exist on the same server with different file names or on a different server. Replication provides an easy way to make backup copies of databases. However, it does not meet all backup requirements:

- It is possible that an error could be propagated to the replicas before being detected. In that case, the replicas also have the error, and there would be no valid copy from which to recover.
- The replicas take up additional disk space.
- If the replicas are on the same server, they need to be copied to tape and taken off-site.

- If the replicas are on another server, the data must be transferred across a network connection, which is slow and may require additional network capacity.

Therefore, while replication can be useful, it should not be relied upon as the only backup method.

5.2.2 UNIX System Services archive command

OS/390 UNIX System Services provides a number of utility commands to perform data backup type functions. The most common data backup option is *tape archiver* (tar).

Tar manipulates archives. An archive is a single file that contains the complete contents of a set of other files. An archive preserves the directory hierarchy that contains the original files as well as the file attributes like owner, group, permissions, and sticky bit. The name *tar* was derived from tape archiver but it can use archives with any medium. This utility is similar in function to the common PKZIP tool of the PC world, including the compression capability. The same tar utility can be used to build the archive from a file structure or to restore the file structure from the archive. It is just a matter of specifying the proper command options.

Since tar acts on UNIX files, it requires the “container” HFS data set to be mounted, but the backup is not an HFS backup, only a backup of those files and/or directories selected in the tar command. Tar can be used to back up both.

In the Domino for S/390 environment, tar can be used to back up both the binary modules and the Notes databases. For data integrity reasons, the Domino server whose data is being backed up cannot be running while the archive is being built. Otherwise, unpredictable results may occur when running the server after restoring the archive.

From a recovery perspective, it is obvious that tar will recover the data to the point-in-time status of the backup. Any updates to the data after that time will be lost.

5.2.3 Traditional OS/390 backup utilities provided by DFSMS

DFSMS provides two basic tools with backup functionality, DFSMS Data Set Services (DFSMSdss) and DFSMS Hierarchical Storage Manager (DFSMShsm). In addition, there are hardware devices that support an enhanced version of DFSMSdss operation called Concurrent Copy.

5.2.3.1 DFSMS Data Set Services (DFSMSdss)

DFSMS Data Set Services is the traditional backup/restore utility of the DFSMS environment and is a common element in the disaster recovery strategy for OS/390 installations. It can work either at the volume or data set level. It can handle HFS data sets as separate entities, but does not have the granularity to access UNIX files within the HFS.

The DFSMSdss DUMP command can back up HFS data sets in mounted or unmounted state (COPY is not supported for HFS). DFSMS will quiesce activity against the specified HFS data set and then invoke backup processing. When the backup is complete, the file system is unquiesced and user activity is resumed.

The file system quiesce process for mounted HFS data sets opens the door for backups to be taken while the Domino server owner of the data is active. In that scenario, the file system could be quiesced just between writes in the middle of a database update by the Domino server. The resulting backup would produce a database with incomplete data. What would happen if the need to restore arose and the Domino server was started with the restored database? The startup database consistency check scan would detect the partial update and a subsequent FIXUP would remove it. This is acceptable if the partial update was the only update performed by the application's logical unit of work. In addition, it is obvious that all other successful updates from the time of the backup would also be lost.

There is another scenario, where the quiescing and backup may take place at a clean point, meaning that there are no documents being updated; but if the application's logical unit of work consists of updates to two or more documents, the server may be caught between updates during the backup. Domino has no externalized concept of *logical unit of work,* nor the corresponding controls for it, such as syncpoint and bucket. Under these conditions, the backup would contain logically incomplete data, although not inconsistent from a database structure point of view. This is a serious issue that must be considered.

In addition to the previous considerations, the time involved in dumping large HFS files can translate into unacceptably long system quiesced times.

5.2.3.2 DFSMSdss concurrent copy

A possible solution to shorten the quiesce time during backups is the use of DFSMSdss Concurrent Copy (CC).

Concurrent Copy (CC) is a D/T3990 extended function that enables data center operations people to take a copy or dump of data while applications

are updating that data. Concurrent copy delivers a copy of the data in a consistent form, as it existed when the process was initiated, before any updates that may occur while the rest of the process takes place.

The Concurrent Copy is implemented through a mix of 3990 Licensed Internal Code and DFSMS code. For DFSMSdss, Concurrent Copy is requested by adding the CC keyword to the `DUMP` or `COPY` command input. When a CC operation is started, there is a brief initialization phase, during which there is serialized access to the data being dumped. In a matter of seconds, once the CC environment is initialized, the copy operation is logically complete and the data becomes available for application updates. The actual physical copy starts after initialization, moving data from source to output device at normal speeds. The copy is physically completed later when all the data has actually been moved to the output device.

Without getting overly detailed, the mechanism used by Concurrent Copy to provide a frozen data image as of initialization time is as follows. First, at initialization, a table is built identifying the device tracks involved in the backup operation. After that, two I/O filters are set in place for the duration of the physical copy. All writes to the affected tracks are intercepted, inserting a track read to capture the “before image”, saving it in a sidefile kept in storage in a data space. Once this is done, the requested write is performed. The second filter handles all the reads issued by the backup process, determining whether the data should be taken from the sidefile track images or from the real device.

From a Domino perspective, there are two ways to look at the use of Concurrent Copy:

- DFSMSdss dump/cc while the server is up: The file system is quiesced for a brief period of time and then released for normal server operation. Although there is a clear advantage over non-cc dumps, the issues related to possible database inconsistency and application data integrity remain a serious concern.
- DFSMSdss dump/cc with the Domino server down: The advantage is that the server can be kept down for little more than the initialization time, and then restarted while the physical dump takes place. For this to work, it is important to force a single common initialization phase for all the data related to the server by INCLUDEing all of the server's HFS data sets in a single DFSMSdss DUMP command. This option may be very appealing for installations where only full backups of Domino databases are considered acceptable. An example could be the mail databases, assuming that in a recovery situation the loss of mail since the last backup is acceptable.

5.2.4 DFSMShsm Hierarchical Storage Manager (DFSMShsm)

DFSMShsm is the tool for automated data management in the OS/390 SMS environment. It provides granularity at the data set level (that is, HFS data set), not at the UNIX file level within an HFS data set. Conceptually, you can think of HSM as a predecessor of Tivoli Storage Manager (TSM).

Ideally, when a data set is created in the SMS environment, a data management class is assigned to it. The data management class determines the kind of handling the data set will be subject to for the duration of its life. This includes backup frequency, backup retention, and number of versions to be kept. DFSMShsm is the agent responsible for executing the actions dictated by the data management class, such as taking new backups and deleting old backups when obsolete.

Since DFSMShsm uses DFSMSdss as the data mover, its use has issues similar to the use of DFSMSdss. In addition, there is the consideration of the timing of the backup in relation to the status of the Domino server. This is particularly critical if the backup is to be taken with the server down.

5.2.5 Ramac Virtual Array backup options

The RAMAC Virtual Array (RVA) employs an advanced storage system architecture. Logical volumes, called *functional volumes*, are indirectly mapped to physical storage within the system.

5.2.5.1 Configuration of RVA

RVA configuration is done by defining up to 256 volumes in terms of traditional 3380/3390 DASD. An update to the Licensed Internal Code will bring this number to 1024. Data is dynamically mapped across a collection of physical devices organized into a logical group called an array. Mapping between the RAMAC Virtual Array functional 3380/3390 volumes and the arrays is accomplished with tables of pointers located in the control unit. The RVA is built upon the concept of *virtual DASD* to the host, where the volumes look like 3380/3390 devices attached to 3990 control units.

5.2.5.2 SnapShot function

IBM RAMAC SnapShot is an advanced function made possible by the virtual architecture of the RAMAC Virtual Array. SnapShot, which consists of both host software and an added RAMAC Virtual Array feature, makes it possible to quickly create duplicate virtual copies of either volumes or data sets that may be accessed like traditional copies. Yet the copy function requires no CPU time to complete and requires no storage space for the virtual copies

until changes are made to the data. Most significantly, the elapsed time normally associated with creating copies of data is eliminated.

With SnapShot, actual data is not moved but rather pointers to the data are duplicated (snapped), taking seconds rather than minutes or hours. Furthermore, physical disk space is not required to accommodate the snap until data at either the source or the target is updated. From a device perspective, SnapShot is an atomic process. That is, it takes place as the result of a single I/O operation during which the data involved cannot be updated, providing the base for consistent point-in-time backup.

5.2.5.3 Storing HFS and Domino data

The use of the RAMAC Virtual Array as storage for the Hierarchical File System has characteristic effects that must be understood. Although these effects are not unique to HFS and may be common to other data set organizations (such as a PDS), the reason they are considered here is because of the dynamics of Domino data and its sheer volume. A Domino installation with a significant number of mail users may require terabytes of HFS storage. Due to the nature of mail processing and the Domino housekeeping (database compaction), this data can be subject to a high update rate that results in a magnified effect. Proper understanding of the RVA/HFS behavior will help to take advantage of the storage subsystem features.

When an HFS data set is allocated, the net effect in the allocation of physical space in the RVA is nearly nothing. DFP may create a minimum data structure, but the fact that all the tracks are empty will result in no physical space being used by the data set. Once the HFS is mounted and files are allocated, more structures will be created within the HFS data set; but it is not until data is actually written into these files that the physical utilization or Net Capacity Load (NCL) of the RVA will be affected. There are three main considerations:

- Experience indicates that the Domino data tends to yield lower RVA compression ratios than average, causing the RVA to hold comparatively less data. In practice, actual compression ratios depend on overall usage patterns. For example, widespread use of binary attachments will create denser data with lower compression ratios. Some installations have seen ratios of 2:1 or lower. This presents no problem other than the requirement to plan for the adequate capacity to handle the volume of data. Understanding the characteristics of the user's data and planning accordingly will help to avoid surprises.

- HFS data sets may cause a higher RVA load than they appear to according to the output of UNIX commands. For example, database housekeeping (compaction) may increase the Net Capacity Load. This could happen because, with pre-R5 data, when Domino compacts a database, it copies the file into a new one and then deletes the original database, renaming the new one—not unlike the process to reorganize VSAM files. If the HFS structure is fairly new, the copy of the file will take place over never-used tracks, increasing the load in the RVA. On the other hand, deleting the original file returns unused space to the HFS structure, but not to the RVA. For the RVA, the old data is still there. Over time, the whole HFS will tend to appear to the RVA as fully loaded, although to UNIX Services, it may show available space.

  In principle, there is no need to reorganize an HFS, but with the RVA technology, it would be advantageous to force the release of unused space at the physical level.

  - The immediate solution is to reorganize the HFS by allocating a new one, copying all the UNIX files from the old one to the new one, and then deleting or renaming the HFS data sets. Obviously, there are mount and unmount considerations. This could be a good option after a weekly compaction process, and it does require stopping the server(s). Note that the use of DFSMSdss to restore or create a new copy of an HFS data set does not reorganize the internal data structures. Thus, it will not have the desired effect on the RVA, and the problem will be perpetuated.
  - An alternative would be to write a utility to create a file with low values, spanning most of the empty space in the HFS, and then subsequently deleting the file. While there would still be data written to the RVA, it would replace denser data and greatly help to decrease the NCL because of the high compression ratio of the low values.

- High update activity, a characteristic of the Domino database compaction process, can create a shortage of free collected space, causing temporary performance degradation or even an outage. During compaction, a database is rewritten in another location and the old one is deleted. The effect is that free collected space is consumed and free uncollected space is generated. If this process is repeated over a large number of databases, as would occur on a weekly compaction schedule, and the RVA is operating at a high NCL, the free collected space can be exhausted because it is used at a faster rate than the collection process consolidates uncollected space. If this situation is detected, the background collection process is given the highest priority in the RVA and production performance suffers. If the collection process cannot catch up, free

collected space is exhausted, resulting in the RVA not being able to process writes at all, not even a volume table of contents (VTOC) update to delete a data set to release space. To prevent this undesirable situation from happening, it is important to understand the specific RVA workload and determine what can be tolerated as the operational NCL. Also, spreading out compaction by servers over different nights will alleviate the problem.

5.2.6 Tivoli Storage Manager (TSM)

The Tivoli Storage Management product set allows you to align storage management policies with business practices, so that you can make optimal use of the IT organization for a powerful competitive advantage. Following is an overview of the Tivoli Storage Manager, S/390 Edition function provided for the OS/390 server platform:

- Administrative clients that allow the administrator to control server activities, define storage management policies for workstation files, and set up schedules to provide automated backup and archive services at regular intervals.
- Support for backup archive clients that allow users to restore or retrieve files from a Tivoli Storage Manager, S/390 Edition server.
- Support for backing up and archiving files on a variety of devices ranging from high-speed disk systems to removable cartridge media to tape systems.
- Java-enabled Web browser interfaces to support remote administration and remote backup-archive operations from any platform with a Java 1.1.6 capable browser, anywhere in the enterprise.
- Support for customized reporting and analysis using an industry-standard SQL interface.
- Enterprise management that helps improve administrator productivity by enabling an administrator to proactively manage, operate on, and view many Tivoli Storage Manager, S/390 Edition servers from a single point.

5.2.6.1 Support for multiple backup-archive clients

Tivoli Storage Manager, S/390 Edition Version 3.7 backs up data from many different operating systems, including OS/390 UNIX System Services.

5.2.6.2 Tivoli Disaster Recovery Manager

Tivoli Disaster Recovery Manager is a separate product which assists with recovery of a TSM server in case of a disaster. It requires Tivoli Storage

Manager as a prerequisite, and is available for OS/390 TSM servers. It assists with:

- Automated generation of a customized TSM server disaster recovery plan
- Off-site recovery media management
- Inventory of machine information required to recover the TSM server and its clients
- Centralized management of the disaster recovery process
- Executable scripts that assist in recovery automation
- Electronic vaulting of storage pool and database backups to another Tivoli Storage Manager server

5.2.6.3 UNIX System Services backup-archive client

The IBM ADSTAR Distributed Storage Manager (ADSM), now known as Tivoli Storage Manager (TSM), is a tool that allows the backup of data sets from many client platforms to many server platforms. It provides functions similar to DFSMShsm on OS/390 but across a network. The TSM server and UNIX backup-archive client are both available on OS/390. The TSM server executes as a native MVS application; the UNIX backup-archive client executes under OS/390 UNIX services. TCP/IP is required as the communication protocol between the TSM server and UNIX client even when the client and server execute on the same machine.

Note: Even when TDP (see 5.2.6.4) is implemented, the USS backup-archive client is still needed for backing up the Domino executables and links (database and directory), since TDP backs up only Domino databases and templates.

There are four options for backing up files:

- **Static** - If being modified, TSM will not back up the file or directory.
- **Shared Static** - If being modified, TSM will not back up the file or directory, but will retry the backup a predetermined number of times.
- **Shared Dynamic** - If being modified, TSM will not initially back up the file or directory. It will retry the operation a predetermined number of times; it will back up the file on its last attempt, even if the file is being modified.
- **Dynamic** - TSM will always back up the file or directory, regardless of whether or not it is being modified.

If your OS/390 was installed using ServerPac, the TSM OS/390 UNIX client code is in the /usr/adsm directory. For more information on command usage, see *ADSM V3R1 Using the UNIX Backup-Archive Clients*, SH26-4075.

5.2.6.4 Tivoli Data Protection for Lotus Domino

Tivoli Data Protection for Lotus Domino for OS/390 UNIX System Services is a new application client that enables online backups, restores, and activation of Lotus Domino R5 server database and transaction logs. Lotus Domino R5 has introduced some significant changes in the Notes server architecture. These include transaction logging for interactions with Notes databases as well as a new application program interface (API) for backup and recovery of Notes databases. Tivoli Data Protection for Lotus Domino is designed to use these new backup and recovery APIs in Lotus Domino R5 to provide online backup and restore capabilities using a Tivoli Storage Manager server.

Note: Even when TDP is implemented, the USS backup-archive client (discussed in 5.2.6.3) is still needed for backing up the Domino executables and links (database and directory), since TDP backs up only Domino databases and templates.

Backups are stored on Tivoli Storage Manager or Tivoli ADSM storage. Tivoli Data Protection for Lotus Domino helps protect and manage Lotus Domino Server data by making it easy to perform the following actions:

- Backup online Lotus Domino databases
- Maintain multiple versions of Domino databases
- Archive Lotus Domino transaction log files, when archival logging is in effect
- Restore backup versions of a Lotus Domino database and apply changes made since the backup from the transaction log
- Restore Domino databases to a specific point in time
- Expire database backups automatically based on version limit and retention period
- Expire archived transaction log files when no longer needed
- Automate scheduled backups

Tivoli Data Protection for Lotus Domino provides two types of database backup (incremental and selective) and a log archive function.

- Incremental backup provides a conditional backup function that creates a full online backup of Domino databases, when necessary. The specific

conditions that determine when a new backup is necessary vary depending on whether the database is logged or not.

- Selective backup unconditionally backs up the specified databases, unless they are excluded from backup through exclude statements.
- When archival logging is in effect, changes to logged databases can be captured in between full backups, by archiving the transaction log.

Even if you choose not to enable transaction logging, TDP can still provide benefits. For instance, it provides the only way to guarantee a consistent backup on an online Domino database, because it can access the pre-image data buffers through the Domino backup and recovery API.

Tivoli Data Protection for Lotus Domino communicates with a Tivoli Storage Manager Server using the Tivoli Storage Manager API. Tivoli Data Protection for Lotus Domino communicates with a Domino server using the Lotus Domino API.

For more detailed information on the Tivoli products, see the following Web site:

`http://www.tivoli.com`

There may also be other vendor products with similar functionality that can be used for backup purposes.

5.3 Using transaction logging

Domino R5 supports transaction logging and recovery. When you enable this feature, the system captures database changes and writes them to the transaction log. Then, if a system or media failure occurs, you can use the transaction log and a backup utility, such as Tivoli Data Protection or any other third-party backup utility, to recover your database.

5.3.1 Implementing transaction logging

These are the basic steps for implementing transaction logging:

1. Allocate a space for the transaction logging directory. For optimal performance and data integrity, transaction log files should reside on a separate, high-volume HFS.
2. Place all databases to be logged in the Domino data directory, or one of its sub-directories. (Domino can also log databases referenced by way of database/directory links).

3. Upgrade databases to R5 format, by running `compact` on them. When you enable transaction logging on a Domino R5 server, all databases in R5 format are automatically logged.
4. Configure transaction logging on the server. You can disable it for individual databases if you need to, though this is rarely advantageous.

   Configuring a Domino server for transaction logging is a simple matter of choosing settings in the Server Configuration document.

   Once you have set up a dedicated HFS for the log files and moved or upgraded the databases you want to log, all that is left to do is configure transaction logging on your Domino servers. Table 3 on page 105, describes the settings to be configured, using the Domino Administrator.

Table 3. Transaction logging settings

Setting	Description
Transactional Logging	The default is Disabled; choose Enabled.
Log path	Specify the location of the dedicated log disk, or a pathname. You should point to your transaction logging file system path. The default, which is simply a placeholder, is LOGDIR in the Domino data directory.
Maximum log space	The maximum size, in megabytes, for the transaction log. Domino formats from three to 64 log files, depending on the space allocated. Used only for Circular logging.
Use all available space on log device	Choose Yes once you have set up a dedicated disk. Domino uses the whole volume instead of the maximum log space.
Automatic Fixup of corrupt databases	Enabled by default. This means that, if a database is found to be corrupt after restart, in which case the transaction log cannot be used to recover it, Domino runs the Fixup task automatically. If disabled, Domino notifies the administrator to retrieve a backup and recover the database from the logs, or to run Fixup manually.

Setting	Description
Run time/Restart performance	Controls how often Domino records a recovery checkpoint in the transaction log. Checkpoints entail system overhead, but they are important for restart performance. You can choose the default (recommended), or skew the frequency of checkpoints to favor server performance (fewer checkpoints) or to shorten recovery time (more checkpoints). The fewer the checkpoints, the longer restart takes if a system failure occurs.
Logging style	Choose Circular (the default) only if you do not have an R5 compatible backup utility in place. Otherwise choose Archive. **Note:** when starting out, many people choose circular logging to get an idea of the space required.

Although logging is turned on at a Domino server level, it can be disabled for an individual database. We recommend you not do this, because then `fixup` will need to be run for this database if there is a failure.

Note: Transaction logging does not work with databases that use formats from earlier releases. Domino has a facility to convert them to the new format if you desire.

Edit the server document as shown in Figure 46, and set the transaction log parameters on the Transaction Logging tab.

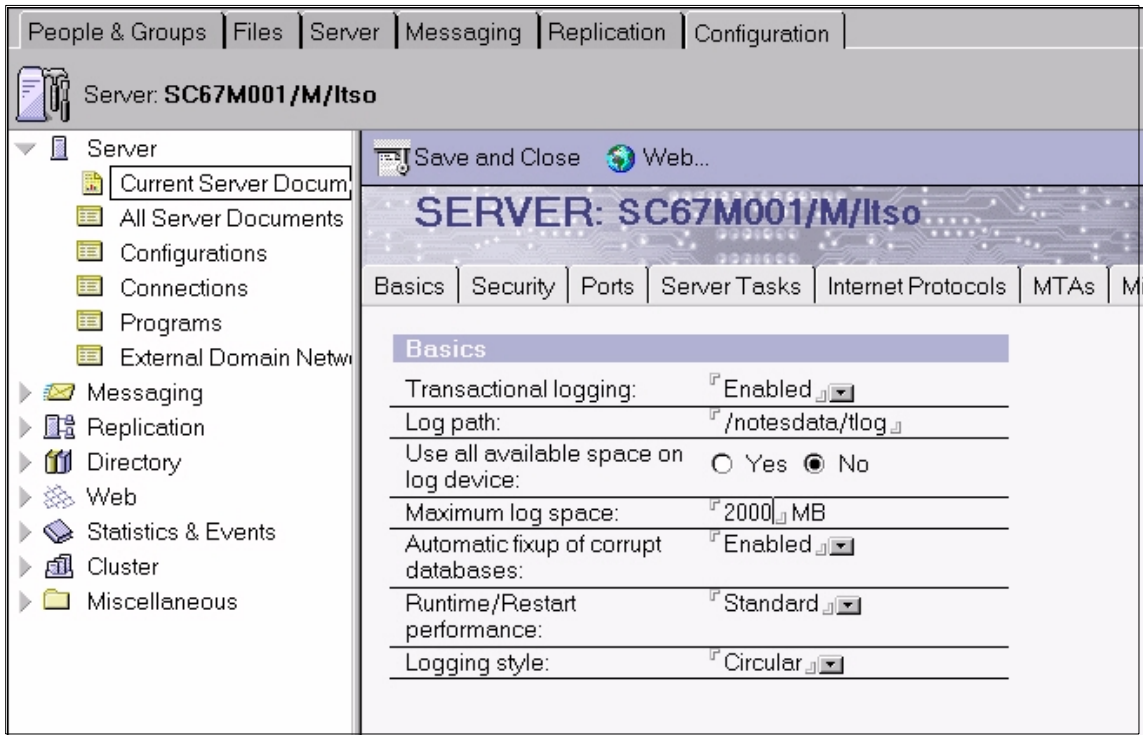


Figure 46. Enabling transaction logging

5.3.2 The transaction logging HFS

In order to increase the speed of write operations, and ensure data integrity, you should store transaction log files on a separate, dedicated HFS with the following characteristics:

- The size of the separate disk used for transaction logs must be at least 1 GB; we recommend at least 2 GB for enterprise scale implementations. You need to allocate adequate space for at least a day's worth of log storage under the most demanding conditions. You need at least two days' worth of space for transaction logs. This will help you cope with problems related to backups or excessive server utilization. You can determine the maximum size for the Transaction Log Extent, 4Gb being the highest value. However, the size of the volume may limit this maximum.
- To guarantee the integrity of your mission-critical Domino data, we recommend that you use mirrored disks, in a separated disk controller from the other Domino database disks. Mirrored disks ensures that you never lose your transaction logs, even if one log disk fails. With DFSMS

1.5 you can use multiple volumes. We recommend that you use one single volume for the Transaction Log Extents file system.

