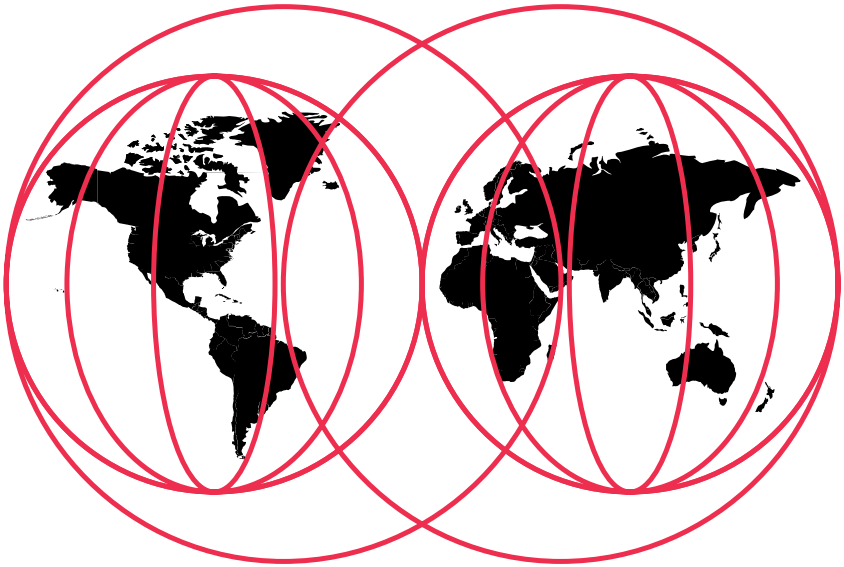
Lotus

IBM

Lotus Notes and Domino R5.0 Security Infrastructure Revealed

Søren Peter Nielsen, Frederic Dahm, Marc Lüscher, Hidenobu Yamamoto, Fiona Collins, Brian Denholm, Suresh Kumar, John Softley



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5341-00

SG24-5341-00

International Technical Support Organization

Lotus Notes and Domino R5.0 Security Infrastructure Revealed

May 1999

Take Note!
Before using this information and the product it supports, be sure to read the general information in the Special Notices section at the back of this book.

First Edition (May 1999)

This edition applies to Lotus Domino Release 5.0.

Comments may be addressed to: IBM Corporation, International Technical Support Organization
Dept. JN9B Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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

Preface

This redbook describes how to build a secure infrastructure with Lotus Notes and Domino R5.0. It reveals the following:

- The strong security infrastructure that has always been part of Notes and Domino
- How Notes and Domino R5.0 supports today's open Internet security standards (X.509)
- How Domino can be part of full security solutions that go beyond single systems, single platforms and single companies

First we introduce basic security concepts and the Domino family in general.

We will then describe the public key infrastructure that has always been a part of Notes. We will discuss the content of the Notes ID, certification hierarchies, and authentication between client and server, and we will explain how mail encryption and mail signing work in Notes.

Following that, we discuss how Domino supports secure Internet protocols such as SSL and S/MIME using X.509 certificates. We show how to acquire an X.509 certificate from a browser or a Notes client, and how to set up a Domino server as a Certificate Authority.

The latter part of the redbook is about Domino as part of a full security solution. A chapter is dedicated to Domino and firewalls. Examples are given of how to replicate or how to send SMTP mail through a firewall.

In the last chapter, we discuss the directory requirements for single sign on (SSO) solutions. We show how Domino Directory supports directory catalogs and can act as an LDAP service provider. We also discuss how Domino and IS can be integrated and allow single sign on. Finally, single sign on for Domino users running against Windows NT and OS/400 is discussed.

This redbook is written for Domino technical specialists and administrators, customers, Business Partners, and members of the IBM and Lotus community who need a good technical understanding of how to deploy Lotus Notes and Domino R5.0 in a secure environment.

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 Center at Lotus in Cambridge, Massachusetts, USA.

Søren Peter Nielsen works for the International Technical Support Organization at Lotus Development, Cambridge, Massachusetts. He manages projects with the objective of producing redbooks on all types of Lotus products. Before joining the ITSO in 1998, Søren worked as an IT Architect for IBM Global Services in Denmark on solutions for a wide range of industries. Søren is a Certified Lotus Professional at the Principal level in Application Development and System Administration.

Frederic Dahm is a Senior Systems Engineer currently working in Ottawa, Ontario, Canada. He also works as part of the Canada East team. In the field of Information Technology, Frederic has more than 10 years of experience in Systems Architecture, Computer Security, Application Development and Systems Administration. He is a Certified Lotus Professional in Application Development and System Administration. Other information pertaining to his Super.Human traits and qualities cannot be disclosed in this redbook as they are provided only on a highly classified "need to know" basis**.** Frederic can be reached at frederic.dahm@lotus.com**.**

Marc Lüscher is working as a System Architect to assist Lotus Professional Services in UK, Germany and Switzerland in selling and building complex solutions in the area of Domino System Architecture and Security, with a strong focus on directories, PKI's and Secure Messaging. Marc has over 5 years experience in consulting with client organizations in the pharmaceutical, banking, transportation and telecommunications industries. He is a Certified Lotus Professional at the Principal level in application development and system administration. Marc can be reached at marc_luescher@lotus.com**.**

Hidenobu Yamamoto is an IT Specialist at Advanced Technical Support, IBM Japan. He joined IBM in 1993 and has been working in various areas of technical support since then. Recently he has been focusing on the security aspects of e-business technology, especially integrated security/directory architecture based on Internet PKI in the multi-platform environment. He has also participated in the writing of several other ITSO redbooks**.**

A number of people have provided support and guidance. In particular, we would like to thank **Kevin Lynch**, Product Manager for Domino Security. In addition, we would like to thank the following people from Lotus (unless otherwise noted):

- Patricia Booth
- Charlie Brown
- Peter Carrescia
- Joanne Clerk
- Oliver Frömel, PRS GmbH Germany
- David Goodman
- Daniel H. Jaffe
- Charlie Kaufmann, Iris Associates
- Bob Lomme, Iris Associates
- Stacy O'Neil, Iris Associates
- Michele Pennell
- Gail Shlansky
- Catherine Stone
- Graphic Services, Lotus North Reading

Material from a related project at the ITSO Center at Lotus in Cambridge in 1998 have also been utilized. The specialists that worked on that project are listed below. We want to extend a special thanks to Suresh Kumar for helping out remotely on the last project as well.

Fiona Collins is an International Technical Support Specialist for Notes and Domino at the International Technical Support Organization Center at Lotus Development, Cambridge, Massachusetts. She manages projects with the objective of producing redbooks on all areas of Domino. Before joining the ITSO in 1996, she provided technical support for Lotus Notes/Domino as well as the AS/400 for Lotus and IBM in the UK.

Brian Denholm is an Advisory IT Specialist working in the Systems Integration Department for IBM New Zealand. He started with IBM in 1986 as a System/370 Engineer. Over the next 10 years he found himself fixing all types of hardware and software problems. After moving into SI in 1996 he has been working on various customer projects, implementing secure Notes/Domino e-business solutions in complex multi-platform environments. He hopes, in a future project, to implement Domino on the System/390 and get back to working with real machines!

Suresh Kumar is an I/T Specialist working for IBM UK, Hursley, in the e-business Services group of IBM Global Services. His broad range of technical knowledge in networking operating systems, Internet technology, and messaging systems is put to use in designing and implementing solutions for many of IBM's clients. Suresh has been working in IBM for 6 years, the last 4 years working with Lotus Notes and Domino on a range of infrastructure rollout, messaging migration, and systems integration projects.

John Softley is the Managing Director of Team Technologies Ltd, a UK Premium Business Partner based in Surrey, England. He has been in computing for 18 years and has worked for several hardware manufacturers, prior to setting up his own Notes Consultancy in 1992. His continued expertise in leading-edge technology ensures that his development team is one of the best in its field, specializing in workflow and e-commerce solutions, which integrate into back-end systems. In response to a growing concern in the industry with regard to security issues, John has chosen to focus on this area of consultancy.

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 at the back of this book to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Chapter 1
Basic Security Concepts Revealed

Two revolutions have greatly reshaped the world in which we do business.

The first revolution occurred in the early 1980s with the advent of the IBM® Personal Computer, which was the first true business microcomputer on the market. It permitted companies and individuals alike to have access to computing resources which at that time were relatively expensive. Connecting these machines through a local network enabled the flow of information like never before. This led to an explosion of business solutions that have significantly changed the way businesses operate.

The second revolution occurred in the mid 1990s, when a twenty-year-old collection of networks (originally named ARPANet, now called the Internet) became joined with the Web browser, finally making it easy to access information on the public network. This final piece launched the world into a new era in its history: the information age.

As its name implies, in this new age, information has become a vital commodity. It is correctly referred to as *knowledge capital*, which is a type of asset that many businesses depend on to the same degree as monetary capital. As a matter of fact, businesses live and die by the measure of control they have over their knowledge capital. If this capital is stolen or disseminated, a company can suffer greatly.

There are individuals who make it their purpose in life to penetrate computer systems and networks to gain access to the information they contain. Viewed in the best possible light, these individuals do it for the sheer thrill of it, to boast of their computer skills and nothing more (they are usually referred to as *hackers)*. In the worst possible light, these individuals do it for malicious purposes, either to gain from it financially or to willfully corrupt or destroy what they find (they are usually referred to as *crackers*).

No matter what their reasons, these people are your company's computer systems' worst nightmare. Even hackers without malicious intent can create conditions that compromise the information in your company's computer systems and allow the potential for this information to be destroyed, corrupted, or accessed by less scrupulous people.

It is thus crucial for companies to ensure the safety of their knowledge capital and adopt specific measures to guard it against attack, theft, or disclosure. This is where the topic of security in the information technology world comes into play.

The purpose then of this chapter is to introduce you to computer security.

We will cover basic terminology, an overview of computer security and the cryptographic tools, and techniques and mechanisms that are generally available.

Even though the topics presented here are general in nature (that is, they are not specific to Notes™ and Domino™), you should take the time to carefully read and understand the content, because this chapter lays the groundwork for the rest of this redbook.

Important Terminology

In order to provide a consistent understanding of the terms and concepts used throughout this redbook, you should be familiar with the definitions presented below.

Computer System

Even though this book specifically deals with security features and facilities of Notes and Domino R5.0, it is important to understand that computer security applies to computer systems as whole entities.

A computer system, by definition, includes all the necessary software (that is, the operating system and the applications that reside on top of it) and all the necessary hardware (that is, all the physical aspects of the computer).

The definition for hardware is not solely limited to the computer, it also includes all connectivity and telecommunication devices, such as hubs, routers, gateways, switches, and so on.

Computer Security

On page 6 of the publication *An Introduction to Computer Security: The NIST Handbook*, *Special publication 800-12*, the National Institute of Standards and Technology (NIST) defines computer security as:

the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources (includes hardware, software, firmware, information/ data and telecommunications).

In other words, computer security is a facet of computer science whose primary objective is to assure safety of information and to offer measures to guard against attack, theft, or disclosure so that:

- The information is timely, accurate, complete, and consistent and when transmitted over a computer network, that it has not been changed during transmission (integrity);
- The information is inaccessible to anyone but the people to whom it is intended to be seen. When transmitted over a computer network, that it is only accessible by the sender and receiver (privacy);
- The receiver that accesses or receives the information can have the proper assurance that it was created or was sent by the original author (authenticity);
- The sender can be sure that people accessing the information are genuine. When transmitted over a computer network, that the receiver is genuine (non-fabrication and authentication);
- The author cannot deny that the information was created by him or her. When transmitted over a computer network, the sender cannot deny he or she sent the information (non-repudiation).

In addition to understanding the concepts above, it is also important to understand the nature of the information you and your company possess and to understand the concept of sensitive information.

Sensitive Information

The *Computer Security Act of 1987*, *Public Law 100-235 (H.R. 145), January 8, 1988*, provides this definition of "sensitive" information:

"(4) the term 'sensitive information' means any information, the loss, misuse, or unauthorized access to or modification of which could adversely affect the national interest or the conduct of Federal programs, or the privacy to which individuals are entitled under section 552a of title 5, United States Code (the Privacy Act), but which has not been specifically authorized under criteria established by an Executive order or an Act of Congress to be kept secret in the interest of national defense or foreign policy"

Even though the Computer Security act has been passed in the United States and aims to protect the interests of this country, its application is global and the definition above applies to any country and any company in the world. In other words, sensitive information is such that it needs to be kept confidential and must be protected from unauthorized access and disclosure. Furthermore, this also means that appropriate measures also must be applied to prevent the destruction or alteration of this information.

Computer Security Services

Now that we've identified the nature of sensitive information, it is now necessary to define the services that enable this information to be protected.

The definitions used in this redbook can be best explained by using the IBM Security Architecture, which is based on ISO 7498 security standards.

The IBM Security Architecture is a model for integrating security services, mechanisms, objects, and management functions, across multiple hardware and software platforms and networks. The architecture supports the strategy for providing end-to-end protection of applications and information within an organization.

The descriptions of these security services are drawn from the IBM redbook *Enterprise-Wide Security Architecture and Solutions*, SG24-4579, and can be categorized as follows:

- Data integrity
- Confidentiality
- Identification and authentication
- Access control
- Non-repudiation

Keep in mind that the categories are not exclusive; for example, you cannot do access control without also addressing questions of authentication and data integrity.

Data Integrity

Data integrity provides detection of the unauthorized modification of data.

Organizations must allow for the use of data by authorized users and applications, as well as the transmission of data for remote processing, while at the same time, ensuring that this information is not altered by unauthorized users. Data integrity facilities can indicate whether information has been altered.

Data may be altered for two reasons: because of hardware or transmission errors or because of an attack.

For years, many IBM products have used a check sum mechanism in disk and tape storage systems and in network protocols to protect against transmission and hardware errors. Active attacks on data integrity require a different mechanism, which uses cryptography and allows for the verification of data integrity.