- It is important to point out that when you implement transaction logging, your databases are no longer your most critical data--your transaction logs are. With transaction logging, you could lose all your active, logged databases, yet lose no data, because your logs can replay every transaction against your database backups.
- The dedicated log disk should have a separate controller. With transaction logging, Domino writes each transaction first to a sequential log file, then to the Domino database(s). Since all transactions are written to the log, I/O rates will be high on the HFS that contains the transaction logs. For this reason, the log files should be on an isolated file system. If you put other databases into this same file system, you will have I/O contention. Isolation of logs is not only a cost-effective way to boost write performance significantly, but also improves reliability.

Note: If you are planning an upgrade to Domino R5 and also to your disk subsystem, you should first make the hardware changes to your disks and controllers, and then upgrade to R5. That way you can isolate issues with the new hardware using a familiar Domino infrastructure.

To change the location in which the transaction logs are stored, edit the server document, as shown in Figure 46 on page 107. Change the value for the Log path field on the Transactional Logging tab.

5.3.3 Benefits of transaction logging

The great advantage of transaction logging is that it allows Domino to automatically recover data loss through system failures by replaying the deferred database changes from the transaction log. The principal benefits are:

- Much faster server restart times, which means enhanced system availability. With transaction logging, you will rarely need to run `Fixup` on a logged database, except when Domino detects errors following recovery.
- Improved data integrity, because the recovery process detects any missing changes and replays them, while “undoing” any changes that were incomplete at the time of the failure. Thus, Domino can start with “clean” databases.
- Simplified backup procedures. On a daily basis, most sites will need to back up only the transaction logs for that day.

- An increase in overall system performance. Fewer file operations mean the OS kernel spends less time in “kernel mode” or “privileged mode” and more time in “user mode,” doing the work requested by applications. Fewer overall CPU cycles are needed to complete the same work, potentially allowing a server to support more users.

  By writing to the log sequentially, rather than to the databases randomly, write operations to databases take less time. Less physical disk arm movement at critical times means faster writes. This is possible because, with transaction logging enabled, Domino R5 posts all database transactions to the log file, without waiting for the actual write/commit process. The system, instead, considers the successful posting of an operation to the log to be a commitment to persistent storage. Transactions either succeed (commit) or fail (rollback). So, database operations can complete before the physical write process to the databases occurs.

- Transaction logging speeds recovery not only from system failures, but also from media failures. If restarting from a system failure, Domino R5 automatically restores affected databases. Domino automatically replays the transaction logs, performing only those transactions not written to the databases at the time of the failure, without needing to run Fixup as part of a system restart. The server looks at the transaction log and identifies the databases that had changes, ignoring the others. Then, the server checks consistency for only the changed databases, a much smaller number. Even in those databases, only some of the documents, identified in the transaction log, will have to be scanned to see if they are correct. This way, the startup process becomes lighter.

  When disk drives fail, databases can be corrupted or lost. To recover with R5, you use a third-party backup utility to restore database backups and archived transaction logs, and then apply current transaction logs to recreate the affected databases.

5.3.4 Domino backup and recovery when using transaction logging

Though transaction logs protect you from system failures, regular backups are still essential for recovery from media failures, where one or more disks fail or are corrupted. Domino R5 provides a new backup API, enabling TDP or any other third-party vendors to provide you with tools that simplify all regular backup procedures. All R5 compatible backup products work in combination with transaction logging.

If you have a Domino R5 compatible backup solution and have upgraded to Domino R5 and enable transaction logging, you don’t need daily full backups,

but incremental ones instead. Once that utility is operational, the critical backup jobs you will need to perform regularly with transaction logging are:

- Scheduled (typically daily) archiving of the transaction logs, in the form of incremental backups.
- Periodic (typically weekly) full database backups, so that you are prepared for media recovery. Unlogged databases should be backed up more frequently; a full backup daily is recommended for unlogged databases.

In this scenario, recovery following a system fault simply consists of allowing Domino to automatically recover all affected databases using the transaction logs. Restoration of a corrupted database due to bad media basically consists of restoring the most recent full backup from tape and using the transaction logs to roll forward the data to the time of the failure.

The Domino R5 backup API has four parts:

- **Online full backup:** Enables backup utility to flush the database image to disk, and to capture any changes made during the backup.
- **Online incremental backup**: Only available when archive transaction logging is enabled. Backs up all of the full log segments and reports to Domino when complete. Domino chooses to reuse these or delete them at server shutdown.
- **Online restore:** Takes the database offline, brings both the dirty database image and all before-images back from tape, applies the before-image blocks and puts the database back online.
- **Online media recovery:** Enables a backup utility to automatically roll forward, a restored database using transaction logs. Note that this also establishes an end point in time, should you want to recover to a point before the end of the logged activity.

During the backup process, users are able to access and update the databases. If some file is changed, the backup API provides pre-image buffers to the backup program with data at the point in time that the backup was started.

With transaction logging, the Domino API identifies changes in the databases using the logs. With tools such as Tivoli Data Protection for Domino R5, you can restore from a full backup and then apply only the changes on the restored database. This way, you can restore databases for a specific point in time.

5.3.5 Considerations when using transaction logging

There are some considerations that we have discovered in production situations, and we describe them here.

5.3.5.1 Transaction logging styles

One of the configuration settings for transaction logging is the log style: circular logging or archive logging.

Circular logging is the default style. When circular logging is enabled, Domino continually reuses a fixed amount of disk space for transaction logs, overwriting old transactions. Log segments are reused in rotation, oldest first. The maximum amount of space you can allocate for circular logging is 4GB, but we recommend you use 2GB.

The more space you utilize, the more time it takes before your log starts overwriting itself. If your circular log is long enough for only one day, then daily backups are still required for both logged and unlogged databases. Media recovery is possible only if all the required logs are still available; hence, this style offers less protection from media failure.

Circular logging offers four advantages:

- It provides you with faster recovery.
- Transaction logging (archive as well as circular) can increase server performance somewhat, because it allows database updates to be deferred.
- It allows you to continue to use an R4 backup architecture. Any non-R5 compliant backup program should be tested to verify that it copies only closed databases. If a logged database is opened when copied, there may be changes missing from it, and you'd need to run Fixup on it. However, in some cases Fixup might have to remove many documents to regain consistency. Therefore, any R5-compliant backup program must be tested to ensure that it does not backup files being updated. For example, the USS backup-archive client can back up open databases with integrity, as long as the proper options are set (static or shared static).
- You do not need to back up your transaction logs, since they are reused.

Archive logging is not the default setting. It requires you to deploy an R5-compliant backup solution. With archive logging, log files are created as needed, and are not overwritten until they are archived. Thus, the disk space required to support archive logging is dictated by the amount of time between incremental backups (typically once per day). As a rule, the daily log size is usually 3% to 5% of the total size of the active databases on the server, but

each installation needs to determine its own log size. A good way to determine how much space you need for archive logging is to work with circular logging. This allows you to find out how much your Domino server writes each day without the risk of filling up your file system. After you have this information, you can plan how much space you will need for archive logging. A busy Domino transaction log may grow about 1.2 MB per day.

Prior to Domino R5.04, a log file could be archived only when filled. With R5.04, support has been provided for archive of the currently filling transaction log file using TDP.

If you choose archive logging, you will have to back up logs regularly. If they fill up, the Domino server stops. Archival of transaction logs needs to be scheduled at appropriate intervals to accommodate the size of the log and the change rate for logged databases. Allocate as much disk space to allow for transaction log writing as you would for a full backup of your databases. It is advisable to allocate extra space, in case you need to deal with any backup failures and to avoid filling up the file system. If possible, allocate enough disk space for two full backup periods.

You need to take steps to manage disk space with archive transaction logging; on the other hand, it allows you to implement incremental backups. Again, you can only use archive logging with an R5-compliant backup tool.

5.3.5.2 Database instance ID

When transaction logging is enabled, Domino assigns a unique instance ID (DBIID) to each database. Transactions recorded in the transaction log include the instance ID, which is used to match transactions to databases during recovery. If the DBIID changes for a database (as a result of a Compact with options, Fixup, full backup, or some other action), a new full backup must be done. At the time of the full backup, a database is stamped with the log ID and log number of the initial transaction log extent that can contain transactions against the database. This information identifies the initial log file to be searched during database recovery.

Incremental backups of Domino databases need to be scheduled at appropriate intervals to ensure that full backups are taken after DBIIDs change, and that potential restore processing times are kept to a minimum. Therefore, you should make a full backup of the database immediately after you submit a compact job using the *in-place, space recovery and reduced file size,* or *copy* style. This is because these types of compact jobs create new database instance IDs (DBIIDs). TDP automatically does a full backup of the databases, instead of an incremental one, in these cases. If you submit an

in-place compact job with *space recovery only*, the DBIID is not changed, so you do not need a full backup of the database.

If you have to run Fixup on a database, schedule a full backup of the database as soon after Fixup finishes as possible, because Fixup will create a new DBIID to the fixed database.

In summary, if you have Compact, Space Recovery and Reduced File Size, or Fixup runs scheduled, try to adjust them to occur right before a full backup.

More details about Compact, Update, Updall and Fixup can be found in *Lotus Domino for S/390 Release 5: Problem Determination Guide,* SG24-5599.

5.3.5.3 Backup

Backups created with DFSMSdss cannot be used in conjunction with the Domino transaction log to provide forward recovery.

The backup or restore utility must execute on the Domino server whose databases are being backed up or restored. Archived transaction logs can be used only by the Domino server from which they were archived.

5.3.5.4 Restore

Recovery times are shorter, but restore times are longer. This is because restore with transaction logging means that, first, the transaction logs and the last full backup of the databases will be restored. Next, all changes will be applied to the database. So the database restore process is more complex.

5.3.6 Conclusions on transaction logging

Consider using transaction logging when choosing your backup/recovery solution. If you do, make sure that its log extents are written into an isolated file system, to avoid I/O contention. For example, avoid a file system where other logs are present, such as console output. The transaction log causes a large amount of I/O, so its HFS should be configured properly. If you cannot dedicate an isolated HFS for the log extents, then it is probably better for performance to not use transaction logging at all.

If you have decided to use TDP for backup purposes, make sure to use at least one TDP client per DPAR.

Refer to Chapter 3.2, “Network connections” on page 36, for tips on planning your network to get the best throughput.

Chapter 6. Problem determination

This chapter gives suggestions on how to approach problems that may occur while running a large Domino system in a production environment on the OS/390 platform. Two redbooks, *Lotus Domino for S/390 Release 5: Problem Determination Guide*, SG24-5599, and *Debugging UNIX System Services, Lotus Domino, Novel Network Services, and other Applications on OS/390,* SG24-5613, describe in detail how to approach problems for Lotus Domino Release 5 on S/390. From a functional standpoint, the Domino server on OS/390 is the same as on other platforms, including the problem determination tools. In addition, OS/390 offers some unique problem determination capabilities relevant to Domino. In this chapter, we describe functions that are not available on Windows NT.

The primary requirement for Domino on OS/390 is for higher availability than on other platforms. In two specific areas, OS/390 has a different approach from other platforms. One is to capture enough information on the first occurrence of a problem to diagnose the cause of the problem in order to make a fix and apply it (hopefully) before the problem recurs. In doing so over the years, OS/390 has proven to be a very reliable operating system with high availability. The other is that when a failure occurs, it is isolated to a specific piece of work. For an operating system that runs multiple workloads on one server like OS/390, this is important. A failure of one workload will almost never affect OS/390 itself or other workloads on the system. OS/390 will isolate the failure to a single component, clean it up after failure, and optionally restart it. For instance, OS/390 forces an abend of any program that addresses memory outside of its authorized range. A dump that points directly to erroneous instruction is taken. Thus, the system is protected and the cause of the problem is easily diagnosed.

In order to help you capture enough material for problem determination, the following sections describe information and tools that are available on OS/390.

6.1 Tools in the OS/390 environment

- Syslog
- Joblog
- Operator commands
- USS commands
- Component trace
- Domino trace

- nsd script
- RMF Monitors II and III
- OS/390 Console Support for Domino (formerly known as DOMCON)
- SVC dumps
- CEE dumps
- Sliptraps
- SMF
- PTF checker
- Service Update Facility
- IBM problem databases

6.1.1 SYSLOG

Because Domino runs under OS/390 using UNIX System Services (USS), OS/390 writes messages to the system log during startup and shutdown. OS/390 also routes error messages to the syslog.

6.1.2 JOBLOG

This option is only possible when the OS/390 Console Support for Domino is installed. For more information about this feature, see Chapter 7, “Operations and monitoring” on page 127. The only exception is RACF messages, which are in the joblog even without OS/390 Console Support for Domino.

All messages that Domino issues are kept in the joblog of the started task.

6.1.3 Operator commands

Basic commands that help in diagnosing the overall health of the system and commands regarding Domino can be very useful in problem determination. Some useful commands are discussed here.

6.1.3.1 Displaying network connection of the Domino server

All clients must be connected to a Domino server using TCP/IP in OS/390. You can see whether this connection is active or not by issuing the following command:

```
TSO PING <ip address>
```

6.1.3.2 Displaying Global Resource Serialization contention

The output of the `D GRS,C` command provides information about contention on the system. If a contention situation persists for an unreasonable amount of time, then a cancel of the address spaces causing the contention should be issued. For diagnosing the cause, a dump is useful.

We recommend issuing the D GRS,C command at least every 15 minutes using AOC or another automation tool.

6.1.3.3 Displaying omvs summary

The D OMVS,S command displays the status of OS/390 UNIX processes, file systems, servers, and the BPXPRMxx parmlib member set by initialization or specified by the SET OMVS=xx command.

6.1.3.4 Displaying address space ID

The D OMVS,A=ALL or D OMVS,A=xx commands enable you to display process information of the USS address spaces. Using A=ALL gives you information on all USS address spaces, but you can also specify a single address space by using A=xx.

6.1.3.5 Displaying user ID

The D OMVS,U=<UID> command allows you to get process information for all processes associated with the specified TSO/E user ID.

6.1.3.6 Displaying a specific process

Using the D OMVS,PID=<processid> command enables you to display thread information (in decimal numbers) for the specified process ID.

6.1.3.7 Displaying options

If you want to list all of the options that are set in parmlib member BPXPRMxx, use the D OMVS,O command. Keep in mind that the actual status is displayed. If a previous SETOMVS or T OMVS command was given, then that information is displayed and not the contents of the BPXPRMxx member.

6.1.3.8 Changing options

You can dynamically reconfigure USS system characteristics in the BPXPRMxx member using the SETOMVS command. For instance, SETOMVS, MAXPROCUSER=xx changes the MAXPROCUSER parameter to xx.

6.1.3.9 Displaying file system information

The D OMVS, FILE command gives detailed file system information on currently mounted files.

6.1.4 USS commands

To navigate in the UNIX System Services environment, it is very helpful to have a short reference of the most common UNIX commands. Table 4 covers

most of the commands related to problem determination, in alphabetical order.

Table 4. Useful UNIX commands

Command	Short description	Example
chmod	Changes the access permissions or modes of the specified file or directory. Modes are read(r), write(w), and execute(x).	chmod -w /usr/lpp/test
chown	Changes the owner or group of a file or directory.	chown 755 /etc/profile
df	Displays the amount of free space in a file system and the total amount available. The -Pk options list complete information on the space used, in the following order: File system name Total space Space used Space free Percentage of space used File system root	df -Pk
domps	Displays thread information. This shell command will be used for better problem determination especially during “server not responding” situations.	domps
echo	Writes argument to standard output.	echo $PS1
env	Displays or sets the environment variables.	env
kill	Ends a process by sending a signal.	kill 715 (715 is a processid)
ls	Lists files and directories. The -l option displays permissions, links, owner, group, size, time, and name.	ls -l
ps	Displays a list of currently running processes.	ps -ef
pwd	Displays the fully qualified pathname of the current working directory.	pwd
su	Switches to superuser or another ID.	su uid / use exit to switch back

For a complete list of all available commands and parameters, refer to *OS/390 UNIX System Services User's Guide*, SC28-1891.

The following figures display output examples of several of the commands:

df command

```
HAUFF @ SC63:/u/hauff>df -P /usr/lpp/lotus
Filesystem          512-blocks        Used  Available  Capacity Mounted on
OMVS.DOMINO5.PROD.HFS   784800      695968      88832       89% /usr/lpp/lotus
```

Figure 47. Sample output of the UNIX command df -P

env command

```
DOMINO4 @ SC63:/u/domino4>env
Mail=/usr/mail/DOMINO4
HOSTNAME=SC63
PATH=/bin:.:/usr/lpp/java/J1.1/lib/mvs/native_threads:/usr/lpp/lotus/bin:/usr/lp
p/lotus/bin/tools:
SHELL=/bin/sh
_WORK_UNIT=SYSDA
_INCDIRS=/usr/include /usr/lpp/ioclib/include
Notes_Directory=/domino4/notesdata
_PLIB_PREFIX=CEE
PS1=$LOGNAME @ $HOSTNAME:$PWD>
_CEE_RUNOPTS=AL(ON) ENVAR(_BPXK_SETIBMOPT_TRANSPORT=TCPIPMVS) POS(ON)
_=/bin/env
CLASSPATH=:/usr/jpp/Java/J1.1/lib/classes.zip
LOGNAME=DOMINO4
STEPLIB=none
LANG=C
LIBPATH=/lib:/usr/lib:.:/usr/lpp/Java/J1.1/lib/mvs/native_threads
_SLIB_PREFIX=SYS1
TERM=vt100
_CLIB_PREFIX=CBC
_BPX_SHAREAS=YES
HOME=/u/domino4
TZ=EST5EDT
MANPATH=/usr/man/%L
NLSPATH=/usr/lib/nls/msg/%L/%N
Notes_SHARED_DPOOLSIZE=8388608
```

Figure 48. Sample output of UNIX command env

ps command

```
HAUFF @ SC63:/u/hauff>ps -a
       PID TTY       TIME CMD
 268435467 ttyp0002  0:43 /bin/find
 117440529 ttyp0001  0:01 /bin/sh
 100663327 ttyp0001  3h05 /usr/lpp/lotus/notes/latest/os390/cconsole
 234881059 ttyp0002  0:43 /bin/sh
        38 ttyp0003  0:02 /bin/sh
  50331687 ttyp0003  2h14 /usr/lpp/lotus/notes/latest/os390/cconsole
 671088680 ttyp0002  0:43 /bin/ps
```

Figure 49. Sample output of the UNIX command ps

domps command

The domps command was introduced in Release 5.0.3. It provides thread information for root cause analysis by IBM.

6.1.5 Component trace

A CTRACE can be turned on for OMVS to gather useful information for doing root cause analysis of performance problems. See *Lotus Domino for S/390 Release 5: Problem Determination Guide*, SG24-5599, section 8.5.2. A CTRACE can be turned on for other components as well. In the case of difficult network problems, a CTRACE of TCP/IP should be considered.

6.1.6 Domino trace

As of release 5.0.3, it is possible to dynamically turn a DTRACE on and off. The DTRACE is a function in Domino that enables the systems programmer to follow the flow of load modules being used in USS. DTRACE is designed to collect data for highly intermittent problems which may occur on production servers. It is provided as an IBM support tool only. This tool makes it possible to record voluminous internal Domino trace records, both in memory and in a wrap HFS file with a specified maximum size, without filling up your notes.log database or console output file. This is used by IBM/Lotus for root cause analysis.

6.1.7 nsd script

The nsd.sh script is located in /usr/lpp/lotus/bin/tools/diag directory. It gathers information about your Domino server, such as environmental variables, job call stack, job logs, and shared memory. See *Debugging UNIX System Services, Lotus Domino, Novell Network Services and other Applications on OS/390,* SG24-5613, section 2.2.1, for more details.

6.1.8 RMF Monitors II and III

The RMF component of OS/390 provides detailed information on the utilization of resources, such as processor, I/O, and memory. RMF Monitor II provides real-time displays of resource consumption, enqueues, and more. DELTA mode allows for the displayed statistics being *update*d when you press Enter, to show how much resource was used since the last time the Enter key was pressed. Monitor III, the work delay monitor, can display resources that are slowing down the Domino server. For instance, if the Domino server is waiting for CPU resources because other work is consuming them, it will show you this. RMF III can also display enqueue and I/O delays.

6.1.9 OS/390 console support

This feature provides routing of messages to the OS/390 console under SDSF. For in-depth information, see 7.1, “Using OS/390 Console Support for Domino” on page 127.

6.1.10 SVC dumps

When Domino fails or abends, OS/390 can automatically produce an SVC dump of the system. When necessary (for example, in case of a hang) the operations department can also manually dump the address spaces. The dump is analyzed by the IBM Support Center using IPCS. To handle a Domino-related SVC dump, the dump data sets should be 2000 MB.

6.1.11 Sliptraps

A sliptrap tells OS/390 to capture requested information when certain specified conditions are met. For example, you can specify that any time a specified message is produced, OS/390 should take an SVC dump to an OS/390 dump data set.

6.1.12 CEEdumps

Because Domino is written in C language, it runs under the OS/390 Language Environment. If LE encounters errors, it can trigger a CEEdump. This dump is normally placed in the /notesdata subdirectory and contains a date and time stamp and process ID. The IBM Support Center uses these dumps for root cause analysis of server code processes. See *Lotus Domino for S/390 Release 5: Problem Determination Guide,* SG24-5599, section 8.6.

6.1.13 SMF

SMF collects data about processor, storage, and I/O consumption for address spaces in OS/390, in more detail than RMF does. For a Domino workload, the following record types are relevant:

- **Record type 30: Common Address Space Work**

  Record type 30 contains information about CPU and storage by address space, as well as file reads and writes and data blocks transferred to disk.

- **Record type 42 subtype 6: DFSMS Statistics and Configuration**

  This record shows the activity of all the UNIX files in each HFS data set, such as number of I/Os, average connect, disconnect, pending and control unit time, and response time.

- **Record type 92 subtypes 5 and 11: UNIX File System Activity**

Subtype 5 contains information about the UNIX file system. Subtype 5 contains information on the requests made by to the UNIX files by the Domino address space, such as read and write calls, directory I/O blocks, blocks read and written, and bytes read and written.

- **Record type 108**

  Starting with Release 5.0, the Domino server writes information into this record type. Some of the information includes:

  - Current number of users
  - Currently connected active users or users that have been active in the last 1,3,5,15 and 30 minutes
  - Average size of Domino mail and SMTP messages delivered to local users and other servers
  - Total transactions in interval
  - Total number of pool threads (Server_pool_tasks)
  - Number of virtual threads currently in use
  - Transaction sections for each transaction type showing the number of transactions processed in the interval
  - Accumulated milliseconds of response time

6.1.14 PTF checker

IBM offers an aggressive maintenance policy. It is crucial for the customer as well as for IBM, to keep maintenance as up-to-date as possible. Fixes should be applied as soon as possible to avoid known problems. Some customers did not apply all the fixes that were called for, and ran into problems. For the implementation of Domino Release 5.0, it is absolutely crucial to have all maintenance applied, whether it seems to affect Domino or not.