To address active attacks on data integrity, products must support message authentication based on cryptographic functions that adhere to international standards.

Confidentiality

Confidentiality protects sensitive information from disclosure.

When it is stored locally, sensitive data can be protected by access controls or encryption mechanisms. For network communication security, sensitive data should be encrypted as it is transmitted from system to system.

There are specific ISO standards (8730, 8731, and 9564) relating to the use of cryptography for confidentiality and data integrity.

Identification and Authentication

Identification and Authentication (I&A) facilities verify the identity of individuals.

The basic function uniquely identifies users and programs, verifies these identities, and assures individual accountability.

Authentication may be single authentication, for an individual user of the system, mutual authentication of peers, such as two-party authentication for distributed applications, or three-party authentication when dealing with local authentication servers in a distributed environment.

Authenticated user identification provides the basis for additional security functions, such as access control and auditing. Authentication technology may take the following forms:

- Passwords — which can be simple responses to basic authentication challenges or be used to decrypt a Notes ID as the basis of a sophisticated authentication scheme.
- Smart tokens — which are easily portable devices that do special-purpose operations for their users, in this case, generally identifying the user to a secure system. A smart token can look like any common object: a credit card, a 3 1/2" floppy disk or even a ring (like Sun's Java ring). The important trait of this object is that it carries some secret information for its user and performs the function required when needed. A smart token is often designed to be tamper-resistant; It is difficult to take apart. It is protected with a user password, so that even if it is physically stolen, it will be difficult for someone else to impersonate its owner.

- Smart cards — which are small electronic devices about the size of a credit card. These are built a little bit like a prepaid phone card in that they contain some electronics in the form of memory and an integrated circuit (IC) for processing of some data. The main purpose of such smart cards is or storing network IDs (very similar to a a smart token).

Note To use a smart card or smart token, either to pull information from it or add data to it, you need a smart card or smart token reader, a small device into which you insert the smart card or smart token.

Access Control

Access control allows the installation to protect critical resources by limiting access to authorized and authenticated users.

Depending on the environment, access may be controlled by the resource owner, or, it may be done automatically by the system through security labels.

The resource owner can specify who can access the information, how it can be accessed, when it can be accessed, and under what conditions it can be accessed (for example, when executing specific applications, programs, or transactions).

The functional goal is to assure that security is maintained for resources, whether they are in a central system, distributed, or mobile (as in the case with files and programs).

Non-Repudiation

Non-repudiation may be viewed as an extension to the usual identification and authentication services.

The non-repudiation service can protect a recipient against the false denial by an originator that the data has been sent, and it can protect an originator against the false denial of a recipient that the data has been received.

In general, non-repudiation applies to the transmission of electronic data, such as an order to a stock broker to buy/sell stock, a doctor's order for medication to a specific patient, or approval to pay an invoice by a company to its bank.

The overall goal is to be able to verify, with virtually 100% certainty, that a particular message can be associated with a particular individual, just as a handwritten signature on a bank check is tied back to the account owner.

Computer Security Objectives

The basis for the successful application of computer security is not obtained by buying a set of software or hardware tools and services and applying them to your computer system. Doing so might result in marginally increased protection of your sensitive information, but it cannot be used as a way to achieve the level of assurance required. In fact, it might have the opposite result. It might provide you and your company with a false sense of security that leaves you open to the very attacks and risks that you are trying to avoid in the first place.

In this section, we will cover the minimum of what you must do in order to bring about a comprehensive policy, architecture, and implementation of computer security services.

The Computer Security Policy

The successful application of computer security is done by understanding the needs of your company in matters of computer security and developing an appropriate and well-designed computer security policy.

The computer security policy of your company will outline the architecture of your computer security services. It will also help determine (and prioritize) the activities and tools that are required to ensure that your computer security objectives are met, both by applying computer security tools, techniques, and mechanisms, as well as factoring in the risks involved.

The security policy and the resulting computer security architecture may differ significantly from those at another company. This is because your security needs as well as the threats to your sensitive information are unique to your company.

Most companies see computer security in a negative light. Sure, there is a need to protect sensitive information, but in most cases, this is done by adding a layer of complexity to the company's computer system.

The perceived result, in many cases, is that the computer system becomes difficult to operate and not at all user-friendly.

For this reason, the long-term goal and number two objective (since the first and foremost objective is to secure your sensitive information) is to work towards a computer security architecture that is completely transparent or as transparent as possible, to the end-user.

An ideal environment would permit access to your computer system and the sensitive information it contains without the user being aware that a security mechanism exists. However, it should also ensure that any and all improper access is denied effectively.

Lotus® has done some great work in that respect — few users realize the extent of the computer security architecture built into the Notes client and Domino server because it is so transparent.

An ideal environment with almost transparent security architecture would also allow single sign-on to all resources provided by a computer system. We will discuss single sign-on in the context of Lotus Domino R5.0 later in this redbook.

The Security Architecture

As stated above, computer security is a facet of computer science. As such, there are two important things to keep in mind.

First, there is no exact, quantifiable, measure of "safety." While there are degrees of computer safety, these are more qualitative than anything else, as in: "the information is 'pretty' safe."

Second, computer security relies on computers to safeguard the information. While computers can be programmed and configured to offer those safeguards, other computers can be used to thwart them. All it takes is a person who has the will, the skill, the time and the computer processing power to penetrate and overcome the defenses in place.

You therefore have to accept that you must face some uncertainty about the level of protection that you are applying to your company's sensitive information. It must be stressed that not even an over-investment in computer security — hardware or software — tools will provide you with the level of protection that can allow you to claim that your information is completely and totally safe.

You also should understand that when all aspects of your security policy have been put in place and all the tools and mechanisms have been implemented and tested, the work doesn't end there.

You need to monitor your computer environment for unexpected, unusual, and/or inappropriate activity on a continuous basis. Doing so will provide the necessary assurance that your security architecture is adequate, that the proper safeguards are in place, that no back doors or security holes exist, and that your information is indeed, secure.

Based on all that, understand that computer security simply deals with determining the risks involved and applying the proper methods and safeguards to reduce these risks to an acceptable level. Although it sounds like gambling, it isn't necessarily.

Understanding the Risks

A comprehensive portion of your security policy has to deal with the manner in which you manage the risks faced by your company's computer system.

Often, companies and their IT personnel do not understand the nature of those risks. With all the hype in the media (both printed and electronic), they assume that the real danger comes from the Internet and from people outside of the company.

After all, the portrait that is constantly painted of these individuals is that they are poorly dressed Generation X'ers that have no better thing to do in life than to scour the Internet trying to find vulnerable systems to attack, penetrate and maliciously destroy or corrupt. A prime example of external threats can be found in Clifford Stohl's excellent, true story, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage* (Mass Market Paperback Reprint edition, July 1995, Pocket Books, ISBN: 0671726889). In this book the young cracker wreaks havoc left and right, using the computer systems of one company as the springboard to attack the systems of other companies and organizations.

Therefore, upper-level management at various companies believe that if they properly secure their computer system from people on the outside of the company and shield access from the Internet, they have effectively secured their computer system. At this point, they feel they can cease the work of hardening and securing the system and finally sleep soundly at night.

The bad news is that this view of the computer security world is flawed and shortsighted. The sad reality is that many companies have suffered and died from attacks perpetrated by insiders. These are real attacks by disgruntled employees who then use the newly acquired information to their advantage.

Following are a couple of examples. The names of the companies have been withheld, since the purpose here is to not to further abuse the affected companies, but to expose some of the risks companies face by insiders.

Before exiting a high tech engineering company after being fired, an employee accessed a privileged engineering database and erased part of the designs of a new product the company was depending upon for its survival. The delay in rebuilding the design nearly killed the company.

An employee of a software development firm took all the source code produced by the company and mailed it to another firm. The company for which that employee worked went out of business.

Comparatively, companies are far more likely to suffer serious losses from such insider attacks than from attacks by people outside the company. The latter generally cause embarrassment to a company, while the former can even destroy it.

This is exemplified by the fact that for two days in January 1999, hackers repeatedly tapped into US military computers. The systems were architected against that type of attack and the hackers didn't access top secrets. Had the attack been staged from the inside, from a person with privileged access to the information and without safeguards to protect it, the information could have been placed in the wrong hands with the worst possible consequences for the military.

Therefore, it is imperative that your computer security policy include the necessary safeguards to protect information from people on both sides of the virtual security fence.

With your security policy in hand, which provides you with an understanding of the security threats and risks involved, you now have an idea of the measures you must adopt to secure your sensitive information.

At this point, you are ready to design your computer security architecture and implement the computer security services.

Establishing the Security Services

Establishing the computer security architecture and the services that comprise it is by no means a trivial task. This is not the kind of thing that gets done in one evening by working late at the office and ordering in some pizza.

This section aims to give you an overview of the steps involved in designing and establishing a computer security architecture.

Before you start

There are a number of steps which must be carefully undertaken and all these *must* flow from the newly made computer security policy. So, as a precursor to the implementation of the architecture, it is necessary to ensure that:

1. The computer security policy is complete.
2. It has been revised at all levels (not only the computer systems personnel).
3. It has been approved by upper management and they have an understanding of what it represents:
   - The limitations of the computer security policy;
   - The impositions of security steps to access information via the company's computer system; and, given that,
   - Their willingness to see it implemented.

Once all these steps have been accomplished, then the real work begins, all organized in a number of distinctive steps.

Step One — Identify Your Corporate Information

You need to identify the information that is in the company's possession and to determine the level of sensitivity specific pieces of information have.

This information can be classified using labels, from most sensitive to least sensitive, such as: Top Secret, Secret, Confidential, Protected Sensitive, Protected, and Undesignated or Unclassified.

Step Two — Define your Security Domain

You need to define your security domain, which determines how far reaching the application of your computer security architecture will be. This step will help you apply the security policy of your domain and help you deal with security issues specific to each domain, should you have more than one.

If you are already familiar with the concept of a Notes domain, note that a security domain may or may not be the same as your Notes Domain. You might discover that your Notes Domain is only a part of everything you must secure and plan security contingencies for (such as a perimeter network and other key elements of your security policy).

You may also decide that, while your security domain spans the entire computer system, you may want to deal with different subsets of the domain, given the security that resides in each subset. For example, if you have a particularly large Notes and Domino infrastructure that is installed all over the globe, it might be wise to subdivide your security domain into separate and distinct security sub-domains. This is necessary because the security services that can be applied locally may be different from those applicable in other countries, due to export restrictions on security mechanisms and tools, as well as legislation in some countries regarding said mechanisms and tools.

Step Three — Evaluate Threats and Risks

Determine, within your security domain, what possible inside and outside threats exist and, considering the sensitivity of information already identified in your security domain, determine what security tools, techniques and mechanisms you wish to apply to counter those threats and reduce the risks to a minimum.

Using the computer security policy as the basis for making these necessary decisions, you will be able to prioritize the implementation of key elements of your security architecture.

Step Four — Apply the Security Techniques and Mechanisms

This is essentially the culmination point of all the preparatory work you have done in order to define your computer security policy and your computer security architecture.

In places within your security domain where you have computing devices handling highly sensitive information or where the actual threat of attack is high, this is where you should apply the strongest and most effective security tools, techniques, and mechanisms.

In other places within your security domain where you have computing devices handling less sensitive information or areas that are less threatened by attacks, you may not require as big an investment in security tools, techniques, and mechanisms (and directly, would require a lesser investment in funds).

If you have more than one security domain, it is wise to look at each domain separately, so as not to make unwarranted assumptions about the nature of the threats, the degree of risk, or the nature of the information. This will prevent you from applying improper security tools, techniques, and mechanisms.

Finally, you will have to make sure that the computer security architecture and implemented tools, techniques, and mechanisms are constantly monitored to ensure that they meet the requirements of the computer security policy, satisfy the business needs of the company, and protect your company's information adequately.

Finally, Savor the Fruits of Your Labor

Despite the negative image that computer security has in the eyes of information technology personnel and upper management, applying a sound computer security policy and computer security architecture can actually be the best thing a company can do for itself.

As such, some of the direct benefits of applying a sound security policy include a secure and well thought out computer architecture, as well as:

- Better organization of information and services
- Better assurance of the integrity of the information
- Better overall availability of the computer systems
- Better visibility for Information Technology personnel in the eyes of upper management
- Better computer systems overall

In the end, a well-conceived and well implemented security infrastructure, based on a sound security policy, can be a win/win situation for everyone in the company.

What remains then is a dissertation on the tools, techniques, and mechanisms involved.

Cryptographic Techniques

Notes and Internet security mechanisms make use of a number of common cryptographic techniques. It is important to have a good understanding of these techniques; generally throughout the book, we assume that you have some basic knowledge of them.

However, this is a complex area, so in this section we present a brief overview of the important cryptographic techniques. We believe that you will find this a useful resource and urge you not to skip it.

We will discuss five subject areas:

1. Symmetric key (or bulk) encryption
2. Public key encryption
3. Secure hash (or digest) functions
4. Digital signatures and other combinations of the above techniques
5. Certification mechanisms

If you want to learn more about cryptography, we recommend the RSA Frequently Asked Questions document at:

`http://www.rsa.com/rsalabs/newfaq`

Symmetric Key Encryption

This is a grown-up version of the kind of secret code that most of us played with at some time during childhood. Usually these use a simple character replacement algorithm; if you want to encrypt a message, you just replace each letter of the alphabet with another. For example:

Original letter: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Replacement : GHIJKLMNOPQRSTUVWXYZABCDEF

In this case the letters in the alphabet have just been shifted seven places to the right, so HELLO WORLD would translate to NKRRU CUXRJ. The premise on which this code is based is that both the sender and the receiver know a common key, in this case the number of places to shift the letters.

This *shared secret* allows the receiver of the message to reverse the encryption process and read the scrambled message. Symmetric encryption algorithms used by computers have the same elements as this simple example, namely a mechanism to scramble the message (also known as a cipher) and a shared secret (a key) that allows the receiver to unscramble the encrypted message.