To help you keep up-to-date with your maintenance, you can run the PTF checker tool. This tool is an MVS batch job, downloadable in text format, that tells you if you are missing any needed PTFs. Or, you can do a manual check against the list of PTFs. For either maintenance choice, visit the following Web site:

`http://www.s390.ibm.com/products/domino/servchoice.html`

6.1.15 Service Update Facility

OS/390 Service Update Facility (SUF) is a Java-based OS/390 software tool that makes ordering and receiving OS/390 software quick and easy using the

Internet. As we stated, it is very important to be as up-to-date as possible with your maintenance level. A large percentage of problems concerning Domino are caused by maintenance not being current.

For details regarding prerequisites, entitlement, and how to obtain SUF, refer to the OS/390 SUF homepage at:

`http:/www.s390.ibm.com/suf`

6.1.16 IBM problem database

IBM maintains databases with solved and unsolved problems. The location of these databases varies from country to country. It is best to contact your local IBM Support Center to find out how to access these databases. A good starting point is to visit the following Web site:

`http://techsupport.services.ibm.com/support/s390`

The Lotus Support Center likewise has a database of Domino problems.

6.2 Performance

First of all, we need to state that performance is to be separated from network latency. The latter may cause slow response times, but it is not a system performance matter. It is a network matter.

When running a large Domino system, it is essential to consider performance issues carefully. In Chapter 4, “Performance” on page 61, we describe how to tune Domino to get the best performance. Chapter 4 addresses performance in a proactive way.

Still, performance degradation can occur. In this chapter, we cover the monitoring techniques and ways to react to any real-time performance problems.

Note: Performance degradation can occur if you are running traces or taking dumps to gather data for problem determination.

6.2.1 Monitoring techniques in OS/390

Using tools like RMF II and III from IBM (or similar tools from other vendors, such as BMC’s Mainview), it is possible to monitor in real-time resource consumption, enqueues, and more. RMF II (RMFMON) provides data on processor usage, storage, and address space resource usage. RMF III (RMFWDM) provides information on delays and causes of the delays. This

information is kept to be able to investigate delays after the event has occurred. In general, the things to monitor for Domino are no different from other workloads that require good response times. CPU, DASD response times, paging and storage should be monitored. Concerning Domino, in Chapter 4, “Performance” on page 61, we describe items to be monitored.

In order to monitor enqueues in OS/390, it is advisable to have an automated tool, such as System Automation for OS/390, and issue a `D GRS,C` command at intervals (at the most,15 minutes apart).

6.2.2 Monitoring techniques in Domino

There are several ways to examine performance degradation using Domino. Every Domino server has a log database, log.nsf, which can be examined. Domino has an events4.nsf database in which statistics and events, such as error messages, are stored. Operator commands such as `load stats <servername>` or `show statistics` can also be very useful when investigating performance.

6.2.3 OS/390 Console Support for Domino

A previously separate package for console support called DOMCON is now included in Release 5.0.3. It is described at length in Chapter 7, “Operations and monitoring” on page 127.

One point should be made here, however. The OS/390 Console Support for Domino has the capability of filtering messages in order to prevent flooding of the OS/390 console and the syslog with irrelevant messages. This filtering can hinder performance, because it requires cycles when the monitor is checking to see if the specific message is to be filtered or not.

6.2.4 Other products

Products by other vendors, or other IBM products, of course, can also be used to monitor performance.

6.2.4.1 IntelliWatch

Candle’s IntelliWatch Monitor is specially designed to monitor performance and can automatically react to preset triggers. These triggers can be controlled solely from the Domino environment and, therefore, are platform independent. However, the IntelliWatch code itself is platform dependent. IntelliWatch Monitor on the OS/390 platform is a very recent product. For more information, see the following Web sites:

`http://www.candle.com`

http://www.intelliwatch.com

6.2.4.2 Tivoli Manager for Domino

Tivoli Manager for Domino on OS/390 will be generally available in the second quarter of 2001. This port for the S/390 platform will offer the same functions available on other platforms, such as basic monitoring and alerting on server statistics and server availability. Tivoli Manager for Domino tightly integrates the Lotus Notes/Domino infrastructure and the Tivoli Management Environment (TME) to provide centralized end-to-end control and management for large Domino deployment to the Domino administrator. The details of how this is incorporated in S/390 will become known when the product is available. For additional information, watch the following Web site:

http://www.tivoli.com

6.2.4.3 Omegaview

Using Candle's Omegaview, it is possible to zoom in on started tasks (STCs) to monitor them. Domino, being an STC, can be monitored in that way for enqueues or delays. Disk space usage, response times and so forth, can also be monitored. See the following Web site for more information:

http://www.candle.com

6.2.4.4 Mainview

BMC's Mainview is a collection of monitoring tools. Mainview for MVS is a similar tool to Omegaview, but with built-in RMF-like screens. Mainview for USS is designed to monitor HFS. See the following Web site for more information:

http://www.bmc.com

Chapter 7. Operations and monitoring

This chapter describes the use of the OS/390 Console Support for Domino, operational procedures, and automated and scheduled procedures.

- Using OS/390 Console Support for Domino
- Automated procedures
- Scheduled procedures
- Tivoli Manager for Domino
- Candle Intelliwatch

7.1 Using OS/390 Console Support for Domino

Domino servers are usually administered from a Telnet or rlogin session or in the background using rc.notes. If connectivity to the Telnet or rlogin session is lost, the only way to issue Notes commands is the use of the Notes live console or the OS/390 console. From an OS/390 point of view, we recommend the use of the OS/390 Console Support for Domino, simply called *console support*. Formerly known as DOMCON, which was an as-is add-on product, the new console product is now fully integrated and supported with R5.0.3 of the Domino server.

Domino console provides system and status messages on the OS/390 console. For any invocation of a DOMINx procedure, at least one message appears. Errors are indicated by a DOM99609E message, which contains an error code that further describes the problem. The error message and this code should provide enough information to solve most problems.

Figure 50 on page 128, shows a high-level view of a single Domino partition. It shows how commands can be issued and various logs viewed on an S/390 server.

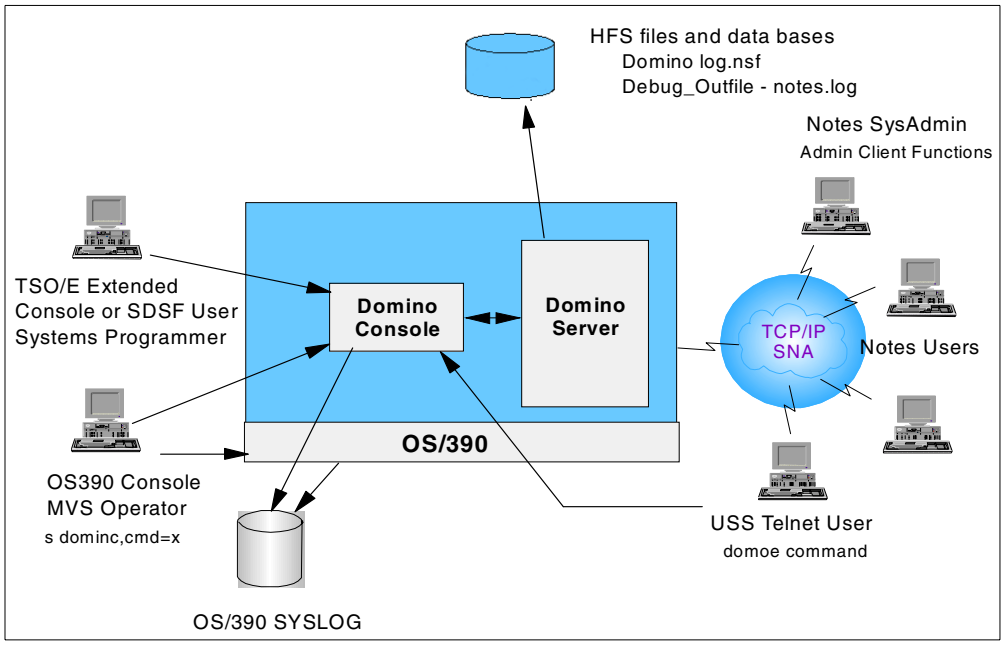


Figure 50. Issuing commands to Domino from different sources

The console support provides:

- The ability to manage multiple Domino partitioned servers from the same OS/390 console, or from a single UNIX System Services Telnet or rlogin session.
- The ability to start the Domino server automatically at IPL time.
- The capability to do an orderly shutdown of the server from the OS/390 operator's console.
- The ability to start and terminate the messages sent to the console, independently of starting and stopping your Domino server.
- The ability to establish filters to limit what information is sent to the OS/390 console for Domino.
- An interface to automation systems that can start, stop, or send commands to the server in response to error situations.
- The ability to coordinate starting and stopping the Domino server with the Domino for IBM HTTP Server (Web connector).

The following OS/390 system console procedures are provided:

- DOMINS: Starts a Domino server and the monitor process which routes messages to the operator's console.
- DOMINK: Shuts down a Domino server and cleans up any leftover resources after a server has stopped.
- DOMINM: Stops and restarts output from the Domino console to the OS/390 system console.
- DOMINC: Sends a command from the console to a Domino S/390 server.

Note

Even when monitoring is disabled, all output is kept in the notes.log file which is located by default in `/yournotesdata/noteslog/notes.log`.

The UNIX System Services command, `domoe`, can also be used to issue Domino console commands and display the output of the command from any UNIX command line interface (Telnet, rlogin, OMVS or ISHELL session).

The Domino console support uses the following OS/390 user IDs:

- Domino console ID, which has root security access
- Domino server IDs, which own the Notes data directory and are used to start the server
- Domino and Lotus Notes administrator IDs, used to Telnet or rlogin to UNIX System Services and invoke `domoe`

Note: Be aware that you should track your HFS files on a regular time basis for the available amount of space. The Domino console support writes several log files to the `/yournotesdata/noteslog` directory as well as to the `/u/domcnsl/logs` directory. A good tip is to allocate a separate HFS for the Domino console logs.

You may use the cron daemon or BPXBATCH to delete log files that you do not want after a couple of days or weeks. An example of a cleanup shell script which does this can be found in A.1, "Sample shell script to clean up UNIX files" on page 163.

Figure 51 on page 130, shows the output on SDSF of the Domino command `show server`, sent to the server using procedure DOMINC.

```
S DOMINC,CMD='SH SERV'
+SRV01 SH SERV
+SRV01
+SRV01 Lotus Domino (r) Server (Release 5.0.4a  for UNIX) 09/28/2000
08:58:54 AM
+SRV01
+SRV01 Server name:            wtsc67oe/Itso
+SRV01 Server directory:       /notesdata
+SRV01 Partition:              /notesdata
+SRV01 Elapsed time:           5 days 21:14:48
+SRV01 Transactions/minute:    Last minute: 0; Last hour: 0; Peak: 516
+SRV01 Peak # of sessions:     6 at 09/26/2000 10:01:33 AM
+SRV01 Transactions:           5479
+SRV01 Availability Index:     100 (state: AVAILABLE)
+SRV01 Message Tracking:       Not Enabled
+SRV01 Shared mail:            Not Enabled
+SRV01 Number of Mailboxes:    1
+SRV01 Pending mail:  0        Dead mail:  0
+SRV01 Waiting Tasks:          0
```

Figure 51. Issuing a command with Domino Console on SDSF

Starting with Domino for S/390 R5.0.3, address spaces used by the S/390 Domino server can be given names that are easier to identify. You can assign unique job names to every task associated with the server by overriding the default job name prefix with a name of your choice. The new MVS job names consist of the prefix you defined, plus the first two characters of the name of the added task being started (for example, RO for router) and a one-character suffix which goes from 1 to 9 and A to Z (or SER if it is the main server process).

Using the MVS job names, it is now easy to identify which Domino task belongs to which MVS started task. Table 5 shows an example of using SRV01 as the job name prefix for our server.

Table 5. Relationship between MVS started task names and domino tasks

MVS started task name	Domino task
SRV01SER	Server
SRV01MO1	Monitor
SRV01RO1	Router
SRV01AM1	Agent Manager
SRV01HT1	Http Task
SRV01RE1	Replicator

Migration Note: If you are migrating from a pre 5.0.3 version (and using the old DOMCON product), you may have to choose different server alias names

and update your server-specific variables accordingly. Use a saved copy of the domino_global_env file from the earlier version as a reference to map individual variable values to the new file.

See Chapter 9.2, “Administration tips” on page 144, when running the server without the Domino console to take advantage of MVSJOBNAMES.

Using the Domino console, an alias name is set for the server name in the domino_global_env configuration file. Figure 52 shows an example of this file.

```
#
# Partition Server using an alias name 'SRV01' and SAF ID 'ebbers1'
#
export SRV01_NOTESDATAD='/notesdata'
export SRV01_safid='ebbers1'
export SRV01_start_srv_sleep=300
export SRV01_kill_srv_sleep=120
export SRV01_saf=0
```

Figure 52. Excerpt of a domino_global_env configuration file

The value for `SRV01_kill_srv_sleep` may be too small for a large installation. You should check the notes.log to see how long it takes until the server shutdown has been completed after the `quit` or `exit` command is executed, and adjust this value accordingly.

Figure 53 shows an example of how the different MVS tasks are named.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC67  SC67     PAG    0 SIO    13 CPU  15/ 13  LINE 1-11 (11)
COMMAND INPUT ===>                                            SCROLL ===>
NP   JOBNAME  STEPNAME PROCSTEP JOBID    OWNER    C POS DP REAL PAGING
     SRV01MO1 STEP1             STC27251 EBBERS1    LO  FF  263   0.00
     SRV01UP1 STEP1             STC27256 EBBERS1    IN  F7 7744   0.00
     SRV01SER STEP1             STC27252 EBBERS1    IN  F7  11T   0.00
     SRV01RO1 STEP1             STC27254 EBBERS1    IN  F7 7996   0.00
     SRV01RE1 STEP1             STC27255 EBBERS1    LO  FF 7751   0.00
     SRV01AM1 STEP1             STC27257 EBBERS1    IN  F7 7790   0.00
     SRV01AM1 STEP1             STC27258 EBBERS1    IN  F7 8275   0.00
     SRV01AD1 STEP1             STC27259 EBBERS1    IN  F7 7879   0.00
     SRV01SC1 STEP1             STC27260 EBBERS1    IN  F7 7704   0.00
     SRV01EV1 STEP1             STC27261 EBBERS1    IN  F7 7906   0.00
     SRV01ST1 STEP1             STC27262 EBBERS1    IN  F7 7688   0.00
```

Figure 53. Display of the Domino tasks

For more information on the Domino console support, refer to the database *OS/390 Console Support for Domino Installation and Users Guide*, which is delivered on the Domino code CD.

7.2 Using procedures

This chapter describes how to run operational procedures for Domino on OS/390.

7.2.1 Automated procedures

Message handling in UNIX is done by using *stdin* into UNIX and *stdout* leaving UNIX. These messages are sent to the Notes log, as shown in Figure 54.

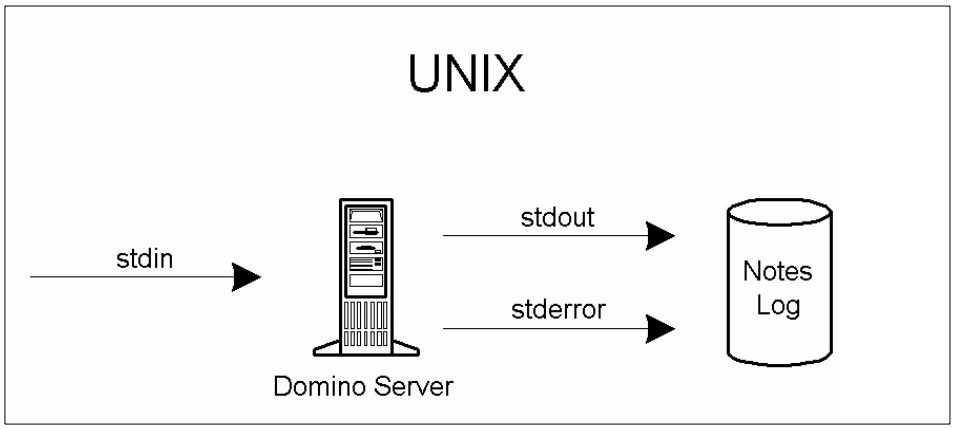


Figure 54. UNIX message handling

By using the DOMINC started task of Console Support for Domino, commands can be issued from the OS/390 console and routed to the Domino server. Console Support for Domino also routes messages from the server to the OS/390 console during startup and shutdown of the server. All of this makes it possible to use System Automation for OS/390 or any other product designed to perform system automation to trigger Console Support messages and send commands to the Domino server.

By running the DOMINM started task, all messages from the Domino server are routed to the OS/390 console until DOMINM is stopped again. These messages could also be triggered by System Automation for OS/390. A substantial drawback to this is the sheer number of messages routed to the OS/390 console. Filtering these messages can cause performance

degradation. To prevent the console from flooding, DOMINM is, by default stopped, which means that if you send a command to the server, no output is returned to the OS/390 console. Another limitation for automation is the fact that Domino messages have no message ID. Messages generated by Console Support itself do have message IDs, as shown in Figure 55.

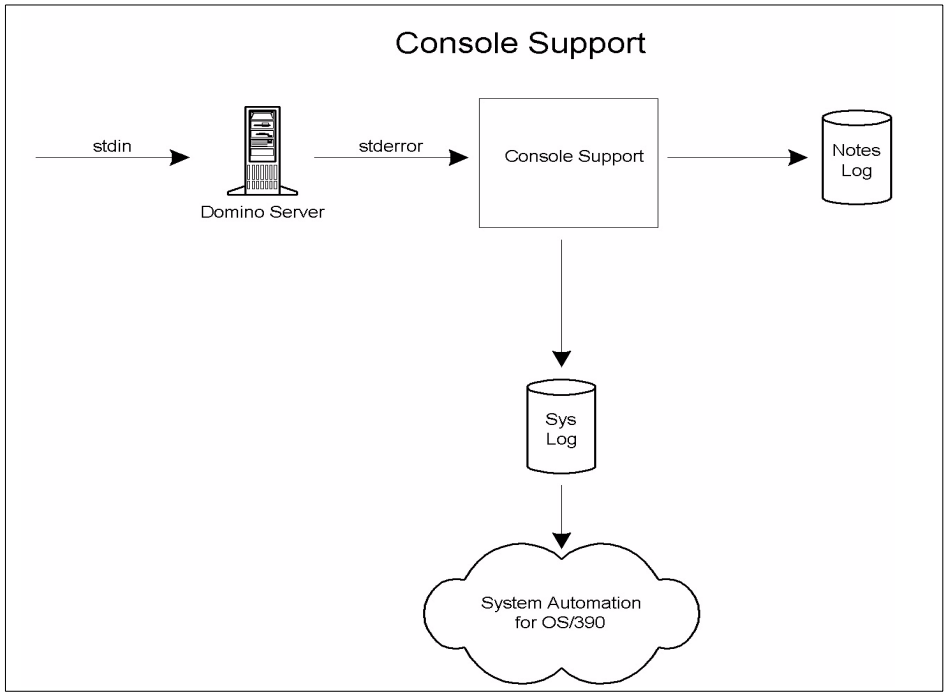


Figure 55. Message handling using Console Support

7.2.2 Scheduled procedures in Domino

Server tasks perform complex administration procedures; for example, compacting databases and updating indexes. You can run a server task manually, by loading the task at the server console. Or you can run the task automatically when the server starts by adding the name of the task to the `ServerTasks` or `ServerTasksAt` settings in the notes.ini file. In addition, you can create a Program document in the Domino directory to run a task at scheduled intervals.

In addition to the server tasks that Domino supplies, you can write and run custom tasks that are stored as OS/2 command files, Windows batch files, UNIX shell scripts or programs, and API programs.

7.2.2.1 In the notes.ini file

Many tasks run, by default, at specific times. You can schedule additional tasks to run by editing one of the following settings in the notes.ini file:

ServerTasks: starts tasks automatically every time the server starts up.

ServerTasksAt: starts tasks at a specified time.

7.2.2.2 From the console

To start a scheduled procedure from the console, complete the following steps:

1. From the Domino Administrator client, click **Server** -> **Status.** Then**,** click **Console** to open the console for the server on which the program runs.
2. Next, type:

   `load taskname`

   where `taskname` is the name of the server task that you want to run.

7.2.2.3 In a program document

To run a task on a server at a regularly scheduled time or at server startup, create a program document in the Domino directory. You can also use a program document to run an OS/2 command file, a UNIX shell script or program, or an API program.

If you create an OS/2 command file, UNIX shell script, or API program, you can use any of these characters for the name: A through Z, 0 through 9, ampersand (&), dash (-), period (.), space ( ), underscore (_), apostrophe (‘), and forward slash (/). Do not use a backward slash (\) or any other character, because unexpected results can occur.

1. From the Domino Administrator, open the Domino directory. Go to the **Servers** view and open the Server document.
2. Choose **Create** -> **Server** -> **Program**.
3. On the **Basics** tab, complete the following fields:

Field	Enter
Program name	The name of the server task you want to run. On an OS/2 server, if you use CMD.EXE as the program name, start the command line with /C, followed by the instructions for the batch procedure you want to run
Command line	The command that starts the task, including any arguments to the command.

Field	Enter
Server to run on	The full hierarchical name of the server on which to run the task.
Comments	A program description or additional information.

4. Click **Schedule**, and complete the following fields:

Field	Enter
Enabled/disabled	Choose one: 1. Startup to run the program only when the server starts. 2. Enabled to schedule the task to run at specific times.
Run at times	The first time of the date you want the task to run
Repeat interval of	The number of minutes before the task should run again
Days of week	The days of the week when the task will run.

5. (Optional) Click **Administration**, and then enter the names of additional owners.
6. Close and save the document.

   Tip*:* To view all tasks scheduled to run on a server, use the `Show Schedule` command.

7.3 Additional products

There are other products designed to perform daily operational procedures for Domino on OS/390. Tivoli Manager for Domino and IntelliWatch are among the most promising ones. See 6.2.4, “Other products” on page 124 for more details on these two products.

Chapter 8. Maintenance

This chapter provides guidelines on how to maintain your Domino server and OS/390 system software. It explains the Lotus maintenance philosophy and discusses the following topics:

- Domino server maintenance
- MVS maintenance
- Checking maintenance levels
- Service strategies
- Different Domino levels in an LPAR

8.1 Domino server maintenance

The Domino server on S/390 is not maintained using SMP/E. Instead, the Notes server is updated with a QMR/QMU process, which is a replacement of the code (either complete or incremental).

Lotus Development produces Quarterly Maintenance Releases (QMRs). These are scheduled quarterly updates to the major releases. For example, Release 5.0.4 is a quarterly update for the R5.0 code stream. QMRs are released approximately every 90 days for Domino and Notes. QMRs include all QMUs (see next paragraph) for the quarter. Lotus has two QMRs each quarter, one for each active code stream (NT and UNIX).

Lotus Development also produces Quarterly Maintenance Updates (QMUs). This is a method of shipping a small collection of critical fixes on an as-needed basis before the next QMR. These are typically used to fix problems introduced in a QMR.

Distinguishing between a QMR and a QMU is easy. A QMR is represented by a number, while a QMU has an additional letter. For instance, the first QMU for QMR 5.0.4 was R5.0.4a. These releases and updates of Domino and Notes supersede the previous QMR. They contain all of the fixes in the previous release, plus the additional fixes for the problems being addressed in the QMU.

8.2 OS/390 maintenance

One of the first questions you may be asked by the Support Center is the maintenance level of your OS/390 system. This is because many problems in Domino are fixed by having the required maintenance level plus all the PTFs. You can find if your level is current by going to the S/390 Web site and viewing

the required PTF list. Or, you can download the PTF checker and run it. This utility checks your PTFs against the Domino prerequisites.

Note: This task should be performed at least every two to three weeks, either by using the checker or by manually reading the list. Both are found at the following Web site:

`http://www.s390.ibm.com/products/domino/servchoice.html`

8.3 Checking maintenance levels

A new Web page, shown in Figure 56, lists the Notes/Domino fixes included in a QMR/QMU, including the release dates of each QMR/QMU. This page can be found at the following Web site:

`http://www.notes.net/r5fixlist.nsf`

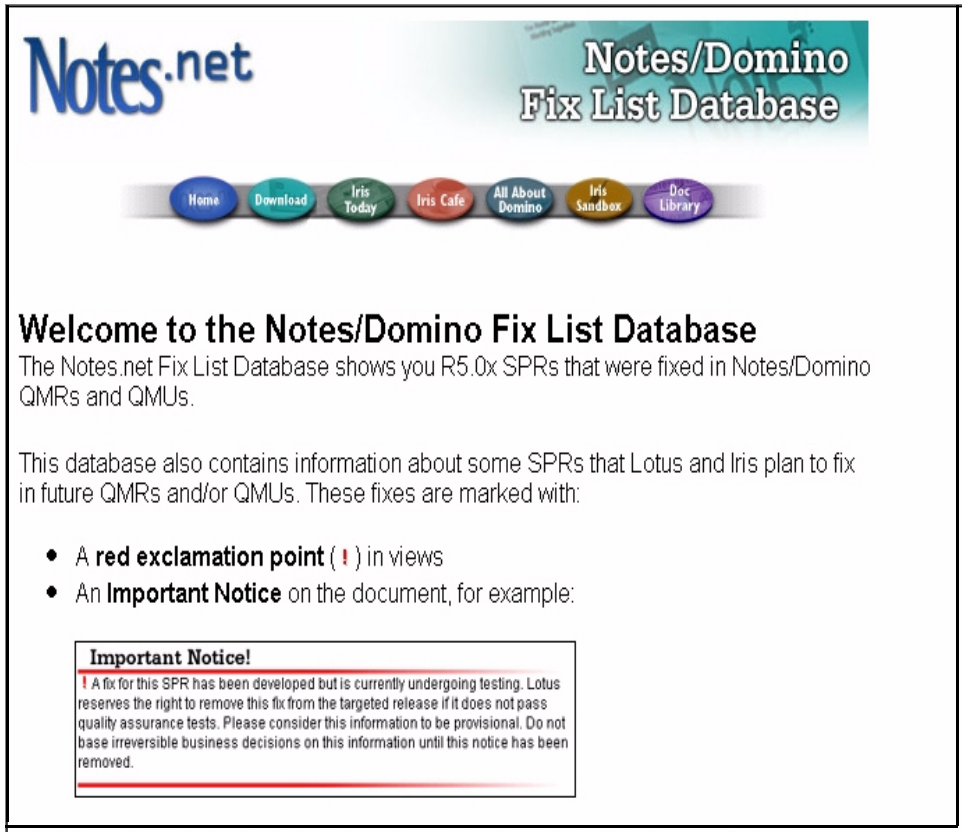


Figure 56. Notes/Domino fix list database

The Domino for S/390 incremental installers and OS/390 specific fix lists are available at the following Web site:

http://www.s390.ibm.com/products/domino/iinst.html

8.3.1 Service for Lotus Domino on S/390

Use the following Web sites to determine the service required for Domino running on OS/390.

- List of PTFs

  Organized by release of Domino and OS/390. The list also provides required service for the Domino for IBM HTTP Web Connector.

  http://www1.s390.ibm.com/products/domino/domptf.html

- PTF checker tool

  For Domino R5 and above with OS/390 V2R6 or V2R7. Download the file (JCL) and place it into an OS/390 PDS. Follow the instructions in the comments of the file before submitting the JCL.

  http://www1.s390.ibm.com/ftp/os390/domino/domp5067.bin

- PTF checker tool

  For Domino R5 and above with OS/390 V2R8. Download the file (JCL) and place it into an OS/390 PDS. Follow the instructions in the comments of the file before submitting the JCL.

  http://www1.s390.ibm.com/ftp/os390/domino/domp5028.bin

- PTF checker tool

  For Domino R5 and above with OS/390 V2R9. Download the file (JCL) and place it into an OS/390 PDS. Follow the instructions in the comments of the file before submitting the JCL.

  http://www1.s390.ibm.com/ftp/os390/domino/domp5029.bin

8.4 Service strategy

From our experience, there is only one effective strategy regarding service. Use the URLs above to determine the fixes that are required to match the Domino release you are running. Then, install all PTFs on this list, whether they seem to apply or not. Domino uses a wide variety of system services, so all listed maintenance needs to be applied.

8.5 Different service levels in an LPAR

With Release 5 of Domino, it is not possible to run Domino partitions with different service levels within an LPAR.

Chapter 9. Administration techniques

This chapter describes a set of administration tips, including:

- Mail policy implementation
- Administration tips
- Modifying ACLs on the server
- Differences between NT and UNIX systems
- Migrating users from other platforms
- API tools

9.1 Mail policy implementation

To implement the mail policies mentioned in 3.7, “Using mail policies and quotas” on page 56, you can use configuration document settings, mail file quotas, and third party tools. To offer the user a more structured alternative to excessive mail use, an archiving and an oversize mail solution needs to be implemented.

9.1.1 Archiving mail

The objective is to separate the active mail from the old mail. This requires the use of two mail files. The active mail file should be small in size, so it can more easily be managed and the indexer does not take much resource every time new mail is delivered. The archive mail file should only be accessed whenever an old e-mail needs to be retrieved. This means that you will see low user activity on this mail file. As this mail file does not require immediate updates, the load of the indexer is limited as well.

The most important question to ask is, which messages should be moved to the archive file and which should not. To keep things simple, on the active mailbox you can set a cutoff date, for example 60 days. You can then have a replica of the mail file on a Domino server which is dedicated to storing archiving mail files, or you can let the user store the archive mail file on the local hard drive. In the latter case, backup of the archive becomes more tedious. For this reason, this section assumes you are placing the archive on a dedicated Domino server. The mail file stored on this server does not have a cutoff date set. Because deletions that are made by a cutoff date do not replicate, they are not removed from the archive file.

The replication schedule between the mail server and the archive mail server (or the personal replication schedule, in the case of a local copy) can be set to replicate only once a day, or even less frequently. This prevents the users from only using their archive mail, because their most recent messages will

not be in there. This is why one archive mail server can service many regular, or active mail servers. Also, if you make this a billable service, not everyone would require this, which would lower the number of required archiving servers.

9.1.2 Quotas on mail files

On both active and archive mail files, it is advisable to use quotas to enforce the SLA-agreed maximum sizes. To set a quota on one or more databases, follow these steps:

1. Start the Domino administrator client.
2. Click **File ->Open Server**.
3. Enter the name of the server where the databases you want to set the quotas on resides and click **OK**.
4. Click **Files** and select the databases for which you want to set the quota.
5. Click **Tools ->Database**.
6. Select **Quotas**.
7. You can now set the quota for all selected databases.

When a quota is exceeded, no more mail is delivered to the mail file, if *Obey database quotas during message delivery* is set to **enabled** in the server's configuration document. The messages are returned to the sender with a delivery failure notification. The notification states that the recipients mail file has exceeded the quota.

9.1.3 Handling attachments

The main reason that mail files grow so large is the use of attachments. Even though e-mail is not designed to be a file transfer system, it is often used as such. This does not need to be a problem. However, users often do not detach the files from their messages once they have received them. This leads to large mail files and also larger messages when the “reply with history” button is used.

There are two things you can do to reduce the unnecessary use of attachments:

- Modify the mail file template so that the “reply with history” button does not include the attachments. Call your Lotus support contact before making this change to see if this will affect your support.

- Install the “Attachment Mover” tool. This tool will detach all attachments from the mail files and store them in a separate database. In the message itself, a link is created that points to the attachment.

9.1.4 Using message restrictions in configuration documents

From the configuration document, you can set various restrictions. From the **Router/SMTP** - **Restrictions and Controls**- **Restrictions** tab, you can set the maximum message size and the sizes of messages that you want to force to low priority. We recommend that you set the field `Obey database quotas during message delivery` to `enabled`. On the **Router/SMTP** - **Restrictions and Controls**-**Transfer Controls** tab**,** you can set times at which low priority messages should be routed.

These restrictions apply to the servers on which you set them. Depending upon the role of the server, this impact will vary. In Figure 57, an example is given for a section of the mail routing environment.

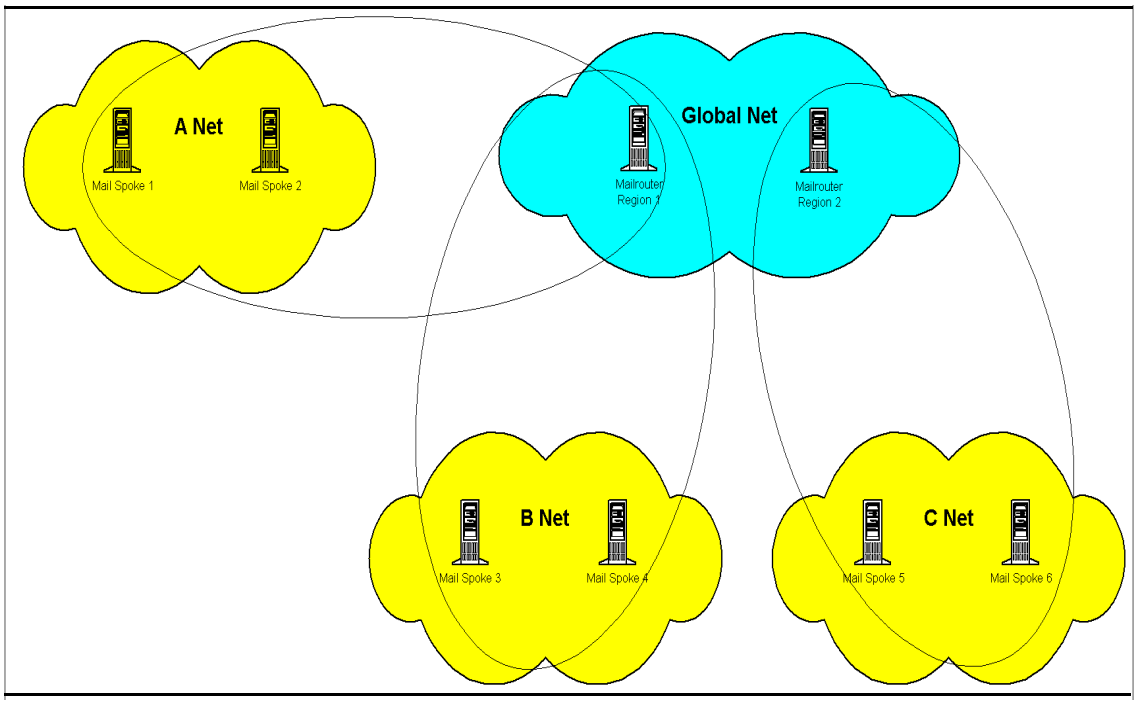


Figure 57. Section of the mail routing environment

In Figure 57, the names in the clouds represent the Notes Named Networks (NNNs). The mailrouter of region 1 is also a member of A NET and B NET. The mailrouter of region 2 is also a member of C NET.

Mail restrictions which are set on one of the spokes will only affect the spoke itself. Restrictions being set on the mailrouter of region 1 will affect all messages going between and outside A NET and B NET. However, this will *not* affect messages being routed inside these respective NNNs. So, for example, a message going from mail spoke 1 to mail spoke 2 is not affected. Restrictions on the mailrouter in region 2 will affect all messages going between and outside C NET.

9.2 Administration tips

For administrative activities, use Domino tools whenever possible, since they are generally platform independent and can run on any system that your Domino servers run on. Following are some administration tips:

- Remove old files from your notesdata and noteslog directories to prevent their HFS files from filling up completely. See A.1, “Sample shell script to clean up UNIX files” on page 163. This script also handles log compression using UNIX tools.
- Check file system size, and monitor HFS and disk space. You can use the sample script provided in A.2, “Sample shell script to monitor HFS space usage” on page 167, to find all HFS files that are 90% full.
- Check and release dead mail. Use the Events database to set an event monitor for dead mail (use the Dead.Mail statistic), and monitor the occurrence of that event. Define an action, such as paging or directing a message to the operating system log, that will cause notification to Domino administrators. See A.7, “Setting up an event monitor” on page 178.
- Limit the number of agents to those you really need. For Notes servers, the number should be as low as is possible. For HTTP servers, the concurrent Web agents justify a higher number.
- Monitor mail pending. Set a server threshold on the Mail.Waiting or Mail.TotalPending statistic that corresponds to the activity of your server. If you have a busy mail server (for instance, more than 500 mails pending delivery), there is an issue to resolve. You must observe your pending mail numbers to determine a reasonable threshold. Define a trigger in the events database and set your threshold in the trigger. For UNIX systems, you can use Sendmail separately from Notes to send event notification

messages that can no longer be sent by the Domino server itself. See A.7, “Setting up an event monitor” on page 178.

- Check whether the address book has replicated (nightly from hub to spokes). Using the events database, set a replication monitor. See A.7, “Setting up an event monitor” on page 178.
- Check for any NAB conflicts. Name and Address Book (now called Domino directory) conflicts require special care. They should be handled by Notes administrators. If there are many conflicts, consider writing a LotusScript agent, accessible only to Notes administrators, that removes the conflicts.
- Check and correct any corrupt messages. Use Domino Console to route “DbMarkCorrupt” messages to the MVS console for MVS message automation to run `Fixup -f` against the corrupted file automatically.
- Check whether the default ACL of mail files is set to “no access.” In many installations, this is accomplished by a LotusScript agent that enforces the default administrators group and corrects the default ACL to `noaccess`. See A.5, “Sample agent to set a default ACL on mail files” on page 175.
- Check whether NAB is still indexing in the afternoon. This is related to your Directory (address book) replication topology. Clearly define your replication topology for your Directory and check whether its indexing completes in time, regardless of whether this is afternoon or night. Contact an administrator if it does not.
- Run Compact on a weekly basis, for both system and mail databases, using program documents in the Directory, before doing a full backup of the files. If you use -B as compact option, the free space will be released. Note that if you have enabled transaction logging, the DBIID will change, so you should make a file backup of the database.
- Full-text indexes can grow quite large. If you need them enabled, one way to reclaim full-text index space is to delete them regularly. If you do not need them, you should disable the use of full-text indexes by setting a notes.ini variable: `update_no_fulltext=1`. Disabling full-text indexes on a server reduces server load and the amount of file space needed on the servers.
- Watch out for “Unknown OS” errors. They indicate problems in the HFS file system, and may need fixing by the OS/390 and Notes administrators.
- Include the following in your **.profile** in order to display job names when you start your Domino server from a Telnet session: **export Notes_OS390_JOBNAME_PREFIX=XXXXX.** If this parameter is not set, all MVS job names are related to the current user ID.

9.3 Modifying ACLs on the server

Customers coming from platforms other than OS/390 are accustomed to having a local Notes client available with the Domino server. This setup allows administrators (even with no access rights defined to specific server databases) to update the ACLs of these databases through the local client, except when Enforce Consistent ACL is turned on.

Although Domino R5 does support a local client on NT, it does not support local clients on any UNIX platform, including OS/390. One of the first questions that a Domino administrator from another platform asks is: how do I modify the ACL of a database when I am locked out of that database?

If Enforce Consistent ACL has not been turned on for a database, the easiest solution is to FTP the database to the administrator's workstation, fix the ACL, and then replicate or FTP the database back to OS/390. When using FTP, make sure the server is down, or FTP the file to a different file name and subsequently rename the file. This will have the least impact on the server.

If Enforce Consistent ACL has been turned on, access to the database is denied on any remote client, unless the user is given the appropriate authority in the ACL. The best solution is to ask the manager of the database to add the Notes administrator to the ACL of the database when possible.

For this situation, another option is through the C API Toolkit for Domino.

Not Ported

The ACL sample in the toolkit has not been ported to OS/390 environments, and may need modification to be able to run.

Within the /notesapi/samples/admin/acls directory of the Toolkit, there is a file called acls.c which contains a sample program with various scenarios to update the ACL of a database, one of which is adding a user with Manager access level to the ACL.

Theoretically, this program could be executed from a Telnet session of the user ID of the Domino server. For more information on implementing this program, see the readme.txt file in the /notesapi/samples/admin/acls directory.

We provide a sample program in A.4, “Sample program to modify an ACL” on page 169. Designed to run on OS/390 and AIX, it adds an entry to the ACL of

a database with access rights defined by parameters. You should adapt this program to your own environment, and compile and test it before using it.

Another sample is included in 9.6.5, “Modifying an ACL of a Notes database using JAVA” on page 160.

Third-party vendors also market tools that add a user's name or group to the ACL of a database. Table 6 lists some of these vendors. We also describe ACLmaster below, and there may be others.

Table 6. Administration tools vendors

Vendor	Web Site
IVE Technologies	http://www.ivesco.com
Percussion Software	http://www.percussion.com
4NF Information Technology Ltd	http://www.4nf.co.uk
Tiger Technologies	http://www.tigerti.com

9.3.1 ACLmaster description

Currently, if the administrator is not in the ACL of a database, the administrator must leave their workstation, go to the server's console to open the database locally in order to edit the Access Control List of the database. This works if there is a client on the server, and the server is nearby. ACLmaster for Domino can add or remove a user name or a Notes group to a database, and it can be executed from the server console. You can execute ACLmaster from the Domino remote console on multiple platforms, including Windows NT, IBM-AIX, OS/400, and OS/390. There are near-term plans for Linux and Sun-UNIX.

You can access this Web page for a free download:

http://www.akornn.com/download

For more information, see:

http://www.akornn.com/software

9.4 Differences between NT and UNIX systems

There are three major differences between NT and UNIX systems related to running Domino:

- Case sensitivity
- Disk space management

- Access to the operating system resources

Other differences include the ability to enable automatic recycling of a Domino server using a parameter in notes.ini, and the possibility of using UNIX sendmail to report on conditions that a Domino server cannot report if it is no longer sending mail.

9.4.1 Case sensitivity

Filename case sensitivity on UNIX can impact a Domino migration from NT to OS/390 or any other UNIX system in many ways, caused by the differences in the file systems. These differences lead to errors as observed by a Notes client in accessing files and applications.

Notes automatic filename conversions and caching may result in inconsistent behavior when accessing files and applications.

9.4.2 Disk space management

UNIX and NT servers have a different way of handling data across the volumes. In NT servers, one physical disk or an array of disks are associated as a logical drive. Each logical drive is represented with a letter, beginning from C. Each of these logical drives has its own hierarchy of directories, and directory space depends on the space available in the drive. In Figure 58 on page 149 there are three drives. We refer to the tlog directory in the first volume as C:\notesdata\tlog. Noteslog in the second volume is referred to as D:\notesdata2\noteslog.

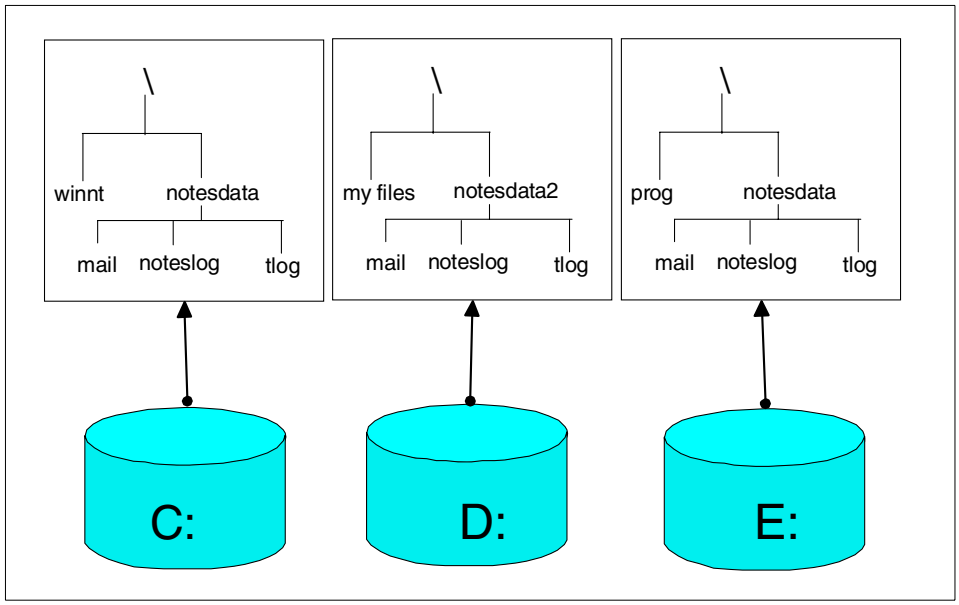


Figure 58. NT server and its drives

On UNIX, there is only one logical tree. Volumes are created as file systems and mounted on mount points. In Figure 59 on page 150, we can see a unique tree and smaller subtrees mounted in subdirectories, called mount points. In order to access one of these volumes, we just refer to its path, such as /notesdata1/mail/1.

This way of organizing the volumes across the tree allows you to explore I/O parallelism and increase data space. On the other hand, you have a new and different concern: how to administer the data space in file system space. You also to plan your tree so that it can allow you to add new file systems without major changes.

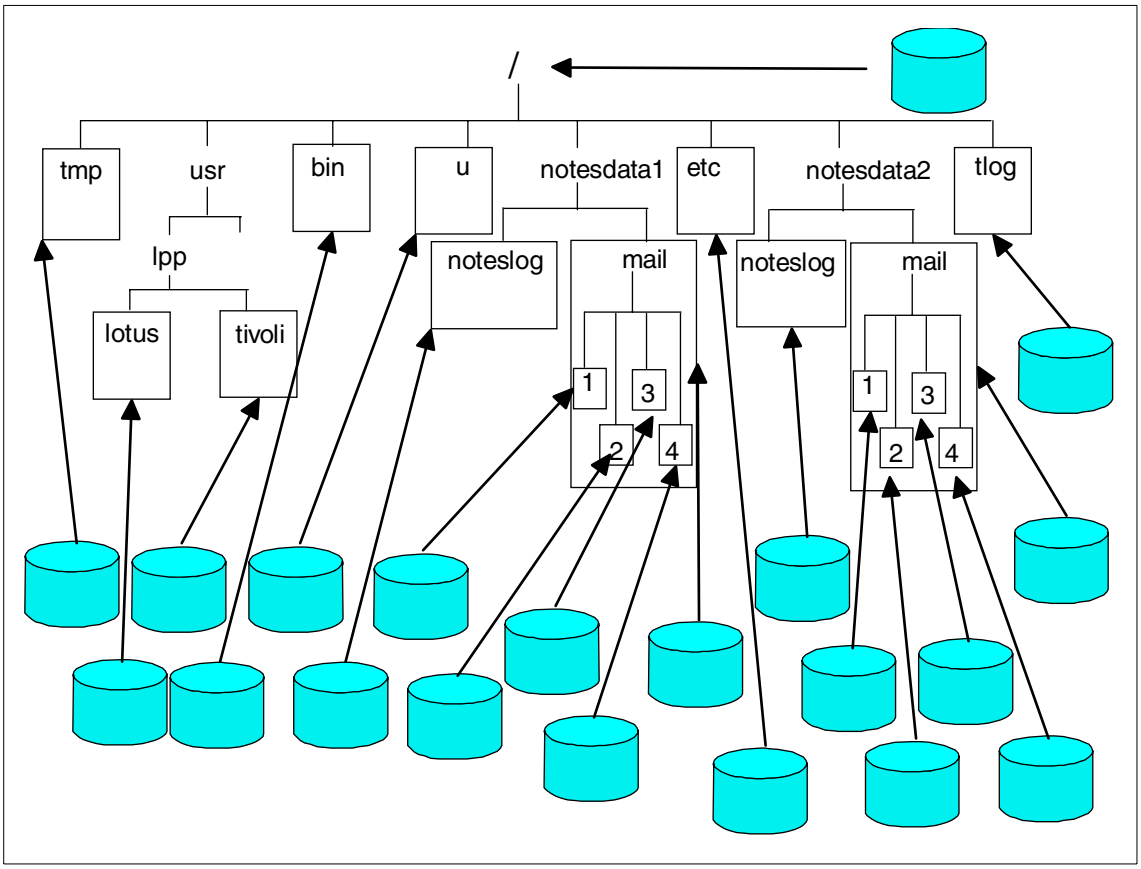


Figure 59. UNIX file systems

See 4.3.3, “HFS” on page 85, for more detailed information.

9.4.2.1 File pathnames

On all UNIX platforms, S/390 included, file pathnames are case-sensitive. NT preserves case, but does not enforce it. For example, on NT, apps\sales.nsf, apps\Sales.nsf and APPS\sales.nsf are all equivalent files. On UNIX systems, however, apps/sales.nsf, apps/Sales.nsf and APPS/sales.nsf are three distinctly different files to the operating system.

9.4.2.2 File pathname delimiters

There are other incompatibilities. UNIX uses forward slashes as a delimiter between directories and files in path statements, while NT uses backward slashes.

9.4.2.3 Drive letters

All devices in UNIX reside under the root directory (top of the UNIX tree) rather than under multiple drives identified through drive letters. Any references to drive letters in file pathnames results in errors.

9.4.3 Operating system administration

Notes administrators in NT are responsible for both Domino and the server. But in S/390 we have three distinct layers:

- Domino
- UNIX System Services
- OS/390

Domino administrators must share the server administration as described in 1.5, “Organizational responsibilities” on page 4.

9.4.4 Errors resulting from differences

Errors can occur from various situations. Thorough testing of any application before putting it into production is necessary.

9.4.4.1 Errors from a client perspective

From a client perspective, errors can be encountered when selecting database icons or bookmarks pointing to file names which do not match the case specifications of the files in the operating system.

Mail file names with incorrect case specifications in person documents within the Domino directory can result in mail delivery failures by the Domino mail router.

When entering URLs containing a Domino database or file from a Web client, errors are possible if the file name does not match the name in the operating system.

9.4.4.2 Errors from application code perspective

Errors can also surface in application code. Domino/Notes code developed in Formula Language or LotusScript to access server files may fail due to filename mismatches between the application code and UNIX.

For a complete list of UNIX platform differences in LotusScript, see the *Domino 5 Designer Help Database (LotusScript Language, Platform Differences).*

9.4.5 Automatic conversion of pathname delimiters

On the Notes client side, Windows NT directory delimiters are usually converted when accessing UNIX server files. For example, forward and backward slashes are interchangeable in database icons, bookmarks and mail file names in location documents. From a Domino mail server perspective, back slashes are properly handled if they are specified in mail pathnames within person documents in the Domino directory.

9.4.6 Caching and case sensitivity

Domino and Notes client caching must be taken into consideration when testing applications to uncover problems with filename case sensitivity and other UNIX issues. Domino stores database information on the server in DBCACHE. A database that has been successfully opened and closed (and is not being used by any other process) is cached to improve response time on subsequent opens. Once a database has been opened with the proper case specification, it becomes case-insensitive as long as it remains in cache.

Since DBCACHE is not static (over a period of time databases are dropped from it), users invoking applications may experience inconsistent results. For example, if a mail file has been opened with the correct case by an end user, all deliveries to that end user's mail file is successful while the file remains in cache, regardless of the mail file case specification in the user's person document.

When a user correctly specifies a database name in a URL, that database is cached. With the appropriate access rights, other users can access the same database using any case variant. If a database is in cache and a user bookmarks the URL page with the wrong case, the bookmark will work for a period of time and then stop working when the database is removed from cache.

Certain Formula commands, such as `@Command([DBExists])` do not work when the case or directory delimiter have been specified incorrectly; however, if cached, errors may not surface until the database is removed from both the Domino server and Notes client cache.

The CACHE.dsk file on the client contains design elements (such as forms and subforms) of recently accessed databases on the server to improve client performance the next time these databases are accessed.

9.4.7 Clearing caches

When testing applications for possible UNIX issues, it is important to appropriately clear server and client cache to obtain accurate test results.

9.4.7.1 Clearing server cache

The Domino database cache is cleared when the server is recycled or an administrator enters the console command `dbcache flush`.

The flush command must sometimes be invoked multiple times to clear the cache. For more information on server caching, see *Lotus Domino 5 -- Managing Domino Databases,* Lotus p/n CT73MNA.

9.4.7.2 Clearing Notes client cache

The easiest way to clear the client cache is to shut down Notes, delete the CACHE.dsk file, and then restart Notes. The CACHE.dsk file is recreated if not found by the client.

9.5 Migrating users from other platforms

When migrating users from other platforms to Domino servers on S/390, there are some platform-related issues to consider. OS/390 is a pure UNIX system seen from Domino; if your originating platform is UNIX-like, there are few changes required. Other UNIX platforms are AIX, HP-UX and Sun Solaris. However, when the originating system is an NT Domino server, there are a few file system issues to consider.

9.5.1 Addressing mail issues

When migrating mail users from NT to OS/390, the primary concern of Notes administrators is to make the move as transparent as possible to end users. Case sensitivity becomes a major concern because the mail file pathname must match in four places (possibly three places if the client is R4) for everything to work correctly:

- The database icon
- The bookmark (R5 clients only)
- The person document
- The operating system

Mail file names in location documents are not case-sensitive.

9.5.2 Local encryption considerations

In the case of a server consolidation or a move to a different server, databases with local encryption specified on the server must be handled separately. These databases have an encryption key unique to the original server, which means that if they are simply copied, the target server would be denied access to them. A “local access protection” error message in the Notes log.nsf identifies which files the S/390 server cannot access. You can monitor this message using IntelliWatch or the Notes Events database. If it is not a question of UNIX file permissions, then local encryption is probably the source of the problem. To resolve this problem, manually create a new replica of each of these databases on the S/390 server using a Lotus Notes client interface. When replication takes place, the encryption key is rebuilt.

9.5.3 Moving databases

The following three methods can be used to move a file from one server to another:

9.5.3.1 AdminP

AdminP can be used to move one or more mail files to one or more target servers. The administration process creates replicas “under-the-covers” for the databases specified in the mover request. This is available in the R5 administration client. Source and target servers must be running Domino R5 and be in the same domain. There is also a `move db` function which works for non-mail databases.

Lotus recommends replication and AdminP uses it. The `move mail file` function updates R5 clients, which vanilla replication does not do. For a detailed description of the rules to follow during this AdminP process, see the online Administration help database or *Lotus Domino R5 — Administering a Domino Server*, CT7VHNA and CT7VINA.

9.5.3.2 Replication

From a Lotus perspective, the recommended process for moving databases to a new server is replication. Replication guarantees full integrity of online databases, even if they are being updated at the time of replication. The disadvantage of using replication is that it is very slow and CPU intensive; a full database replication of hundreds/thousands of databases would not fit into a weekend migration window. However, you can start replication of databases far before an actual migration as a background task, so you are not bound by time.

Replication is multiprocessed, not multithreaded, which means that throughput can only be improved by running multiple replication tasks. The more replicator processes you allow, the greater potential for performance impacts to the server. One possible solution to this dilemma is to stage the creation of the full replicas on the Domino for S/390 server over a period of several weeks, well in advance of the migration cutover window. Then, update the replicas on S/390 through normal scheduled replication, until the time of migration. Since replication only transmits changes to databases, one last replication to all databases on the S/390 during the cutover window would take care of any changes not propagated through scheduled replication.

With both Domino R4 and R5, there is the ability to create replicas of multiple databases on other servers through the administration process. In R5, this area of Domino has also been enhanced to include drag-and-drop capability.

9.5.3.3 File system copy (FTP)

A method which many customers and services providers use to transfer large numbers of Domino files is FTP. This method should be used with caution, but is suitable for mass moving of files. It is frequently preferred because it is faster than Domino replication. It requires fewer Domino administrative skills to perform. Its one major disadvantage is that it does not guarantee database integrity if the source or target servers are online. However, if the following guidelines are adhered to, FTP works very well:

- Shut down the Domino source and target servers.
- FTP the data in binary mode, and make sure you change the extension for the file. Rather than `.nsf`, use `.bak` instead.
- While the target server is still down, run the following Domino utilities against all of the migrated databases:
  - `fixup [database path]` to perform consistency checks against databases
  - `updall [database path] -r` to update views and full-text indexes with databases
  - `compact [database or directory path]` to commit the database to the new on-disk structure if changing releases
- Rename the database to its correct extension `.nsf`
- Restart the servers.
- Send a test e-mail to all migrated users and resolve all delivery failure errors.

To reduce the downtime window for execution of these tasks, multiple instances of the same FTP task can be executed concurrently against different files; for example, against different directories. Through UNIX shell scripts, these tasks can be invoked in the background under the authority of the Domino server user ID. The `Compact` task will accept a directory as a command line option; `Fixup` and `Updall` will not. UNIX scripts must be written to invoke `Updall` and `Fixup` for individual files; a simple script could be written to loop through all the eligible files in a specific directory. Since these tasks are typically CPU-intensive, it is imperative to have sufficient processor capacity to achieve any kind of parallelism.

Tip: You can move the files using FTP in advance of the migration, then replicate the changes at migration time.

9.6 API tools

Domino application developers can extend the power of Lotus Domino with the toolkits provided by Lotus. The toolkits are extremely useful to perform actions on servers where no Notes client is available, or where no administrator client is available to perform maintenance functions locally. Such platforms are Linux and OS/390. As an alternative, the API tools allow you to manipulate Notes objects when the server or the client are not running.

However, be aware that compiled API programs are platform dependent. They may need re-coding and do need to be re-compiled for each platform you intend to use the API program on. Furthermore, the API programs may also need re-coding and re-compilation as new releases of Domino and the API become available.

A general recommendation is not to use the API programs if you do not really have a need for them. The available tools are:

- C API toolkits
- C++ API toolkits
- LSX toolkits

Which one to choose depends on the server platforms on which you run your tools, your Domino version, and the development skills available.

Besides using these toolkits, it is also possible to use Notes Java classes to perform Notes functions outside the Notes client or the Notes administrator client. See: 9.6.4, “Compacting a Notes database using JAVA” on page 159, and 9.6.5, “Modifying an ACL of a Notes database using JAVA” on page 160.

A list of available toolkits can be found at the following Web site:

http://www.s390.ibm.com/products/domino/domdown.html

9.6.1 Lotus C API for Domino and Notes

The Lotus C API for Domino and Notes provides a comprehensive, multiplatform C programming language interface to Lotus Domino and Notes. It gives C developers access to most of the operations available through the Notes client, as well as many features specific to the Domino server. Table 7 lists the various platforms and versions where the Lotus C API is available.

Table 7. Lotus C API for Domino and Notes: platforms and versions

Platform	Toolkit version
PC	Notes C API Toolkit Release 4.51 Notes C API 4.51 Patch Kit for Notes 4.55 Notes C API Toolkit Release 4.6.2 Notes C API 4.6.2 Patch Kit for Notes 4.6.4 Lotus C API Toolkit for Domino and Notes Release 5.0.3
UNIX	Notes C API Toolkit Release 4.51 Notes C API 4.51 Patch Kit for Notes 4.55 Notes C API Toolkit Release 4.6.2 Lotus C API Toolkit for Domino and Notes Release 5.0.3
S/390	Notes C API Toolkit Release 4.51a Notes C API Toolkit Release 4.61 Lotus C API Toolkit for Domino and Notes Release 5.0.3
Macintosh Power PC	Notes C API Toolkit Release 4.51 Lotus C API Toolkit for Domino and Notes Release 5.0.3
AS/400	Lotus C API Toolkit for Domino and Notes Release 4.6.2 Lotus C API Toolkit for Domino and Notes Release 5.0

You can also download the Release 5.0.4 Preview for PC. The C API toolkit is available to download at the following Web site:

http://www.lotus.com/developers/devbase.nsf/HomeData/homecapi

9.6.2 Lotus C++ API toolkit

The C++ API provides an easy-to use, object-oriented interface to Notes and Domino, which you can use to rapidly develop robust, high-performance applications for Notes clients and Domino servers. The C++ API consists of a

set of C++ classes and data types that enable applications to create, access, and manage Domino databases and database design elements.

The C++ API provides the ease-of-use of the LotusScript and Java APIs, but makes the full power of the Notes API available to your applications. The C++ API makes it particularly easy to work with rich text and database design elements. The combination of ease-of-use and power of the C++ API is unmatched by any other API for Notes and Domino.

Which version of the toolkit should you use? If you are developing applications to run on Notes 4.x, use Release 2.01 of the C++ API Toolkit. If you are developing applications to run on Notes 5.x, use Release 2.1.

9.6.2.1 Supported platforms

Table 8 lists the platforms that C++ API Releases 2.0 and 2.01 support.

Table 8. C++ API for Domino and Notes: operating systems and compilers

Operating Systems	Compilers
AIX 4.1.4	IBM C Set++ 3.1.3.2
HP-UX 11.0	HP-UX 11.0 ANSI C++ compiler
Windows NT 4.0 Windows 95	Microsoft Visual C++ 5.0 IBM VisualAge C++ 3.5 Borland C++ 5.0
Windows NT 4.0 for DEC Alpha	Microsoft Visual C++ 5.0
Windows 3.1 or 3.11	Borland C++ 4.52 (Release 2.0 only)
IBM OS/2 Warp 3.0 or later	IBM VisualAge C++ 3.00
Solaris SPARC 2.5.1	SparcCompiler C++ 4.1
Solaris Intel Edition 2.5	SparcCompiler C++ 4.01
Macintosh PowerPC™ System 7.5 or later	Metrowerks CodeWarrior Pro 2 (Release 2.0 only)
IBM OS/390 1.3.0 or later	IBM OS/390 1.3.0 or later C++ compiler
IBM AS/400 v4r2m0 or later	IBM VisualAge C++ for AS/400 compiler

The C++ API toolkit is available for download from the following Web site:

`http://www.lotus.com/developers/devbase.nsf/HomeData/homecpp`

9.6.3 Lotus Software Extensions (LSX) toolkit

The Lotus Software Extensions (LSX) toolkit expands LotusScript functionality by allowing developers to write custom modules that are tightly integrated with both the Lotus Notes and Lotus Suite environments. These modules, called Lotus Software Extensions (LSXs), are separately loadable modules that implement one or more custom classes.

An LSX is a dynamically loaded, shared module that exposes classes to Notes and Domino, or any other LotusScript host, such as SmartSuite. These classes can then be scripted and manipulated, just like the Notes and Domino native classes. Your LSX classes can be accessed from LotusScript or from Java.

In addition, an LSX can be used to integrate Lotus applications with other systems. When your LSX is loaded into the LotusScript host—typically your Notes client—your custom LSX classes appear alongside the native Notes classes in the Notes IDE as if they are native to the Notes environment.

This release of the toolkit brings all supported platforms to the same level with respect to source code and functionality. The release consists of software and documentation for Windows 95/98/NT, DEC Alpha NT, OS/2 Warp, and the UNIX platforms Sun SPARC Solaris, Solaris Intel Edition, HP-UX, AIX, Linux, and OS/390. The OS/400 platform is also supported.

The LSX API toolkit is available for download from the following Web site:

`http://www.lotus.com/developers/devbase.nsf/HomeData/homelsx`

9.6.4 Compacting a Notes database using JAVA

Using the Notes Java classes provided in the notes.jar file, you can manipulate Notes objects directly from Java. If your Java program does not use User Interface (UI) methods, it can be run outside the Notes client.

The example shown in Figure 60 on page 160 compacts Notes databases that cannot be normally compacted because they are always open, such as the mail.box and names.nsf.