The strength of a symmetric key cipher of this kind is dictated by a number of factors. For example, it is important that it effectively randomizes the output, so that two related clear-text messages do not produce similar encrypted results.

Our childish example falls down badly in this area, because each letter always converts to the same encrypted result, and because it does not encrypt spaces. The kindergarten cryptanalyst can quite easily break the code by knowing that any one-letter word is likely to be an A.

For full-strength symmetric ciphers, much of the work of the cryptanalyst involves trying to find patterns in the result of the algorithm, to use as a shortcut to breaking the code.

If the encryption algorithm has no flaws of this kind, the main factor governing its strength is the size of the *key space*; that is, the total number of possible shared secrets. Once again our simple example falls short, because it only has 25 possible places where we can shift the keys. We could mount a brute force attack very easily by trying each key in turn until we find a message that makes sense.

Real symmetric ciphers use numeric keys, usually of between 40 and 128 bits in size. Even for the smallest of these a brute force attack has to try, on average, 2 to the power 39 or about 550,000,000,000 possible keys. Each extra bit of key size doubles the key space.

Characteristics of Symmetric Key Algorithms

There are a number of symmetric key ciphers in use. We have listed the main ones below together with a brief description of their capabilities. However, they also share several common characteristics:

- They are fast and need relatively little system overhead. For this reason symmetric key encryption is often referred to as bulk encryption, because it is effective on large data volumes.
- The algorithms are published openly and there are no commercial licensing issues to be considered in implementing them.
- They all fall under the control of the US National Security Agency export restrictions. The precise operation of these restrictions is not a simple matter; in essence that means:

  1. Any software incorporating cryptographic technology that is exported by a US company has to have a special export license.
  2. If the product includes symmetric encryption code that can be used for encrypting an arbitrary data stream, the license will *only* allow unrestricted export *if* the key size is smaller than a given, NSA-specified, value.

  What this means is that to export full-strength cryptography, a company has to have a special license for each customer. Such licenses are only issued for customers that the US government considers to be friendly, such as major banks and subsidiaries of US companies.

Until recently the threshold key size for a general export license was 40 bits. Several challenges have shown that a brute force attack can be mounted against a 40-bit key with relatively modest computing power. A government announcement in October 1996 opened the door to the use of larger keys, initially up to 56 bits, with the promise of unlimited key sizes when the computer industry develops effective key recovery technology. (Key recovery means that the key for a session can be discovered, given the knowledge of some other, master, key). 56 bits may not sound a lot better than 40, but in fact it is 2 to the power 16, or 65,536 times more difficult to crack.

As a follow-up to this, there was a November 18, 1998 announcement by the Bureau of Export Administration, Commerce that amended the export administration regulations for exports and re-exports of strong encryption commodities and software companies. The key lengths are now a full 56 bits for DES and "equivalent" bulk cyphers (namely RC2, RC4, RC5 and CAST) and 1,024 bits for RSA asymmetric keys to all destinations except Cuba, Iran, Iraq, Libya, North Korea, Sudan, and Syria. This is also under the proviso that there be no key recovery possible. Additionally, unlimited strength crypto keys can be used by US subsidiaries, insurance companies, health and medical firms and online merchants, provided they do not have a presence in any of the previously listed countries.

Common Symmetric Key Algorithms

DES

The Data Encryption Standard is the most commonly-used bulk cipher. It was originally developed by IBM in 1977 and has resisted all attempts at cryptanalysis. DES breaks the data to be encrypted into a sequence of 64-bit blocks and then uses a 56-bit key to apply a sequence of mathematical transforms to it. There are a number of variations on the standard DES algorithm, such as cipher block chaining, in which each block of data is XOR'd (XOR is a Boolean operator that returns a value of TRUE only if both its operands have different values. This is in contrast with the inclusive OR operator, which returns a value of TRUE if either of its operands is TRUE. Whereas an inclusive OR can be translated "this, that, or both," an exclusive OR means "this or that, but not both.") with the previous block before encryption, and triple-DES in which DES is applied three times in succession.

RC2/RC4

These related ciphers were developed by RSA Data Security, Inc. RC2 is a block cipher similar to DES, whereas RC4 operates on a stream of data. They both use a 128-bit key, but they support key masking. This means that part of the key may be set to a known value so that the effective key size is

whatever remains from the 128-bit total. This has been used effectively in producing 40-bit versions of software products for export.

IDEA
The International Data Encryption Algorithm is another block cipher, in the mold of DES. It uses a 64-bit block size and a 128-bit key. IDEA is the bulk encryption algorithm used by Pretty Good Privacy (PGP).

Public Key Encryption

A non-mathematician can intuitively understand how a symmetric key cipher works by extrapolating from a familiar base. However, public key mechanisms are much less accessible to the lay person. In fact, it sometimes seems more like magic than technology. The fundamentals of public key encryption are:

1. Instead of a single encryption key, there are two related keys, a *key pair*.
2. Anything encrypted using one of the two keys can *only* be decrypted with the other key of the pair.

Let us say that two people, Bob and Alice, want to exchange messages using a public key algorithm. Alice generates a key pair and places one of the keys, the *private key* in a safe place. She sends the other half of the key pair (called, naturally, the public key) to Bob. Bob can now encrypt a message using the public key and only the owner of the matching private key, Alice, can decrypt it. Of course, if Alice wants to send a reply, Bob needs to create his own key pair and send the public key to Alice.

The big advantage of this mechanism over the symmetric key mechanism is that there is no longer any secret to share. In fact, it does not matter who has the public key, because it is useless without the matching private key.

There is one further advantage that public key gives us. In the above example, imagine that Alice uses her private key to encrypt a message and sends it to Bob. The message that is sent between them is still scrambled, but it is no longer private, because anyone with the public key can decrypt it (and we have said that we do not care who has the public key). So, what *can* we use this message from Alice for? The answer is: authentication. Because only Alice has access to the private key that created the message, it can *only* have come from her.

Characteristics of Public Key Encryption

Public key algorithms can trace their ancestry back to the Diffie-Hellman key exchange mechanism. Diffie-Hellman is not a general-purpose encryption scheme, but rather a method of exchanging a secret key.

There is only one widely-used general purpose public key mechanism, the Rivest, Shamir and Adelman (RSA) algorithm, which is the property of RSA

Data Security, Inc. Public key, like any encryption mechanism, relies on the fact that certain mathematical problems are very hard to solve. The problem that the RSA algorithm relies on is the factorization of large numbers. For a detailed description of the mathematics behind RSA public key encryption, refer to

```
http://www.rsa.com/rsalabs/newfaq/q8.html
```

Clearly, public key has a big practical advantage over symmetric key, in that there is no need to securely share a secret between the sender and receiver.

However, the RSA algorithm is a much less efficient encryption technique than any commercial symmetric key algorithm, by a factor of 100 or more. It is therefore not a good choice for encryption of bulk data.

RSA is subject to the same US export restrictions as symmetric algorithms. However, the key in this case is actually a very large number. Very approximately, an RSA key size of 1024 bits corresponds to a full-strength symmetric key of 64 bits or more.