```
import lotus.notes.*;
public class Compacter implements Runnable
{
  private String fileName;
  public Compacter(String fn) {
    fileName = fn;
    System.out.println("Name of file to be compacted: " + fileName);
    }

  public Compacter() {
    new Compacter("");
  }

  public static void main(String argv[])
    {
        Compacter t = new Compacter(argv[0]);
        NotesThread nt = new NotesThread((Runnable)t);
        nt.start();
}

public void run()
    {
    try
      {
        Session s = Session.newInstance();
        Database db = s.getDatabase("", fileName);
        System.out.println("Database name: \"" + db.getTitle() + "\",
        size before compacting: " + db.getSize());
        db.compact();
        System.out.println("Database name: \"" + db.getTitle() + "\",
        size after compacting: " + db.getSize());
      }
    catch (Exception e)
      {
        e.printStackTrace();
      }
    }
}
```

Figure 60. Sample java code to compact a database

9.6.5 Modifying an ACL of a Notes database using JAVA

Using the Notes Java classes provided in the notes.jar file, you can manipulate Notes objects directly from Java. If your Java program does not use User Interface (UI) methods, it can be run outside the Notes client.

The example shown in Figure 61 on page 161 adds the name “manager” to the specified Notes database. It can be used when there is no client available on a platform to reset the ACL of a database. This program should be restricted to administrators only.

```
import lotus.notes.*;
public class ACLChanger implements Runnable
{
  private String fileName;
  private String newName;
  public ACLChanger(String fn, String nn) {
    fileName = fn;
    newName = nn;
    System.out.println("Name of file to be changed: " + fileName);
    System.out.println("Name to be added to ACL...: " + newName);

    }

  public ACLChanger() {
    new ACLChanger("","");
  }

  public static void main(String argv[])
    {
        ACLChanger t = new ACLChanger(argv[0], argv[1]);
        NotesThread nt = new NotesThread((Runnable)t);
        nt.start();
    }

  public void run()
    {
    try
      {
        Session s = Session.newInstance();
        Database db = s.getDatabase("", fileName);
        System.out.println("Database name: \"" + db.getTitle() );
           // retrieve the database ACL
        ACL acl = db.getACL();
           // and create the new acl entry with manager access
        acl.createACLEntry(newName, 6);
        // get back the entry
        ACLEntry entry = acl.getEntry(newName);
           // and save the ACL back into the database
        acl.save();
        System.out.println("Database name: \"" + db.getTitle() +
           "\", acl changed. " );
      }
    catch (Exception e)
      {
        e.printStackTrace();
      }
    }
}
```

Figure 61. Sample Java code to add a manager to the ACL of a database

Appendix A. Code samples

A.1 Sample shell script to clean up UNIX files

This sample demonstrates how to implement file retention policies within a shell script. It provides file names to look for in the real world. This script is presented as is, as a model to aid customers in developing their own local tools. It is not operational and must be adapted to a specific installation's Domino Notes storage policies and practices. Over time, each installation develops many local procedures that result in additional file names and directories that must be maintained.

It is important to maintain enough free space for Domino to function. The most important files to maintain are the Notes data files. If Domino runs out of space, either it will not be able to deliver the mail or mail routing between servers will fail.

This shell uses many of the standard list file commands and shell programming constructs, plus the following powerful UNIX tools:

- **awk:** A programming language that lets you work with information stored in files
- **grep:** Allows you to search a file for a specified pattern
- **sed:** A non-interactive stream editor

The main logic of the script builds a file of erase file commands based on the current date and how long you want to keep any specific type of file. After the cleanup script completes successfully, the file that was built is actually executed and the selected files are removed.

This type of space management script should be run daily using the cron daemon at a period of low activity.

Refer to *OS/390 V2R7.0 UNIX System Services Command Reference,* SC28-1892, *and OS/390 V2R7.0 UNIX System Services User's Guide*, SC28-1891, for complete descriptions of how to write UNIX shell commands.

```
#!/bin/sh
#--------------------------------------------------------------------#
# cleanup script
#
# This script is run on the S/390 Lotus NOTES nodes via cron.
# It erases files and also maintains file(s) for a number
# of days, and therefore various copies of files for a period
# of time. It is needed to help save DASD space.
#
#--------------------------------------------------------------------#
```

```
#--------------------------------------------------------------------#
# clean_log is where the list of erased files will reside
#--------------------------------------------------------------------#
clean_log="/tmp/cleanup.$(date +%m%d).$(date +%H%M%S)"
echo "# Today is " $(date) > $clean_log
echo "# All entries in this file starting with ## " >> $clean_log
echo "# indicates that that file has been erased." >> $clean_log
echo " " >> $clean_log

#--------------------------------------------------------------------#
# Function keep_file
#
# argument 1 = fully qualified filenames , used with
# ls -l command to find files to keep.
# Should be a '*' somewhere in the actual
# filename, since multiple copies should
# be around.
# argument 2 = integer, number of days to keep these
# files on the system. Those older than
# this number will be erased
#--------------------------------------------------------------------#
function keep_file
{
print " "
print "# $1 files older than $2 days (all listed have been erased)"
print " "
ls -Rdl $1 2>/dev/null | grep '^-rw' | awk ' BEGIN { numdays = 0
                          num_days_to_log = '$2'
                          currjulian = '$(date +%j)'
                          truejulian = currjulian }
  {if ($6 == "Jan")
      numdays = 0 }
  {if ($6 == "Feb")
      numdays = 31 }
  {if ($6 == "Mar")
      numdays = 59 }
  {if ($6 == "Apr")
      numdays = 90 }
  {if ($6 == "May")
      numdays = 120 }
  {if ($6 == "Jun")
      numdays = 151 }
  {if ($6 == "Jul")
      numdays = 181 }
  {if ($6 == "Aug")
      numdays = 212 }
  {if ($6 == "Sep")
      numdays = 243 }
  {if ($6 == "Oct")
      numdays = 273 }
  {if ($6 == "Nov")
      numdays = 304 }
  {if ($6 == "Dec")
      numdays = 334 }

  {if (length($0) != 0 )
      numdays = numdays + $7}
  {if (numdays > currjulian)
      currjulian = currjulian + 365}

  { if ((currjulian - numdays) > num_days_to_log)
    {
```

```
      currjulian = truejulian
#
      print "rm \"" $9 "\" #File info: " $5" "$6" "$7" "$8 }}'
} >> $clean_log


#--------------------------------------------------------------------#
# call function 'keep_file'
#--------------------------------------------------------------------#

keep_file "/it?so???/log.nsf.*" 60
keep_file "/it?so???/log.nsf.*Z" 60

#-------------------------------------

keep_file "/it?so???/*TMP" 7
...
keep_file "/it?so???/noteslog/rc.notes*" 3
keep_file "/it?so???/noteslog/DEBUG/nsd*" 14
keep_file "/it?so???/noteslog/log*" 7


keep_file "/it?so???/*_dir/*/*/*/*.old" 14
keep_file "/it?so???/*_dir/.../*.bak" 3
keep_file "/it?so???/*_dir/.../*.bad*" 3


keep_file "/misc_err*" 7
keep_file "/usr/lpp/adsm/bin/dsmerror.log*" 14
keep_file "/usr/lpp/adsm/bin/dsmsched.log*" 14


keep_file "/u/it??np*/*.nsf" 7
keep_file "/u/it??np*/*.ntf" 7

#--------------------------------------------------------------------#
# list of directories with 'mail' in their name -> put in tmp.out
#--------------------------------------------------------------------#
ls -dl /it?so???/mai* 2>/dev/null | grep dr | awk '{print $9}' > tmp.out

#--------------------------------------------------------------------#
# loop -- reads in name of mail directories then calls keep_file
# to erase all *.old files greater than 14 days old.
#--------------------------------------------------------------------#
while read mail_dir
do

#--------------------------------------------------------------------#
# loop -- this next loop is from a request from NOTES administrators.
# If /it?so???/mail*/*.ft directory exists, this means that
# a 'full text index' of the *.nsf file is being kept.If *.ft
# exists but there is no corresponding *.nsf file, this means
# that the *.nsf file has been migrated to another node and
# the *.ft is not needed (or any of the files in it).
#--------------------------------------------------------------------#
  ls -d $mail_dir/*.ft 2>/dev/null > temp.out

  while read fn
  do
   dotnsf=${fn%.*}.nsf
    if [ ! -f $dotnsf ]
    then
      rm -r $fn
```

```
      print " " >> $clean_log
      print "## ft = full text index " >> $clean_log
print "## Directory $fn and all its files erased!" >> $clean_log
      print "## ft = full text index " >> $clean_log
      print " " >> $clean_log
    fi
  done < temp.out
  rm temp.out

#--------------------------------------------------------------------#
# loop -- this next loop will run keep_file 14 against all files
# in /it?so???/mail* directories that are NOT *.nsf or *.ft
# files.
# Fixed bug - added $ in 2 places in line below. Before this, it
# was filtering out even *.nsf.bak or *.nsf.moved files, whereas
# we really WANT to cleanup these files (but not the .nsf files!)
#--------------------------------------------------------------------#

  ls -d $mail_dir/* | grep -v "\.nsf$" |grep -v "\.NSF$" |grep -v "\.ft"
2>/dev/null > temp.out

  while read fn
  do
    keep_file $fn* 14
  done < temp.out
  rm temp.out

#-------------------------------------------------------------------------#
# The next line will get all the files in the *_dir directories off of
# /it?so??? and check for any files other than .nsf, .NSF, .ft or
# .gif . If any other files exist we will give them 14 days to exist
# and then erase them.
#-------------------------------------------------------------------------#
  ls -d /it?so???/*_dir/* 2>/dev/null | grep -v "\.nsf" |grep -v "\.NSF" |grep
-v "\.ft"|grep -v "\.gif" | awk '{ print "keep_file ", $0 ,"14" }' > temp.out

  while read fn
  do
    keep_file $fn* 14
  done < temp.out
  rm temp.out

#--------------------------------------------------------------------#
# now to clean up the cleanup files! We'll keep 30 days worth
#
# call function 'keep_file'
#--------------------------------------------------------------------#

keep_file "/tmp/cleanup.*" 30

#--------------------------------------------------------------------#
# Keep the ADSM error file(dsmerror.log) from growing. dsmsched.log
# is already being trimmed by ADSM
#--------------------------------------------------------------------#

/usr/bin/tail -500 /dsmerror.log > /tmp/dsmerror.log
cat /tmp/dsmerror.log > /dsmerror.log
rm /tmp/dsmerror.log

#--------------------------------------------------------------------#
# Erase a bunch of files. These are the erases from the original
# cleanup script.
#--------------------------------------------------------------------#
```

```
rm /it?so???/notes.input 1>/dev/null 2>/dev/null
...
rm mbox 1>/dev/null 2>/dev/null



#-------------------------------------------------------------------------#
# We'll execute the $clean_log file now. It erases all the files we found
# that are older than we need to keep around.
#-------------------------------------------------------------------------#
chmod 755 $clean_log
$clean_log


#-------------------------------------------------------------------------#
# Now to change $clean_log so that it can't erase the files again
#-------------------------------------------------------------------------#
sed 's/^rm/##/g' $clean_log >tmp.out
mv tmp.out $clean_log
```

Figure 62. Sample shell script to clean up Domino files

A.2 Sample shell script to monitor HFS space usage

This sample shell script (rdf) returns all of the directories where the percentage of utilization is equal or higher than that of the argument.

```
#!/bin/ksh
#
# This script reports HFS files that have a utilization percentage
# above the number passed as argument 1.
#
#

ReportPercent=$1

if [ $# != 1 ] ; then
        echo "Usage: rdf value"
        echo "\t where value is the %utilization of the disk space.}n"
        exit 1
fi

echo "Host: `uname -a`"
echo "File systems with more than $ReportPercent% utilization\n"
df -P | sed 's/%//'|awk '{if ($5 >= '"$ReportPercent"') printf("%s\t%s\n", $5,$6)}'
```

A.3 Sample shell script to move a database and create database links

The following shell script is included as a sample of what you might build in order to make the conversion from a large, multiple volume HFS to multiple

HFS data sets, then replace the original instance with a database link. This process would run when the corresponding Domino partition is down. It will balance the distribution by space across the new HFS data sets. It is designed for Domino for S/390 R5 in that it creates database link files in EBCDIC. The script must be run under the authority of the user ID for the corresponding Domino server to ensure proper ownership of the new files. Finally, the target HFS data sets must be mounted during the move process with SYNC(0) to enforce immediate I/O and to ensure proper space calculations.

This shell script could be used for the initial move. It can also initiate the periodic movement of new user databases created in /notesdatax/mail out to the directories with the most amount of space available.

Following is a sample shell script, included in this redbook for discussion purposes only. The source and target directory names need to be modified in the script to the correct values for your site, as they are hard-coded.

```
#!/bin/ksh
#
# This script moves mail files from a default mail directory to
# multiple mail directories outside of the Notes data directory.  It
# creates database links in the Notes data directory for each mail
# file moved. Database links are created in EBCDIC format, which is
# supported by Domino R5 only.

# This script must be executed under the authority of the userid
# associated with the Domino server. For the duration of this script's
# execution, target mail directories must be associated with HFS
# files mounted with the option sync(0) to force immediate I/O.
# THE SERVER SHOULD BE DOWN WHEN THIS SCRIPT IS EXECUTED!!!!

# Set for debugging mode
#set -x

# The server's data directory
DATADIR=/notesdata1

# The file directories to place to new mail files
MAILDIRS="/mailhfs1 /mailhfs2 /mailhfs3 /mailhfs4 /mailhfs5"

# Display the initial utilization of target mail file directories
print "Capacity Mounted on"
df -kP ${MAILDIRS} | tail +2 | awk '{print $5" "$6}' | sort -rn

# Scan the default server source mail directory for databases
SCANDIR=${DATADIR}/test

# Use 'find' command instead of 'ls *.nsf' to locate NSF files
# because directory can grow beyond 1024 files which can
# cause errors with shell file expansion
find ${SCANDIR} -name "*.nsf" -print | while read filename; do

# If the file is not a text file, then move it
if file ${filename} | grep -iv text > /dev/null 2>&1; then
```

```
# Find which directory has the most space
  DESTDIR="$( df -kP ${MAILDIRS} | tail +2 | \
    awk '{print $5" "$6}' | sort -rn | \
     awk '{print $2}' | tail -1)"

# Log the file about to be moved
  print "About to move "${filename}" to "${DESTDIR}
  df -kP ${DESTDIR} | tail +2 | awk '{print $5" "$6}'

# Move the file
  mv ${filename} ${DESTDIR}

# Create database link
  FILE_Name="$(basename ${filename})"
  echo "${DESTDIR}/${FILE_Name}" > \
   ${filename}
  chmod 600 ${filename}

# Log the moved file
  print "Moved file "${filename}" to "${DESTDIR}
  df -kP ${DESTDIR} | tail +2 | awk '{print $5" "$6}'

 fi

done

# Display the final utilization of target mail file directories
print "Capacity Mounted on"
df -kP ${MAILDIRS} | tail +2 | awk '{print $5" "$6}' | sort -rn

# end of script
```

A.4 Sample program to modify an ACL

You can use this sample program to create a platform-independent API tool to add an entry to the ACL of a Notes database.

Note: This sample is provided “as is” without warranty, expressed or implied. You should modify, compile and test this program in your own environment and provide additional checks in the code to prevent unauthorized use of this program.

```
/************************************************************/
/* Sample addacl.c                                          */
/*                                                          */
/* This program will add an entry to the ACL of a database  */
/* with the specified access level.                         */
/* This is a sample program without warranty neither expressed */
/* nor implied. The sample can be used as a model for       */
/* implementing an ADDACL function using the notes c API.   */
/************************************************************/
/* C include files */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

/* Notes C API includes */
```

```
/* Notes includes */
#include <global.h>
#include <acl.h>
#include <nsf.h>
#include <log.h>
#include <stdnames.h>
#include <osmem.h>
#include <osmisc.h>
#include <osfile.h>
#include <names.h>
#include <dname.h>
#include <osenv.h>
#include <kfm.h>

/* Notes Platform include */

/* Include for AIX geteuid */
#ifdef AIX
#include <unistd.h>
#endif

/* Include OS/390 Ascii Interface Layer */
#ifdef OS390
#include <unistd.h>
#include <_Ascii_a.h>
#endif


#define ACL_ENTRY_EXISTS     1059

#ifdef UNIX
#define Compare(comp1, comp2) ( strcasecmp(comp1, comp2) )
#else
#define Compare(comp1, comp2) ( strcmpi(comp1, comp2) )
#endif

int CheckServer(char *);

int main (int argc, char *argv[])
     {
     STATUS nError = NOERROR;
     char NotesError[MAXSPRINTF];
     char NotesErrorOut[MAXSPRINTF];
     char vernum[] = "2.2";
     int acl_level = 0, okay = 0, i1 = 0, ctr = 0;
     char NOTES_domain[MAXENVVALUE+1];
     char server_name[MAXUSERNAME+1];
     char datadir[MAXPATH+1];
     WORD acl_type;
     DBHANDLE hDb;
     HANDLE hACL;
     ACL_PRIVILEGES PrivBits = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
     char tempstr[MAXUSERNAME] = "";
     char *ptr;
     char *execDir;
     char onechar[10] = "";
     DN_COMPONENTS DNComp;
     WORD maj_build;
     char DataPath[MAXPATH+1];
     DBHANDLE as_hDir;

#ifdef OS390
     int ascii_trans;
```

```
     struct passwd *p;
     uid_t uid;
#endif



     /* Make sure the correct number of arguements were entered.  */
     if ((argc != 5) && (argc != 2))
        {
        fprintf(stderr,"Invalid number of arguments in command line! ");
        fprintf(stderr,"Program terminating!\n");
        fprintf(stderr,"Use %s -h for proper syntax.\n", argv[0]);


        return 0;
        }

     /* Initialize the Notes API session.   */
     if ((nError = NotesInitExtended (argc, argv)) != NOERROR)
        {
        fprintf(stderr, "\nError initializing Notes!  Error is %d\n", (STATUS) (nError & ERR_MASK));


        return 0;
        }

     OSGetDataDirectory(DataPath);
     if ((nError = NSFDbOpen(DataPath, &as_hDir)) != NOERROR)
      {
      fprintf(stderr, "Could not open %s to determine code version!\n", DataPath);
      NotesTerm();


      return 0;
      }
     NSFDbGetBuildVersion(as_hDir, &maj_build);
     NSFDbClose(as_hDir);

     /* Check to make sure resource files can be resolved */
     OSLoadString(NULLHANDLE, ERR(nError), NotesError, (MAXSPRINTF-1));
     if (strlen(NotesError) == 0)
        {
        fprintf(stderr, "\nERROR: Could not locate string resource files!\n");
#ifdef UNIX
        execDir = getenv("Notes_ExecDirectory");
        if (execDir == NULL)
           fprintf(stderr, "Could not read Notes_ExecDirectory variable from the environment!\n\n");
        else
           fprintf(stderr, "Ensure that %s/res/C is contained in your PATH!\n\n", execDir);
#endif
        NotesTerm();


        return 0;
        }

#if defined(OS390)
     if (maj_build > 145)
        {
        /* Translate argv from EBCDIC to ASCII in Notes 4.6x and higher on EBCDIC systems */
        for (ascii_trans=0; ascii_trans<argc; ascii_trans++)
           {
           __etoa(argv[ascii_trans]);
```

```
          }
        }
#endif

     /* See if user requested version number */
     if (Compare(argv[1], "-v") == 0)
        {
        fprintf(stdout,"\nThis is version %s of %s.\n\n", vernum, argv[0]);
        NotesTerm();

        return 0;
        }

     /* See if user requested help */
     if (Compare(argv[1], "-h") == 0)
        {
        fprintf(stdout,"\nThe syntax for %s is:\n\n\t", argv[0]);
        fprintf(stdout,"%s <directory/file name> <new entry> <access level> <entry type>\n\n", argv[0]);
        NotesTerm();

        return 0;
        }


            return 0;
            }
          if (DNComp.OLength > 0)
            {
            strncpy(NOTES_domain, DNComp.O, DNComp.OLength);
            NOTES_domain[DNComp.OLength]='\0';
            }
          else
            {
            fprintf(stderr, "No domain found from MailServer variable!\n");
            NotesTerm();


            return 0;
            }
          }
        }

       if (strcmp(NOTES_domain, "") == 0)
          {
          fprintf(stderr,"Could not extract domain from the Domain or MailServer environment variables!\n");
           fprintf(stderr, "Program exiting!\n");
           NotesTerm();


          return 0;
          }



        return 0;
        }
     OSGetDataDirectory(datadir);


            return 0;
            }
          if (strstr(argv[1], "_dir") != NULL)
            {
```

```
         fprintf(stderr, "National/Enterprise Server: Cannot modify non-system databases!\n");
         NotesTerm();


         return 0;
         }
      }
   }

if (argc != 5)
   {
   fprintf(stderr,"Invalid arguments in command line!  Program terminating!\n");
   fprintf(stderr,"Use %s -h for proper syntax.\n", argv[0]);
   NotesTerm();


   return 0;
   }

acl_level = atol(argv[3]);
if ((acl_level < 0) || (acl_level > 6))
   {
   fprintf(stderr,"Access Level was out of range!! Enter a number between 0 and 6!!\n");
   fprintf(stderr,"Program terminating!!\n");
   NotesTerm();


   return 0;
   }

if ((Compare(argv[4], "person")      != 0) && (Compare(argv[4], "server")      != 0) &&
    (Compare(argv[4], "unspecified") != 0) && (Compare(argv[4], "mixedgroup")  != 0) &&
    (Compare(argv[4], "persongroup") != 0) && (Compare(argv[4], "servergroup") != 0))
   {
   fprintf(stderr,"Invalid type entry!  Please enter 'unspecified', 'person',\n");
   fprintf(stderr,"'server', 'mixedgroup', 'persongroup' or 'servergroup'\n");
   NotesTerm();


   return 0;
   }

if (Compare(argv[4], "unspecified") == 0)
   acl_type = NULL;
if (Compare(argv[4], "person") == 0)
   acl_type = ACL_FLAG_PERSON;
if (Compare(argv[4], "server") == 0)
   acl_type = ACL_FLAG_SERVER;
if (Compare(argv[4], "mixedgroup") == 0)
   acl_type = ACL_FLAG_GROUP;
if (Compare(argv[4], "persongroup") == 0)
   acl_type = (ACL_FLAG_GROUP + ACL_FLAG_PERSON);
if (Compare(argv[4], "servergroup") == 0)
   acl_type = (ACL_FLAG_GROUP + ACL_FLAG_SERVER);



   return 0;
   }

/* Converts name to canonicalized version if '/' is found in name. */
if ((ptr = strstr(argv[2], "/")) != NULL)
   {
```

```
   if ((nError = DNCanonicalize(0L, NULL, argv[2], tempstr, MAXUSERNAME, NULL)) != NOERROR)
      {
      fprintf(stderr,"Could not convert name to cannonical format internally!!\n");
      NotesTerm();


      return 0;
      }
   }
else
   {
   strcpy(tempstr, argv[2]);
   }

/* Open the database to get the database handle. */
if ((nError = NSFDbOpen (argv[1], &hDb)) != NOERROR)
   {
   fprintf(stderr,"Error opening file: %s   Program terminating!!\n", argv[1]);
   fprintf(stderr,"Please check to ensure that the directory / file are correct!\n");
   goto termnotes;
   }

/* Read the ACL of the current database.  */
if ((nError = NSFDbReadACL(hDb, &hACL)) != NOERROR)
   {
   fprintf(stderr,"Error reading database ACL: %s   Program terminating!!\n", argv[1]);
   NSFDbClose(hDb);
   goto termnotes;
   }

LogAddText(LOG_EVENT_ENTRY, LOG_AUTO_ROLLOVER | LOG_ADD_TIMESTAMP,
           LOG_ITEM_EVENTS, NULLHANDLE, 0,
           "Notes API %s - Adding %s to %s.", argv[0], argv[2], argv[1]);

/* Add the ACL entry with the specified level. */
if ((nError = ACLAddEntry(hACL, tempstr, acl_level, &PrivBits, acl_type)) != NOERROR)
   {
   if (nError == ACL_ENTRY_EXISTS)
      {
      fprintf(stderr,"ACL entry already exists for %s in %s!!\n", tempstr, argv[1]);
      OSMemFree(hACL);
      NSFDbClose(hDb);
      goto termnotes;
      }
   else
      {
      fprintf(stderr,"Error adding ACL entry for %s in %s!!\n", tempstr, argv[1]);
      OSMemFree(hACL);
      NSFDbClose(hDb);
      goto termnotes;
      }
    }
   else
    {
    okay = 1;
    NSFDbStoreACL(hDb, hACL, 0L, 0);
    NSFDbClose(hDb);
    OSMemFree(hACL);
    }

/* Terminate Notes API session.  */
termnotes:
```

```
if (okay == 1)
   printf("Added entry for %s into file %s!!\n", tempstr, argv[1]);

#if defined(NT_VERSION)
return 0;
#else
LAPI_RETURN (NOERROR);
#endif
}
```

Figure 63. Sample C API program to add an entry to the ACL of a database

A.5 Sample agent to set a default ACL on mail files

The following LotusScript agent is included as a sample of what you can build in order to enforce default entries in the ACL of mail databases. Put the LotusScript into a MAILACL.NSF database and run as a scheduled agent once a week. It processes all databases in mail and mailxx directories and sets default entries */SRV/org as Manager and */Management/QA/Org as Manager.

Usage Notes:

- Change **entryString$** to the generic name of your servers or to your server groupname.
- Change **entryString1$** to the generic name of your administration group or to your administration groupname.
- Change the name of the database in the agentlog.OpenNotesLog call from MAILACL.NSF to the name and location of the database in which the agent resides.
- If you do not want to assign a default ACL, comment out the section that sets a **-Default-** ACL.
- If you do not want to assign a default ACL for Web access, comment out the section that sets **Anonymous** access.
- If you want to change the access level for default or anonymous access from noaccess to a different level, update the access level accordingly in the -Default- and Anonymous sections.

```
'Update ACL entry to Include Management ID and */SRV entries:

Option Public
Option Explicit

Sub Initialize

    Dim session As New NotesSession
    Dim directory As NotesDbDirectory
    Dim curdb As NotesDatabase, db As NotesDatabase
    Dim acl As NotesACL
    Dim entry As NotesACLEntry
    Dim server As String, entryString As String, entryString1 As String, leveltxt As String
    Dim level As Integer
    Dim x As Variant
    Dim agentLog As New NotesLog("Mail File ACL Agent")
```

```
    On Error Resume Next

    entryString$ = "*/SRV/org"
    entryString1$ = "*/Management/QA/org"

 'Set the log file destination
    Call agentLog.OpenNotesLog("", "MAILACL.NSF")
    Call agentLog.LogAction("*** Mail File Agent Invoked ***")
    agentLog.LogActions = True

 ' get current db to obtain server name
    Set curdb = session.CurrentDatabase
    server$ = curdb.Server

 ' iterate through all dbs
    Set directory = New NotesDbDirectory( server$ )
    Set db = directory.GetFirstDatabase( DATABASE )

    Do While Not (db Is Nothing)
 ' open mail only dbs (directories that start with mail including mail1..999)
        x = Instr(1, db.FilePath, "mail", 4)
        If  x <> 0 Then

            Call db.open("","")
            Set acl = db.acl
            level = db.CurrentAccessLevel

            If level <> ACLLEVEL_MANAGER Then

                Select Case db.CurrentAccessLevel
                Case ACLLEVEL_NOACCESS : leveltxt = "No Access"
                Case ACLLEVEL_DEPOSITOR : leveltxt = "Depositor"
                Case ACLLEVEL_READER : leveltxt = "Reader"
                Case ACLLEVEL_AUTHOR : leveltxt = "Author"
                Case ACLLEVEL_EDITOR : leveltxt = "Editor"
                Case ACLLEVEL_DESIGNER : leveltxt = "Designer"
                End Select

                Call agentLog.LogAction(leveltxt + " access to " + db.FilePath + ", skipping database..")