Secure Hash Functions

The third tool in our encryption armory is not actually an encryption mechanism at all. A secure hash is a function used to index the original value or key of a message or a block of data and then used later each time the data associated with the value or key is to be retrieved. A secure hash function has three main attributes:

1. It takes a message of any size and generates a small, fixed size, block of data from it (called a *message digest*). Re-executing the hash function on the same source data will always yield the same resulting digest.
2. It is not predictable in operation. That is to say, a small change in the source message will have an unpredictably large effect on the final digest.
3. It is, for all intents and purposes, irreversible. In other words there is no way to derive the source data, given its digested form.

What use is a secure hash function? Its main function is to detect whether a piece of data has been modified or not. These are used in combination with RSA to generate a digital signature.

There are two secure hash algorithms in common use. The most widely-implemented is MD5, which was developed by RSA Data Security, Inc. and is used in Notes. This generates a 128-bit digest from any length of input data and it is described in RFC1321.

The other algorithm that is becoming increasingly common is the US government-developed Secure Hash Standard (SHS). This produces a 160-bit digest, slightly larger than MD5.

Combinations of Cryptographic Techniques

Although the security protocols of Notes and the World Wide Web differ in detail they both use the three techniques above.

Two combinations of the techniques are used very commonly, they are outlined here:

- **Public Key Delivery of a Symmetric Key**

  It is most efficient to use a symmetric key algorithm for bulk data, but first you have to copy the key from sender to receiver.

  A common approach is to use a public key algorithm to protect the key transfer process.

- **Digital Signatures**

  You do not always want to encrypt data in transit. Very often the contents of a message may not be secret, but you do want to be sure that it really came from the apparent sender.

  If Alice wants to prove to Bob that it was really she who sent him a message she will attach a signature to the message, exactly as she would if writing a real letter on paper. In the digital case she creates the signature by first generating a digest of the message and then encrypting the digest with her private key. When Bob receives the message he first decrypts the digest (with Alice's public key) and then generates his own digest from the received message. If the two digests match, Bob knows that:

  1. The message is the same as when it was sent (because the digest is the same).
  2. It really was sent by Alice, because only she has the private key.

Public Key Certificates

We have seen how public key cryptography overcomes the problem of having to pass a secret from sender to receiver. There is still a need to send a key, but now it is a public key, that anyone can see because it is only useful to an attacker if he also has the private key. However, this overlooks one crucial element of trust: how can you be sure that the public key really came from who you think it came from?

One answer is to only pass public keys to someone you know. Bob and Alice know each other well so they could share their public keys by exchanging diskettes. Otherwise, you need some way to be sure that a public key is authentic.

The mechanism for doing this is the *public key certificate*. This is a data structure containing a public key, plus details of the owner of the key, all digitally signed by some trusted third party. Now when Alice wants to send Bob her public key she actually sends a certificate. Bob receives the certificate and checks the signature. As long as it has been signed by a certifier that he trusts, he can accept that this really is Alice's key.

In real life, certificates are more complex than this. Descriptions of how they are used in a variety of ways are detailed in the sections on SSL and Notes security later in this Redbook.

Public Key Cryptographic Standard

All these cryptographic tools and techniques are not much good without a set of related, agreed-upon, standards to provide the basis for interoperability. These are called *Public Key Cryptographic Standards* (PKCS).

PKCS is the informal inter-vendor standard that was developed in 1991 by RSA Laboratories with representatives of Apple, Digital, Lotus, Microsoft, MIT, Northern Telecom, Novell, and Sun. Since its publication in June 1991, PKCS has become a part of several standards and products, including Notes and Domino.

These standards cover RSA encryption, Diffie-Hellman key agreement, password-based encryption, extended-certificate syntax, cryptographic message syntax, private-key information syntax, and certification request syntax, as well as selected attributes.

Defined Standards

- PKCS #1: RSA Encryption Standard
- PKCS #2: See note below
- PKCS #3: Diffie-Hellman Key-Agreement Standard
- PKCS #4: See note below
- PKCS #5: Password-Based Encryption Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #9: Selected Attribute Types
- PKCS #10: Certification Request Syntax Standard
- PKCS #11: Cryptographic Token Interface Standard
- PKCS #12: Personal Information Exchange Syntax Standard

- PKCS #13: Elliptic Curve Cryptography Standard
- PKCS #15 (Draft): Cryptographic Token Information Format Standard

Note PKCS-2 and PKCS-4 have been incorporated into PKCS-1.

Of interest to us in this redbook are PKCS #1, PKCS #7, PKCS #10, and PKCS #12.

PKCS #1 describes a method for encrypting data using the RSA public-key cryptosystem. Its intended use is in the construction of digital signatures and digital envelopes, which are described in PKCS #7. PKCS #1 also describes the syntax for RSA public and private keys. The public-key syntax of PKCS #1 is identical to that of X.509.

PKCS #7 describes the *Cryptographic Message Syntax Standard.* It defines the syntax for several kinds of cryptographically protected messages, including encrypted messages and messages with digital signatures. PKCS #7 has become the basis for the Secure Multipurpose Internet Mail Extension (SMIME) standard, which provides a uniform method of encrypting browser-based e-mail. PKCS #7 has other applications, such as its use in PKCS #12.

PKCS #10 describes the syntax for certification requests. This certification request consists of a distinguished name, a public key, and an added set of optional attributes, which are all signed by the entity requesting certification. These certification requests are sent to a certification authority who transforms the request into an X.509 public-key certificate.

PKCS #12 describes an import/export syntax for personal identity information, including private keys, certificates, miscellaneous secrets, and extensions. Applications such as Web browsers support this standard, which allows a user to import and export a single set of identity information. This standard also serves for importing and exporting data from smart cards and smart tokens.

Complete information on PKCS, including a detailed description of each standard can be found at the RSA site at the following URL:

`http://www.rsa.com/rsalabs/pubs/PKCS/`

Summary

Now that we have explored all the basic security information and terminology as well as explained cryptographic tools, techniques, and standards, we are now ready to move on to the security facilities built into and offered by Notes and Domino.

Chapter 2
What Is Lotus Domino?

The Domino Server family is an integrated messaging and Web application software platform, for growing companies that need to improve customer responsiveness, and streamline their business processes.

Domino, the only solution built on an open, unified architecture, is trusted by the world's leading companies to deliver secure communication, collaboration, and business applications. Domino R5.0 servers set a new standard for rich Internet messaging, ease of administration, and integration with back-end systems.

This chapter describes the Domino R5.0 Server family, the services Domino R5.0 offers, and the clients for Domino R5.0.

Domino R5.0 Server

The Domino R5.0 Server is offered in different packages, to allow customers to choose the functionality that meets their current requirements and to extend that functionality as their requirements change in the future. We will briefly describe the Domino R5.0 Server family below.