            Else

                Call agentLog.LogAction("Scanning database.. " + db.FilePath + " with Manager Access")
'check for 1st acl entry
                Set entry = acl.GetEntry ( entryString$)
' if not found create and save
                If entry Is Nothing Then
                    Set entry = acl.CreateACLEntry ( entryString$ , ACLLEVEL_MANAGER )
                    entry.CanDeleteDocuments = True
                    Call acl.Save
                    Call agentLog.LogAction("Added  " + entryString$ + "  to " + db.FilePath)
                End If

'check for 2nd acl entry
                Set entry = acl.GetEntry ( entryString1$)
' if not found create and save
                If entry Is Nothing Then
                    Set entry = acl.CreateACLEntry ( entryString1$ , ACLLEVEL_MANAGER )
                    entry.CanDeleteDocuments = True
                    Call acl.Save
                    Call agentLog.LogAction("Added " + entryString1$ + " to " + db.FilePath)
                End If
'check for default entry
                Set entry = acl.Getentry("-Default-")
' if not found create and save
                If entry Is Nothing Then
                    Set entry = acl.CreateACLEntry ( "-Default-" , ACLLEVEL_NOACCESS )
                    Call acl.Save
                    Call agentLog.LogAction("Added " + "-Default-" + " to " + db.FilePath)
                End If
'check for Anonymous entry
                Set entry = acl.Getentry("Anonymous")
' if not found create and save
                If entry Is Nothing Then
                    Set entry = acl.CreateACLEntry ( "Anonymous" , ACLLEVEL_NOACCESS )
                    Call acl.Save
```

```
                    Call agentLog.LogAction("Added " + "Anonymous" + " to " + db.FilePath)
                End If
            End If
        End If
' next database in list
        Set db = directory.GetNextDatabase
    Loop
    Call agentLog.LogAction("*** End of Mail File Agent Run ***")
    Call agentLog.Close