Lotus QuickPlace

Lotus QuickPlace ™ is a teamware application server. It provides an accessible space for sharing common resources and documents among team members, regardless of whether these team members are outside your company firewall or using a different mail/messaging program. Its self-service design makes QuickPlace the ideal solution for ad hoc initiatives and projects that require an instant collaboration solution without the time and costs associated with custom application development. Its easy installation, low cost of deployment, and near zero learning curve dramatically lowers the necessity for any upfront investment. QuickPlace is included with the Domino Application Server and Domino Enterprise Server; it is offered as an individual product as well. QuickPlace will also be available as a "rentable" hosted solution.

Domino Mail Server

Domino Mail Server combines support for the latest Internet mail standards with the advanced messaging capabilities and enterprise-scale reliability and performance of Lotus Domino. Its integrated, cross-platform services include Web access, group scheduling, collaborative workspaces, and newsgroups — all accessible from a Web browser or other standards-based client.

Domino Mail Server is used for messaging only. Customers that want to deploy their own applications on the Domino server should consider Domino Application Server or Domino Enterprise Server.

Domino Application Server

Domino Application Server is the leading integrated messaging and applications server. It delivers best-of-breed messaging as well as an open, secure Web application platform. The server easily integrates back-end systems with front-end systems business processes.

This is the natural evolution of the Lotus Notes® server from which Lotus Domino originates.

Domino Enterprise Server

Domino Enterprise Server is the server for customers requiring mission-critical, highly scalable deployments with uninterrupted access, and maximum performance under all conditions. It extends the functionality of Domino Mail and Domino Application Servers with high availability services such as partitioning, clustering, and billing.

This product was previously called Domino Advanced Services.

Services Offered by Domino Servers

Lotus Domino Servers offer a wide range of services. We will briefly describe the most important ones.

Object Store

Documents in a Domino database can contain any number of objects and data types, including text, rich text, numerical data, structured data, images, graphics, sound, video, file attachments, embedded objects, and Java™ and ActiveX applets. A built-in full text search engine makes it easy to index and search documents. The object store also lets your Domino applications dynamically present information based on variables such as user identity, user preferences, user input, and time.

Directory

A single directory manages all resource directory information for server and network configuration, application management, and security. Domino includes

user account synchronization between NT and Domino and fully accepts client request for directory objects (or information you choose) via the Light Weight Directory Access Protocol (LDAP). The directory is the foundation for easily managing and securing your Internet and intranet applications.

Security

The Domino security model provides user authentication, digital signatures, flexible access control, and encryption. Domino security enables you to extend your intranet applications to customers and business partners. Hang on — you will learn a lot more about this in later chapters of this redbook.

Replication

Bi-directional replication automatically distributes and synchronizes information and applications across geographically dispersed sites. Through replication, Domino makes your business applications available to users around your company or around the world, regardless of time or location.

Messaging

An advanced client/server messaging system with built-in calendaring and scheduling enables individuals and groups to send and share information easily. Message transfer agents (MTAs) seamlessly extend the system to Simple Mail Transfer Protocol (SMTP)/Multipurpose Internet Mail Extension (MIME), x.400, and cc:Mail™ messaging environments. The Domino messaging service provides a single server supporting a variety of mail clients: Post Office Protocol V3 (POP3) clients, Internet Message Access Protocol V4 (IMAP4) clients, clients using the Message Application Programming Interface (MAPI), and Lotus Notes clients.

Workflow

A workflow engine distributes, routes, and tracks documents according to a process defined in your applications. Through workflow, Domino enables you to coordinate and streamline critical business activities across an organization, and with customers, partners, and suppliers.

Agents

Agents enable you to automate frequently performed processes, eliminating tedious administration tasks, and speeding your business applications. Agents can be triggered by time or events in a business application. Agents can be run on Domino servers or Lotus Notes clients.

Development Environment

Domino Designer is general-purpose client software featuring an integrated development environment (IDE) that provides easy access to all features of the Domino server.

Domino Object Model

Domino offers a unified model for accessing its objects through back-end classes, whether you use LotusScript® or Java. This allows you to switch programming languages without having to learn new ways to program for Domino.

Live Integration with Enterprise Data

DECS (Domino Enterprise Connection Services) is part of the Domino Server. It is a Lotus-developed technology, first shipped with NotesPump™ 2.5, that supplies an easy-to-use, forms-based interface to achieve deep, integrated connectivity to external data from Domino applications. This allows developers to map fields in forms directly to fields in relational database tables, without storing any data within the Domino database.

Scalability and Reliability

Domino Enterprise Server enables you to cluster up to six Domino servers to provide both scalability and failover protection, to maximize the availability of your groupware and messaging applications. Real-time replication technology keeps the clustered servers synchronized.

Note A Domino server is not the same as a file server. A file server provides access to shared resources such as printers and applications, and also manages network activity. Domino is an application-level server process that provides services necessary for the effective management of communications and applications.

Clients for Domino R5.0

Previous versions of Lotus Domino had one, all-purpose client that would be used by users, administrators, and application developers. With Lotus Domino Release 4.6, a special client for developers called Lotus Notes Designer for Domino was introduced.

As a result of the strong focus on ease of use in the design of Lotus Domino R5.0, three individual clients are now available. They are:

- Notes R5.0: the user's client
- Domino Administrator R5.0: the administrator's client
- Domino Designer R5.0: the developer's client

Most of the functionality in Lotus Domino can also be accessed from Web browsers. The Lotus Domino server includes a Web administration application. We will give a brief overview of the three clients below.

Notes R5.0

Lotus Notes is the leading integrated e-mail and collaborative software for the Internet. In R5.0, Notes offers a more open, Web-like, customizable environment, so you can work the way you want, with all the power you expect from Notes.

The new Navigation Bar gives you instant forward, back, stop, and refresh actions, as well as access to search engines and the Web, from wherever you are in Notes.

Notes R5.0 has Bookmarks so that you can create links to Web pages, application views, documents, and forms for instant access. The new Window Tabs allow you to keep track of multiple open windows, and navigate between them quickly. Notes R5.0 also has enhanced search capabilities, including search-by-form, fuzzy search, and the ability to perform a domain search — making information tracking quick and intuitive.

Headlines

With Notes R5.0, keeping on top of the latest and most important information is easy. The Notes R5.0 customizable Headlines page lets you select the information that you want to see first. You'll be alerted to important e-mail messages, tasks, or meetings for the day. You can even receive updates from intranet applications and view Web content dynamically — all from Headlines.

Each item on the Headlines page is a point of entry, so if you've received an urgent e-mail message, the full document is just a mouse click away. Plus, IT organizations can customize Headlines to feed corporate intranet information right onto the user's desktop.

Enhanced E-mail and Calendaring

The new mail and calendaring features in Notes R5.0 take the best of industry-leading applications, such as cc:Mail and Lotus Organizer®, and make them better. Notes R5.0 continues to build on its powerful integration by combining your mail and calendar preferences. You can preset preferences for every e-mail you send, including automatic spell check and sending all mail high priority with a return receipt. Notes R5.0 mail also supports signature files, giving you a simple way to identify yourself and add pertinent information to every e-mail you send.