End Sub
```

A.6 Trending worksheet

Table 9 can be used as a worksheet for trending your workload.

Table 9. Worksheet for trending your workload

Service type or usage	Notes Mail	Notes Application	HTTP	HTTP POP3	HTTP IMAP
Number of users		X	X	X	X
Percent active		X	X	X	X
Number of active users					
Transaction rate					
Transaction response time					
CPU					
DASD space					
I/O activity					
Memory					
Network bandwidth					

A.7 Setting up an event monitor

The following figures demonstrate how to set up an event monitor in the Statistics and Events database using the Monitor Setup Wizard. Figure 64 is the first screen of seven in this process.

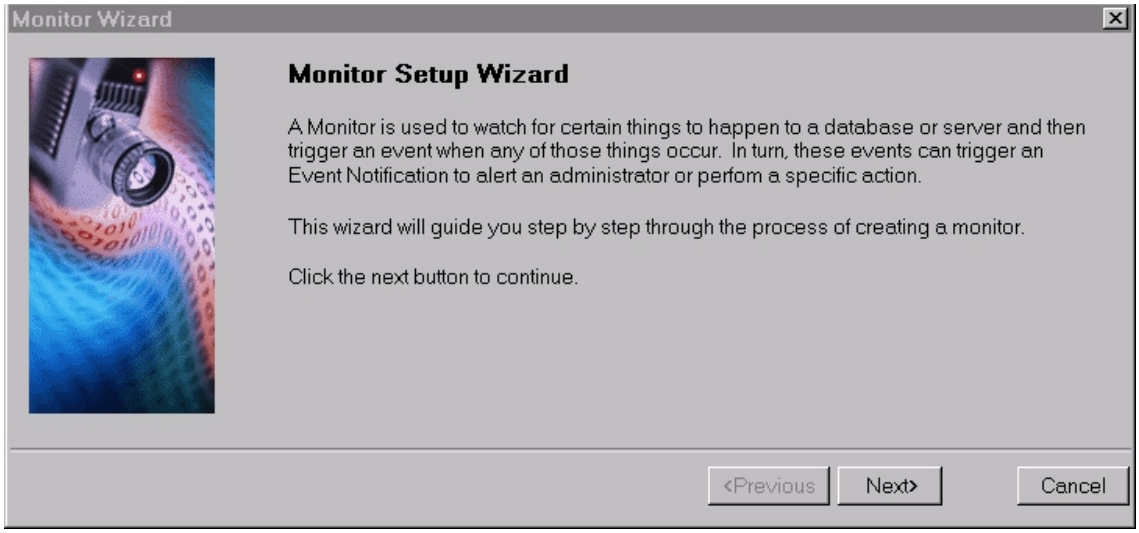


Figure 64. Setting up a monitor

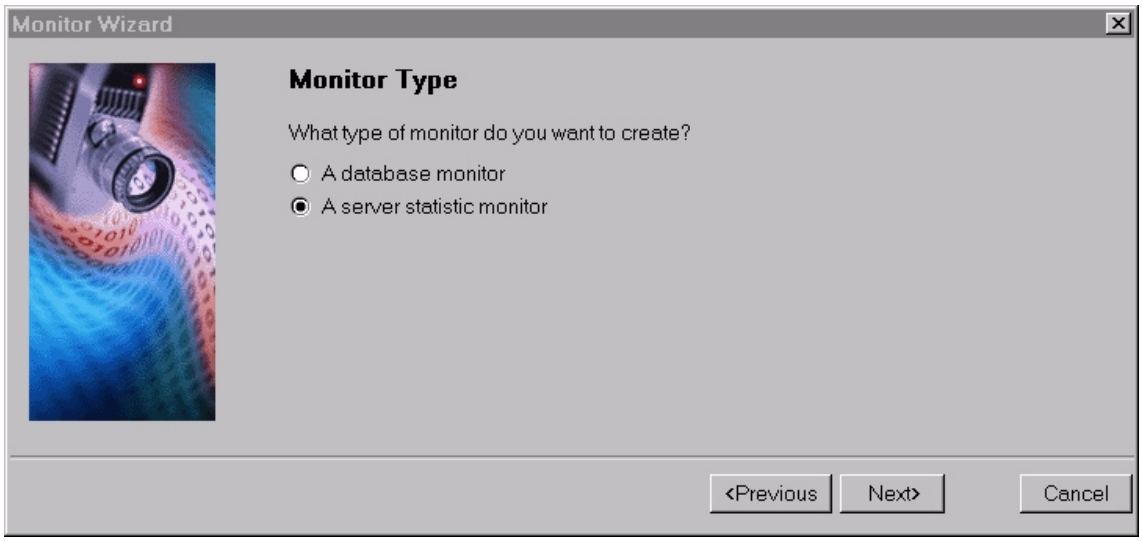


Figure 65. Setting up a monitor: Monitor type

Select the type of event that you want to monitor: a database monitor, or a server statistic monitor, as shown in Figure 65.

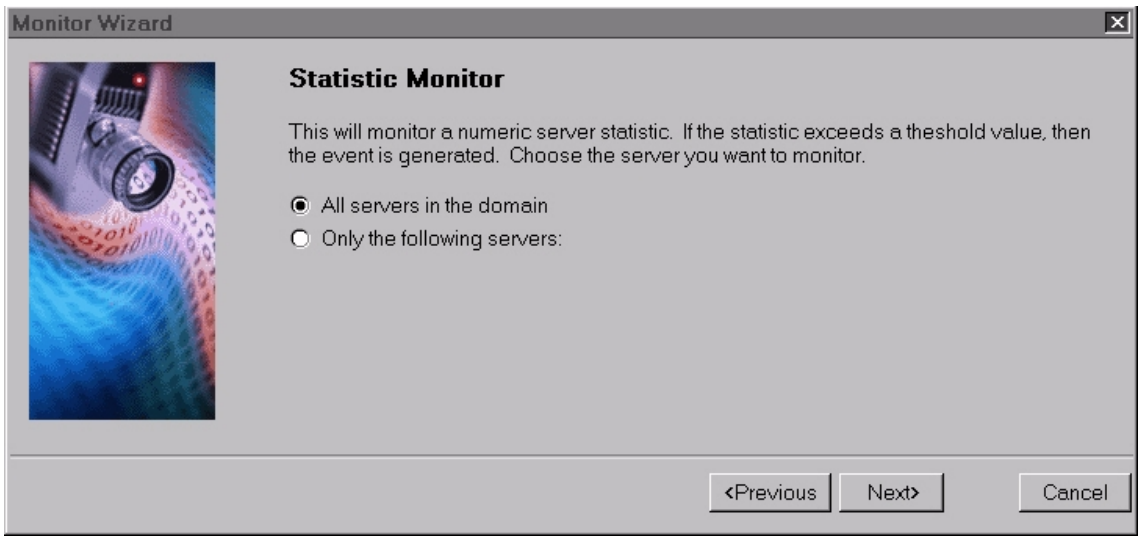


Figure 66. Setting up a monitor: Target server selection

For a server statistic monitor, select a specific server or all servers in your domain, as shown in Figure 66.

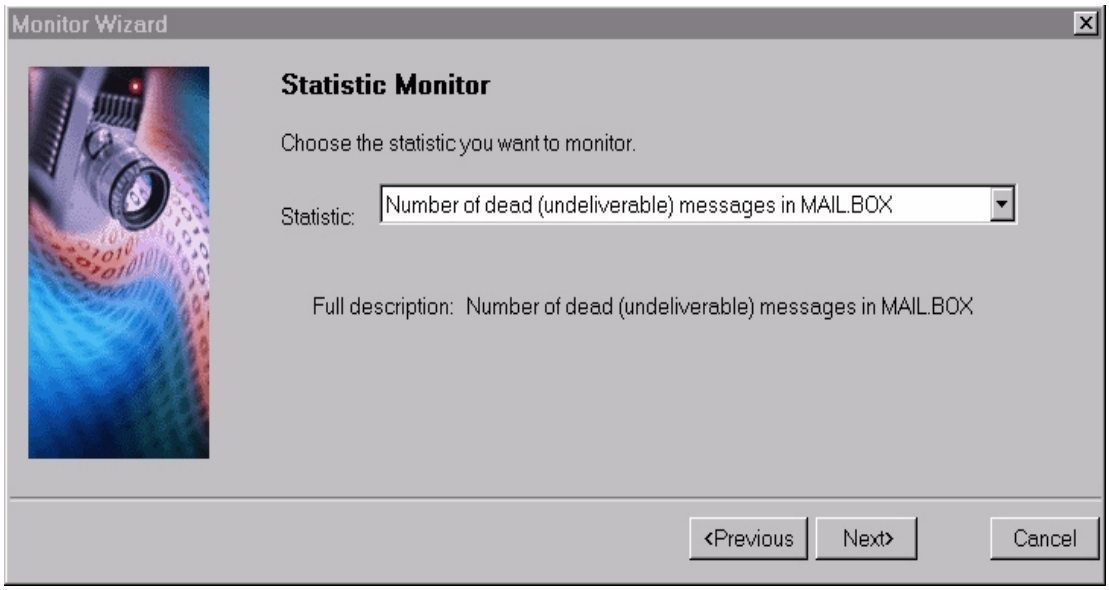


Figure 67. Setting up a monitor: Statistic selection

Select a statistic from the list of statistics available. In the message box, shown in Figure 67, you only see the description of the server statistic.

Table 10 lists all mail-related server statistics in Domino for S/390 V5R5.04a.

Table 10. Mail-related server statistics

Name	Description
Mail.AverageDeliverTime	Average time for mail delivery
Mail.AverageServerHops	Average number of server hops for mail delivery
Mail.AverageSizeDelivered	Average size of mail delivered
Mail.Dead	Number of dead (undeliverable) messages in MAIL.BOX
Mail.Delivered	Number of mail messages moved into MAIL.BOX by router
Mail.Deliveries	Count of actual mail items delivered (may be different from delivered which counts individual messages)
Mail.Domain	Mail domain name
Mail.Hold	Number of mail messages in message queue on hold
Mail.IMAP.Cache.MsgInsertAttempts	Number of IMAP server attempts to insert message in cache
Mail.IMAP.FindCacheAttempted	Number of attempts by IMAP server to find message in cache
Mail.IMAP.NumCacheHits	Number of messages IMAP server found in cache
Mail.IMAP.NumMsgInsertedInCache	Number of messages IMAP server inserted into Cache
Mail.MaximumDeliverTime	Maximum time for mail delivery
Mail.MaximumServerHops	Maximum number of server hops for mail delivery
Mail.MaximumSizeDelivered	Maximum size of mail delivered

Name	Description
Mail.MinimumDeliverTime	Minimum time for mail delivery
Mail.MinimumServerHops	Minimum number of server hops for mail delivery
Mail.MinimumSizeDelivered	Minimum size of mail delivered
Mail.PeakByteTransferRate	Peak transfer rate
Mail.PeakMessagesTransferred	Peak number of messages transferred
Mail.PeakMessageTranferRate	Peak message transfer rate
Mail.PeakMessageTranferRate.Time	Time of peak message transfer rate
Mail.PeakTotalBytesTransfered	Highest number of bytes transferred
MAIL.TotalFailures	Total number of mail failures
Mail.TotalKBTransferred	Total mail transferred in KBytes
Mail.TotalPending	Total mail messages pending
Mail.TotalRouted	Number of mail messages moved from MAIL.BOX to other servers for delivery
Mail.TotalRouted.SMTP	Number of mail messages moved from MAIL.BOX via SMTP
Mail.TotalRouter.NRPC	Number of mail messages moved from MAIL.BOX via NRPC
Mail.TransferFailures	Number of mail messages router was unable to transfer
Mail.Transferred	Number of mail messages router attempted to transfer
Mail.Waiting	Number of outgoing mail messages currently in MAIL.BOX waiting for transfer
Mail.WaitingforDNS	Number of mail messages in MAIL.BOX waiting for DNS

Name	Description
Mail.WaitingRecipients	Number of pending mail messages in MAIL.BOX awaiting delivery into user's mail files

The mail-related server statistics are a subset of the server statistics available. Look in your Statistics and Events database for more detail.

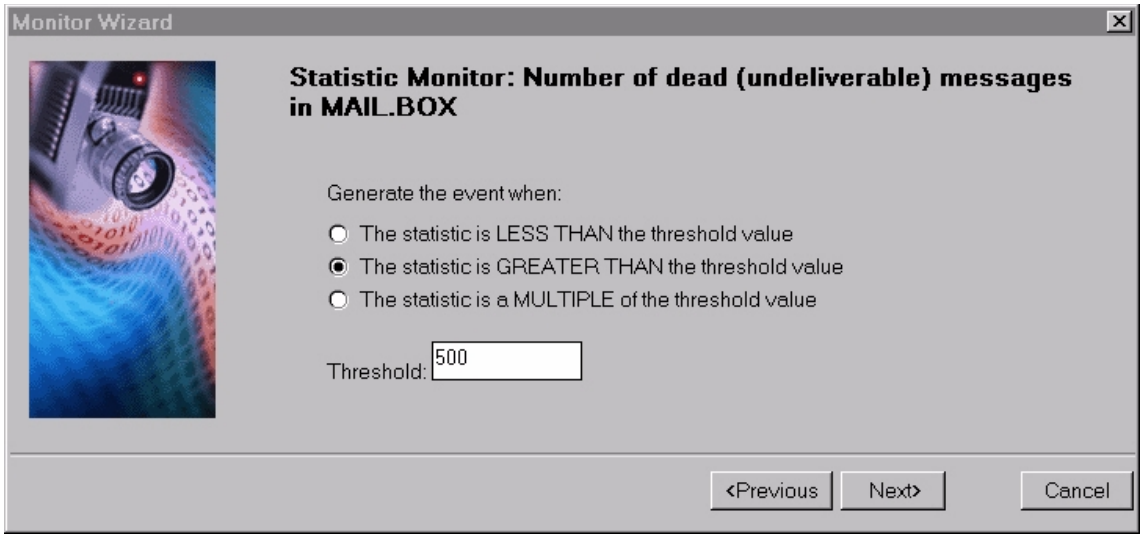


Figure 68. Setting up a monitor: Threshold definition

Define the number and condition for the threshold value of this event, as shown in Figure 68.

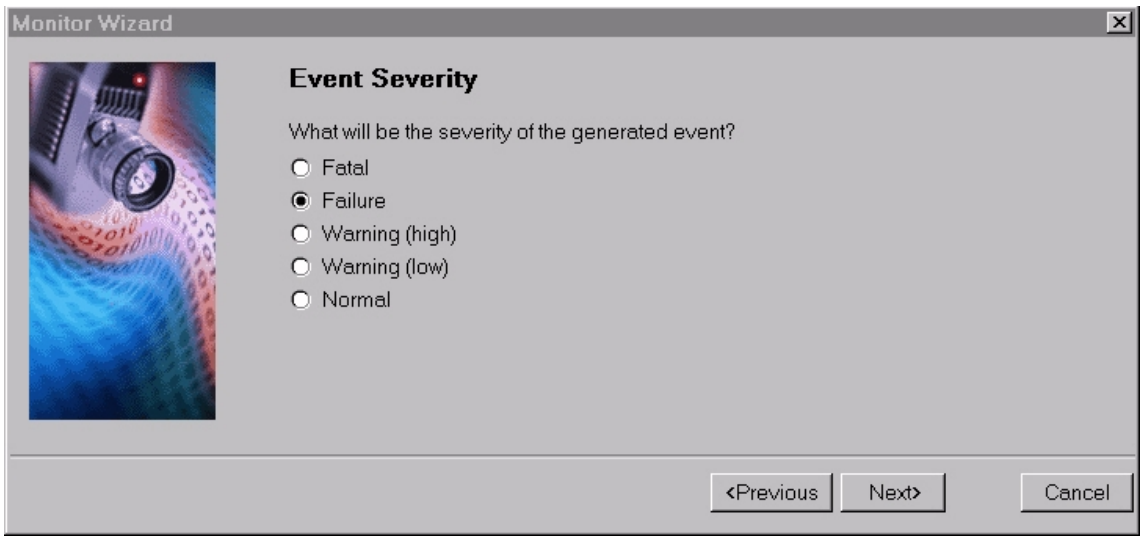


Figure 69. Setting up a monitor: Event severity

Define the severity of an event, as shown in Figure 69.

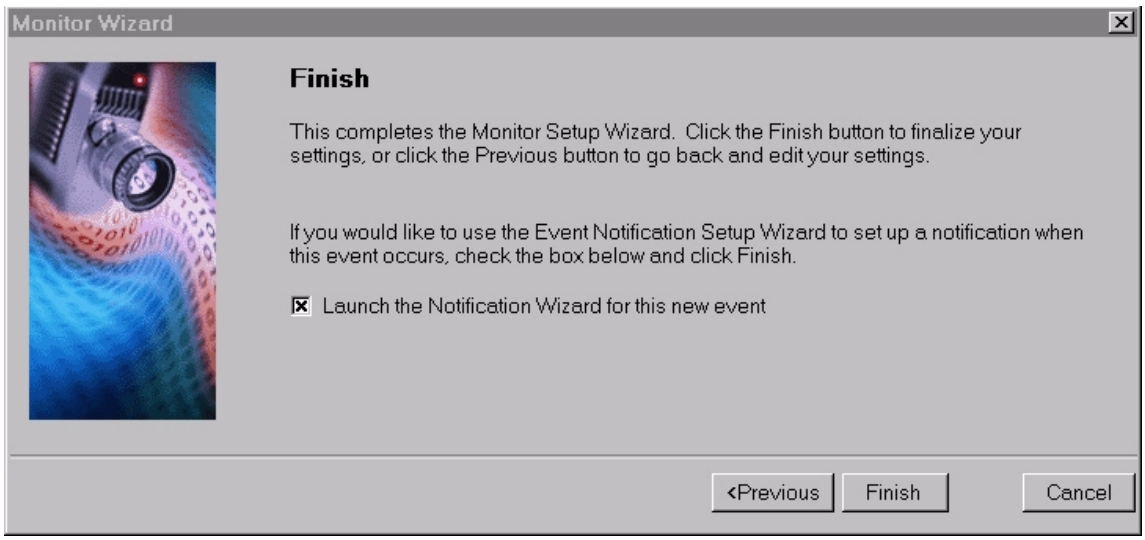


Figure 70. Setting up a monitor: Finish and launch Notification Wizard

As shown in Figure 70, select the box to launch the Notification Wizard, and click **Finish**.

A.8 Setting up an event notification

The following figures step you through the process of setting up Event Notification using the Setup Wizard, beginning with Figure 71.

Figure 71. Set up event notification

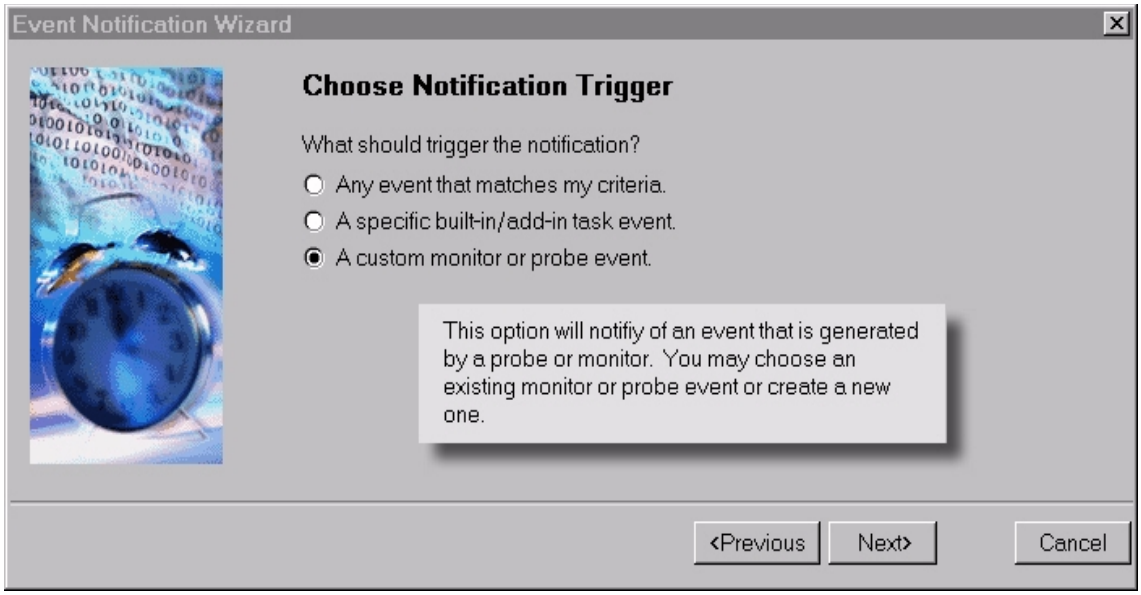


Figure 72. Choose notification trigger

From the panel shown in Figure 72, select a notification trigger.

Figure 73. Select the event

The panel shown in Figure 73 allows you to select the event.

Figure 74. Selecting the notification method

The panel shown in Figure 74 allows you to select the method for generating the event notification.

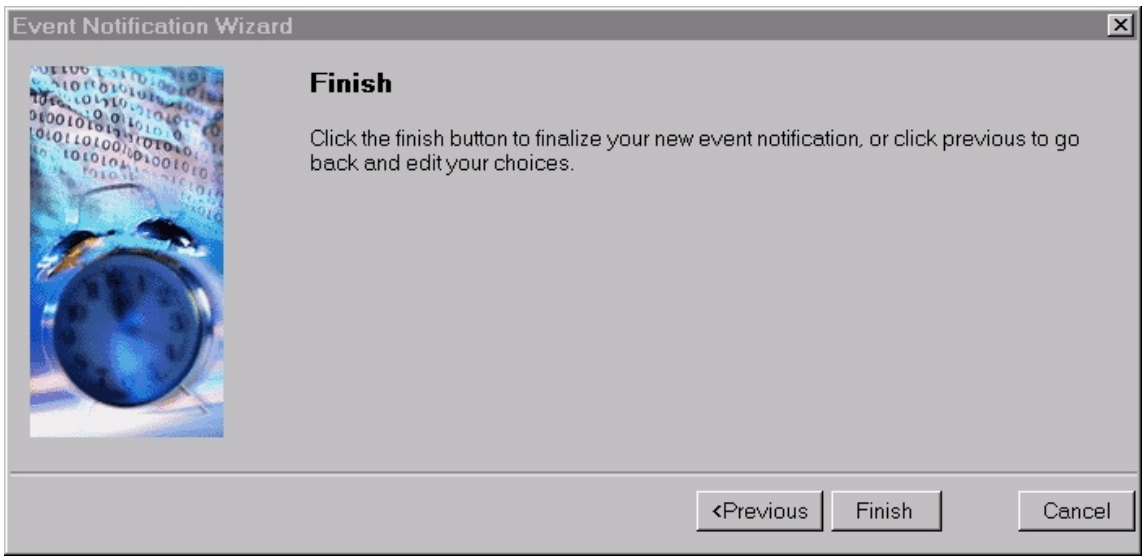


Figure 75. Finish and save event notification

The panel shown in Figure 75 is the final step in the process of setting up an event notification.

Appendix B. Production Domino servers in Poughkeepsie

Many people have shown an interest in the production Domino servers on S/390 that IBM has been running in Poughkeepsie, New York, USA, since November 1996. These are true production Domino servers, used for IBM's internal rollout of Lotus Notes. They have also been used as a production proving ground for Domino for S/390 since 1996. Currently they are also used to pilot new Domino releases on S/390 before they become generally available. We provide this information as of November 2000, to show you an example of a large Domino for S/390 system.

B.1 Hardware environment

The Poughkeepsie production system is currently supporting over 17,000 registered users with two S/390 server machines: one IBM 9672-ZZ7 and one IBM 9672-YX6. The IBM 9672-ZZ7 has four logical partitions (LPARs) each with two Domino partitions (DPARs). This is shown in Figure 76 on page 188.

The IBM 9672-ZZ7 supports almost 10,000 registered users.

One of the LPARs on the IBM 9672-ZZ7 is used as a production test environment. It is the first to run with a new release of Domino or any of the prerequisite software. One of the DPARs in this LPAR was the first to exceed 2,000 registered users. Increasing the number of registered users per LPAR has allowed us to significantly reduce the number of DPARs that we need to administer as well as reduce the amount of CPU used.

Three of the LPARs have 2 GB of central storage and 4 GB of expanded storage and the fourth one has 2 GB of central storage and 2 GB of expanded storage. Each of the 4 LPARs have 6 logical CPs and a weight of 25.

A total of 3,000 GB of DASD has been allocated to the four LPARs. Although each registered user is allocated 100 MB of DASD space, some are currently using over 800 MB of DASD space. We started out with 25 mail files per HFS data set and then moved some of the larger mail files to other HFS data sets.

The HFS data sets are backed up nightly using ADSM. This takes about 10 hours per night running multiple ADSM clients, each backing up about 20 HFS data sets. Nineteen ADSM clients are used to backup the HFS data sets on the 9672-ZZ7.

Transaction logging was initially implemented as a circular log to give us better data integrity and faster restart time. We also saw a reduction in CPU

utilization and some improvement in response time. We are currently implementing archived transaction logging.

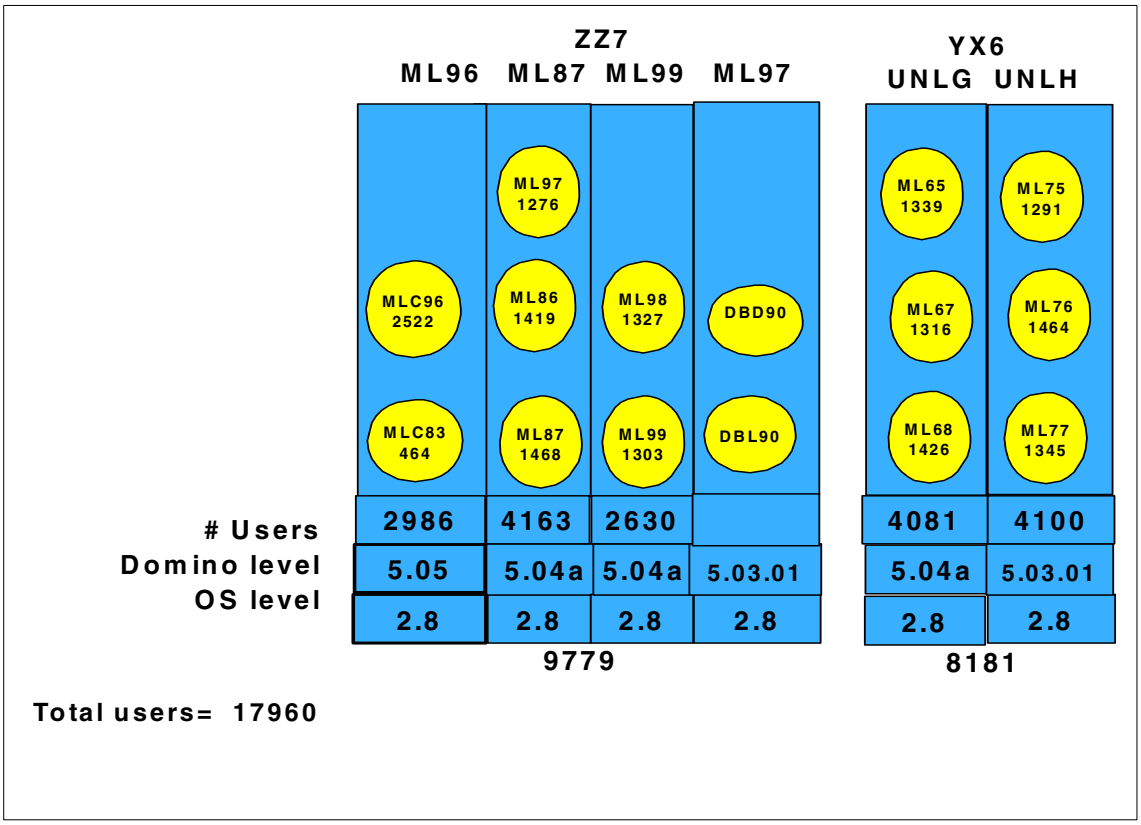


Figure 76. Poughkeepsie production configuration

B.2 Workload

The S/390 server currently supports over 17,000 registered users doing mail, calendar, and scheduling. A typical day's workload consists of:

- About 45 percent of the registered users are connected during the peak hour.
- More than 1,000 server transactions per registered user are run per day.
- More than 100 server transactions per registered user per hour are run during the nine prime-time hours.

Appendix C. Special notices

This publication is intended to help Domino adminstrators and OS/390 system programmers who are maintaining a large Domino Domain on S/390 systems. The information in this publication is not intended as the specification of any programming interfaces that are provided by Domino for S/390 V5R5.04a. See the PUBLICATIONS section of the IBM Programming Announcement for Domino for S/390 V5R5.04a for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers

attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

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

IBM ®	RACF
AIX	RAMAC
AS/400	Redbooks
AT	Redbooks Logo 
CT	RMF
Current	RS/6000
DB2	S/390
DFSMS/MVS	SP
DFSMSdss	System/390
DFSMShsm	VisualAge
IBM	VM/ESA
Language Environment	VTAM
MQSeries	WebSphere
MVS/ESA	Wizard
Netfinity	XT
Nways	400
OS/2	Lotus
OS/390	Lotus Notes
OS/400	SmartSuite
Parallel Sysplex	Domino
Processor Resource/Systems Manager	Notes

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

countries.

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

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.

Appendix D. Using the additional material

This redbook also contains additional material on the Web. See the appropriate sections below for instructions on using or downloading the material.

D.1 Locating the additional material on the Internet

The Web material associated with this redbook is also available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG245684

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number.

D.2 Using the Web material

The additional Web material that accompanies this redbook includes the following:

File name	*Description*
SG245984.zip	Zipped Code Samples

This zip file contains the following files:

File name	*Description*
SG245984.zip	Zipped Code Samples
ACLChanger.java	Source of the java sample that modifies the ACL of database
ActivityTrends.pdf	Description of the Activity Trends Tool
addacl.c	Source ot the C program to add an entry to the ACL of a database
Buckys_Papers	Subdirectory containing Bucky Pope's research papers on notes performance
cleanup.sh	UNIX shell script to clean up old files
Compacter.java	Source of the java sample that compacts a notes database
movemail.sh	UNIX shell script to move a mail file

rdf.sh	UNIX shell script to show OS/390 HFS files with more than xx percent utilization
Trendsv38a.ntf	Notesdatabase Template of the Activity Trends Tool
index.html	HTML page of the contents of this CD-ROM

D.2.1 System requirements for downloading the Web material

The following system configuration is recommended for downloading the additional Web material.

Hard disk space:10 MB
Operating System:Windows and OS/390

D.2.2 How to use the Web material

Create a subdirectory (folder) on your workstation and copy the contents of the Web material into this folder. Unzip the file. Each section has a separate readme file which tells about the section.

You can access the contents of the material by pointing your Web browser at the file *index.html* in the folder and following the links found there.

Appendix E. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 199.

- *Lotus Domino for S/390 Release 5: Performance Tuning & Capacity Planning*, SG24-5149
- *Lotus Domino for S/390 Release 5: Problem Determination Guide*, SG24-5599
- *Lotus Domino for S/390 Release 5: Installation, Customization and Administration,* SG24-2083
- *Debugging UNIX System Services, Lotus Domino, Novell Network Services and other Applications on OS/390*, SG24-5613
- *Open Systems Adapter 2 Implementation Guide,* SG24-4770

E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

E.3 Other resources

These publications are also relevant as further information sources:

- *Configuring TCP/IP and VIPA in OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing*, SG24-5227
- *Hierarchical File System Usage Guide*, SG24-5482
- John Lamb. *Lotus Notes and Domino 5 Scalable Network Design: Web Server Network Infrastructure.* McGraw-Hill, 1999. (ISBN 007913792X, IBM publication number SR23-9218))
- *OS/390 UNIX System Services User's Guide*, SC28-1891
- *OS/390 V2R7.0 UNIX System Services Command Reference,* SC28-1892
- *OS/390 V2R7.0 UNIX System Services User's Guide*, SC28-1891
- *S/390 RMF Report Analysis*, SC28-1950
- *S/390 Processor Resource/Systems Manager Planning Guide*, GA22-7236
- *S/390 OSA-Express Gigabit Ethernet Implementation Guide,* SG24-5443
- *Managing Domino Databases*, Lotus manual CT73MNA.
- *Lotus Domino R5 -- Administering a Domino Server,* Lotus manuals CT7VHNA and CT7VINA.

E.4 Referenced Web sites

Web sites are also relevant as further information sources. These are home pages for organizations mentioned in this redbook:

- Lotus Domino and Notes home page. Contains Iris Today Technical Articles, Documentation, Sample applications in the Iris Sandbox, and more.
  `http://www.notes.net`
- Lotus Development Corporation home page
  `http://www.lotus.com`
- Lotus Domino for S/390 home page
  `http://www.s390.ibm.com/domino/`
- Tivoli home page
  `http://www.tivoli.com`
- Candle homepage
  `http://www.candle.com`
- Candle IntelliWatch home page
  `http://www.intelliwatch.com`

- Notes administration tools
  http://www.ivesco.com
- Notes administration tools
  http://www.percussion.com
- Notes administration tools
  http://www.4nf.co.uk
- BMC Software home page
  http://www.bmc.com
- Tiger Technologies home page
  http://www.tigerti.com

These are web sites with specific helpful information. Here is a Candle page:

- Candle index of Notes management tools
  http://www.candle.com/productservices/appmgmt/lotusnotesmgmt/index.html

Here are web sites with IBM information:

- Domino for S/390 tools download page
  http://www.s390.ibm.com/products/domino/domdown.html
- Domino for S/390 performance overview
  http://www.s390.ibm.com/products/domino/domperf.html
- Domino for S/390 performance utilizing the IBM Storage Server
  http://www.s390.ibm.com/marketing/gf225155.html
- Enterprise integration with Domino for S/390: Moving beyond E-mail
  http://www.s390.ibm.com/marketing/gf225141.html
- High availability options for Lotus Domino on S/390
  http://www.s390.ibm.com/marketing/gf225167.html
- Lotus Domino on S/390 service
  http://www.s390.ibm.com/products/domino/servchoice.html
- Lotus Domino on S/390 technical information links
  http://www.s390.ibm.com/products/domino/domup.html
- OS/390 UNIX home page
  http://www.s390.ibm.com/unix/
- RMF manuals in PDF format
  http://www.s390.ibm.com/os390/bkserv/r10pdf/rmf.html
- S/390 unique functions added in Domino R5.03
  http://www.s390.ibm.com/marketing/gf225166.html

- S/390 white papers and position papers
  http://www.s390.ibm.com/marketing/position.html
- S/390 Service Update Facility home page
  http://www.s390.ibm.com/suf
- Domino for S/390 incremental installation home page
  http://www.s390.ibm.com/products/domino/iinst.html
- PTF list for Domino for S/390
  http:// www.s390.ibm.com/products/domino/domptf.html
- IBM S/390 support
  http://techsupport.services.ibm.com/support/s390

Here are some Lotus web pages:

- To order *Lotus Domino R5 -- Administering a Domino Server*, use Lotus part numbers CT7VHNA and CT7VINA
  http://www.lotus.com
- Fixes for Domino Release 5
  http://www.notes.net/r5fixlist.nsf
- To search the Lotus Notes Knowledge Base, especially technotes 180942, 181533 and 181560
  http://support.lotus.com/lshome.nsf
- Notes C API Toolkit
  http://www.lotus.com/developers/devbase.nsf/HomeData/homecapi
- Notes C++ API Toolkit
  http://www.lotus.com/developers/devbase.nsf/HomeData/homecpp
- Notes LSX Toolkit
  http://www.lotus.com/developers/devbase.nsf/HomeData/homelsx

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

ACL Access control list. A list of database users (individual users, Lotus Domino servers, and groups of users and/or servers) created and updated by the database manager. The ACL specifies which users can access the database and what tasks they can perform.

Address space An OS/390 construct used to represent a batch job, TSO/E user, started task, UNIX Systems Services user, or forked USS process. An address space control block (ASCB) points to many other control blocks used to manage the virtual storage, I/O resources and programs used by the work unit. A USS process creates an address space.

ADSM ADSTAR Distributed Storage Manager. IBM product for distributed file backup, restore, archive and more. The products have been rebranded to Tivoli Data Manager (TDM).

APAR Authorized Program Analysis Report. IBM record of a software problem.

Canonical Format Format for storing hierarchical names that displays the hierarchical attribute of each component of the name. For example, the canonical format for the name Reuben Smith/Ottawa/Acme/CA is: CN=Reuben Smith/OU=Ottawa/O=Acme/C=CA where: CN is the common name, OU is the organizational unit, O is the organization, and C is the country code.

CMG Computer Measurement Group. They hold conferences on technology management and performance evaluation.

DADSM Direct Access Device Space Management. Routines of the OS/390 operating system that manage disk storage.

DAE Dump Analysis and Elimination.

DECS Domino Enterprise Connection Services. A forms-based interface that allows integration to external data from Domino applications.

DFSMS Data Facility Storage Management Subsystem. A software suite that automatically manages data from creation to expiration. DFSMS/MVS provides allocation control for availability and performance, backup/recovery and disaster recovery services, space management, tape management, and reporting and simulation for performance and configuration tuning.

DNN Domino Named Network

DNS Domain Name System. A function to associate names and addresses on Internet domain servers.

Domain A Domino domain is a collection of Domino servers and users that share a common Domino directory. The primary function is mail routing. Users' domains are determined by the location of their server-based mail files.

GAN Global area network.

GRS Global Resource Serialization. An OS/390 base component that manages serialization.

HTTP Hypertext Transmission Protocol. An Internet protocol used to transfer files from one computer to another.

ICM Internet Cluster Manager. Domino component for failover and load balancing of HTTP clustering.

IMAP Internet Message Access Protocol. A mail protocol that allows clients running it to retrieve mailfrom a host mail server also running the protocol.IMAP is similar to POP3 but has additional features.For example, it supports three modes of mailboxaccess. You can enable IMAP on a Domino server.

I/O Input/output.

IPCS Interactive Problem Control System. An OS/390 tool for examining operating system and subsystem dumps. The OS/390 flavor of IPCS is written in all caps.

ipcs Inter-Process Communication System. A UNIX programming model to coordinate and serialize separate threads. The UNIX ipcs is usually written in lower case characters.

IPL Initial program load. The initialization procedure that causes an operating system to start operation.

IRIS A wholly owned division of Lotus. They developed the Domino/Notes product.

ITSO International Technical Support Organization.

JES2 Job Entry Subsystem/2. A component of OS/390 that manages the initiation and termination of batch jobs, TSO users and started tasks as well as printed output.

LAN Local area network.

LDAP Lightweight Directory Access Protocol. A set of protocols for accessing information directories. LDAP is based on the X.500 protocol, but supports TCP/IP, which is necessary for Internet access. Because it's a simpler version of X.500, LDAP is sometimes called X.500-lite. You can enable LDAP on a Domino server to allow LDAP clients to access information in Domino directory, for example, e-mail addresses.

Logical partition In LPAR mode, a subset of the processor unit resources that is defined to support the operation of a system control program (SCP).

Logically partitioned mode (LPAR) A mode that allows the operator to allocate hardware resources of the processor unit among several logical partitions.

Lotus A wholly-owned subsidiary of IBM, and the owner of the Domino/Notes product.

LotusScript A version of BASIC that offers not onlystandard capabilities of structured programming languages, but a powerful set of language extensions that enable object-oriented development within and across products. Its interface to Notes is through predefined object classes.

LPAR Logical Partition. System/390 capability to divide up the processors, memory and channels so that several operating system images can run independently on the same machine.

LS:DO The ODBCConnection, ODBCQuery, and ODBCResultSet classes, collectively called the LotusScript Data

Object (LS:DO), provide properties and methods for accessing and updating tables in external databases through the ODBC (Open Database Connectivity) Version 2.0 standard.

MIB Management Information Base. Defines the scope of what can be managed by SNMP.

MIME Multipurpose Internet Mail Extensions. Software that allows you to attach non-text files to Internet mail messages. Non-text files include graphics, spreadsheets, formatted word-processor documents, and sound files.

MTA Mail transfer agent. A message transfer agent, also called a gateway, is a program that translates messages between mail formats.

NAB Notes Name and Address Book. Now called the Domino directory or Personal Address Book.

NNN Notes Named Network. Now DNN or Domino Named Network.

NNS Notes Name Service. Domino service for converting Fully Qualified Domain Names or Host names to a TCP/IP address.

NNSC Notes Name Service Cache. Hidden fields in the user's personal NAB Location document that is used when accessing the Domino server in question or in the server document of the server that initiated the connection to the other Domino server.

NNTP Network News Transfer Protocol. Protocol that supports reading newsgroups, posting new articles, and transferring articles between news servers. When enabled on a Domino server, allows NNTP clients to access newsgroups on the server and allows the Domino server to exchange news with other NNTP servers.

NRPC or **Notes RPC** Notes remote procedure call. This is the architectural layer of Notes used for all Notes-to-Notes communication. You can set up either the HTTP or the SOCKS proxy to work with RPC.

NSF The file extension for a Notes database file (.NSF). A database file contains the data for an application. Its structure is composed of forms, fields, folders, views, and other presentation features, such as a navigator and a database icon.

NTF The file extension for a Notes template file (.NTF). A template file contains the structure for the database--that is, forms, folders, and views--but does not contain documents. Domino Designer comes with a collection of templates that you can use to create system and application databases.

NTI Notes Transport Interface. The component of Domino server that manages communications.

ODBC Open Database Connectivity. A standard developed by Microsoft for accessing external data. ODBC has four components: the ODBC-enabled application, the ODBC Driver Manager, ODBC drivers, and data sources. Lotus Notes is an ODBC-enabled application.

ODS On Disk Structure. The format in which a Notes database is physically stored on disk. Also called a database format. In Release 5, the ODS version of a database is

listed on the Info tab of the Database Properties box.

POP3 Post Office Protocol (level 3). A mail protocol that allows clients running it to retrieve mail from a host mail server also running the protocol. You can enable POP3 on a Domino server.

PTF Program Temporary Fix. A temporary solution or by-pass for a problem diagnosed by IBM as resulting from an error in a current unaltered release of the program.

QMR Quarterly Maintenance Release. Lotus software upgrade package, distributed quarterly, to give customers the latest release level of Domino/Notes code. The entire package must be installed, as it is a complete code replacement.

QMU Quarterly Maintenance Upgrade. Lotus software upgrade package distributed between QMRs and designated by a letter, for example 5.0.4a. It is for critical or widespread fixes to Domino code.

RMF Resource Measurement Facility. Monitors the activity of the OS/390 system. It is useful for capacity planning, performance tuning, and problem diagnosis. RMF gives information about system resources, workload groups, DASD controllers, and DASD volumes.

SIMS Support Information Management System. Lotus internal databases for tracking problems

SMF System Management Facilities. OS/390 facility for recording performance and capacity data.

S/MIME Secure/MIME. A secure version of the MIME protocol that allows users to send encrypted and electronically signed mail messages, even if users have different mail programs.

SDSF Spool Display Search Facility. OS/390 tool for JES2 systems that allows a TSO user to view androute output from the JES2 spool and to view the SYSLOG and issue commands to OS/390 if authorized.

SLA Service level agreement.

SMTP Simple Mail Transfer Protocol. The Internet standard host-to-host mail transport protocol. It traditionally operates over TCP/IP using port 25. SMTP does not provide a mailbox facility, or any special features beyond basic mail transport.

SNMP Simple Network Management Protocol. A TCP/IP protocol to enable managing remote hosts.

SPR Software Problem Record. Lotus software problem record tracking record.

SSL Secure Sockets Layer. Security protocol for the Internet and intranets that provides communications privacy and authentication for Domino server tasks that operate over TCP.

TCB Task Control Block. An OS/390 construct that represents a dispatchable unit of work. One address space can create multiple tasks. A USS thread creates a TCB in OS/390.

TCP/IP Transmission Control Protocol/Internet Protocol. Network protocols that define the Internet. Originally designed for UNIX, TCP/IP software is now available for every major computer operating system.

TDP Tivoli Data Protection.

terabyte A trillion characters of data.

TSO/E Time Sharing Option/Extended. The principal interactive interface to OS/390.

UNIX An operating system developed at Bell Laboratories (trademark of UNIX System Laboratories, licensed exclusively by X/Open Company, Ltd.).

URL Uniform Resource Locator

VIPA Virtual Internet Protocol (IP) addressing. A virtual TCP/IP address that can direct network traffic to more than one physical adapter.

VTOC Volume Table of Contents. A list on each disk volume of the files and space it contains.

WAN Wide area network.

Index

Symbols

A

B

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5984-00 Lotus Domino for S/390: Running a Large Domino System
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

IBM

Redbooks

Lotus Domino for S/390: Running a Large Domino System

(0.2”spine)
0.17”<->0.4

Lotus Domino for S/390:

Running a Large Domino System

Sample architecture, topologies, and design for large Domino servers

Tips for operations, performance, and problem determination

Administration and maintenance techniques

This IBM redbook describes how to run a large Domino for S/390 production system from both S/390 and Notes administration perspectives. It documents the experience of running the world's largest Domino/Notes e-mail installation at IBM Poughkeepsie, as well as the experiences of other consultants and customers from around the world. The book contains answers to questions commonly asked by customers, and these solutions are valid for any Domino platform.

Part 1 describes the architecture, topologies and design of a large Domino system. It focuses on the organizational aspects and challenges of running Domino on OS/390.

Part 2 reveals the secrets of operating, administering and monitoring a large Domino system on OS/390. It describes backup and recovery possibilities, problem determination, maintainance of software, and hints and tips. As well, it provides sample shell scripts, Notes API programs, and LotusScript agents to aid in administration.

This book is intended for Domino administators, OS/390 system programmers, and operators responsible for installing, configuring, operating, and administering a Domino system on OS/390.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5984-00 ISBN 0738419508