If you manage multiple calendars, Notes R5.0 now gives you the ability to view more than one calendar at a time. Choose to access multiple calendars for a "quick view" of who is available, or get more detail on another user's schedule if necessary. And when you need to take your calendar with you, Notes R5.0 gives you multiple print formats to choose from. Notes R5.0 streamlines the process of managing resources across domains, giving you greater access and control over conference rooms, AV equipment, and more.

Installation and Setup

Setting up Notes R5.0 is easy. Integration with dial-up networking entries means connections are created automatically for you as you install the software. Notes R5.0 also offers several preset configurations for even faster user setup. It's easier than ever for you to access your ISP mail account right from Notes. And if you're upgrading, you'll be ready to go as soon as you finish installation.

A Powerful Tool for Any Infrastructure

Current Notes users can continue to take advantage of all their Domino server-based applications like e-mail and calendaring. In addition, Notes R5.0 offers full standards support including POP, IMAP4, SMTP, LDAP v3, MIME, S/MIME, HTML, Java, JavaScript, and X.509 certificates. So now, even users with non-Domino, standards-based back ends or those who use ISP hosted mail at home will benefit from the power of Notes R5.0.

Domino Administrator R5.0

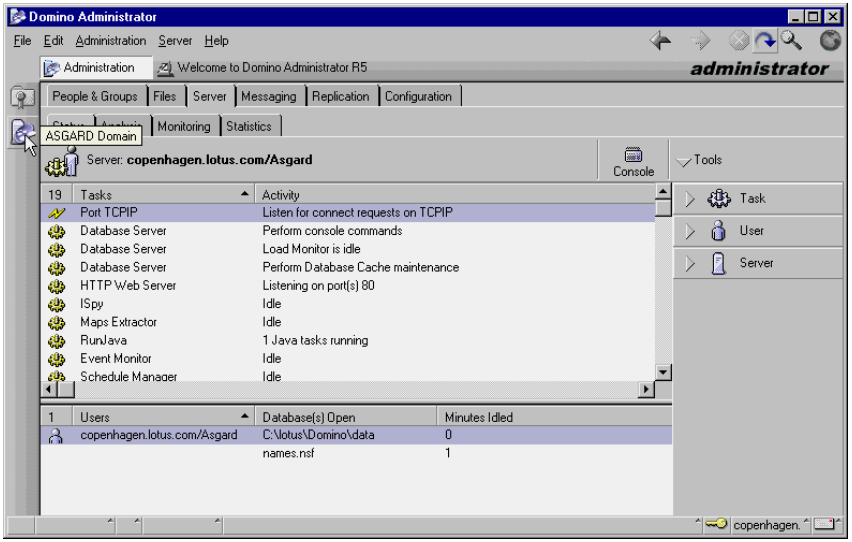
Domino Administrator R5.0 is a new, integrated administration tool that provides simple, yet flexible administration. Administration benefits are universal whether you are a small company just getting started with Domino, or an enterprise managing a large-scale deployment, with thousands of people and applications.

The Domino Administrator R5.0 utilizes the Windows Explorer metaphor, providing an easy, intuitive interface and allowing drag-and-drop functionality for common administration tasks, such as moving a user. Important new server monitoring features now allow administrators to proactively monitor and manage an environment. Finally, administrators have the ability to centrally configure, manage, and enforce user desktop settings. All of these administration enhancements, and more, result in the most comprehensive server management tools and reduce the cost of ownership.

Domino Directories Administration Tab

Domino Administrator provides logical groupings for administration functions and tasks via five specific interfaces reached via tabs across the top of the Administrator UI. These tabs are: People & Groups, Files, Server, Messaging, Replication, and Configuration. On each tab, the UI is divided into three primary work areas or "panes."

- On the left, the Server Scope Pane gives administrators a complete hierarchical view of your Domino Server deployment.
- The Context Pane object on the top gives administrators a view of the specific database, directory, group, server, etc. that you are working on.
- The Results Pane on the bottom gives administrators immediate feedback and results of tasks you invoke.
- The Toolbar along the right side provides context-specific administration tools (also available via right-mouse click).



The People & Groups Tab in the Domino Administrator provides a central interface for all user and group management, such as user registration, certification, and group management.

From the Files Tab, Administrators can easily manage files and applications. Context-sensitive tools let administrators easily perform common database tasks such as check the disk status, move, compact, and more.

From the Server Tab, Administrators can get a graphical representation of the state of their servers, with details on the current status of specific tasks.

Installation

Domino Administrator R5.0 is not a stand-alone client, but is included as optionally installable with Domino Server R5.0 and Domino Designer R5.0.

As a security administrator you will need Domino Administrator R5.0 for tasks like maintaining Domino server and database security.

Domino Designer R5.0

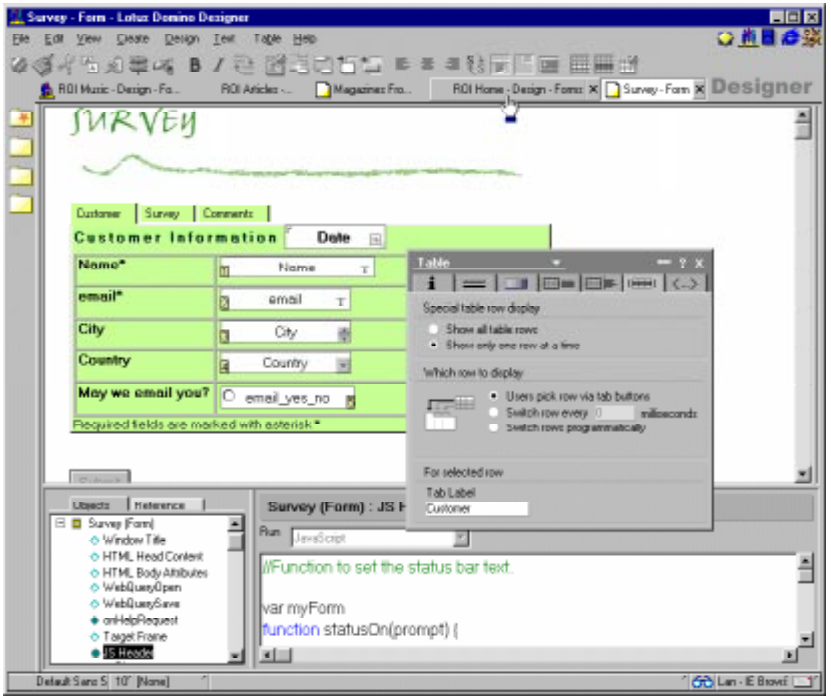
Lotus Domino Designer R5.0 is an open, integrated Web application development environment with everything you need to rapidly build applications that connect enterprise data with strategic processes. Domino Designer includes the visual tools you'd expect in a Web development environment, like HTML authoring, site design and navigation, and application preview. You can create or modify Domino applications using your choice of programming languages, including Java, JavaScript and LotusScript — all in one environment. Domino Designer delivers breakthrough innovations including:

- Support for Domino Enterprise Connection Services (DECS), a visual toolset for creating live connectivity to enterprise data.
- The Domino Object Model, which provides access to the built-in services of the Lotus Domino Web Application Server, from messaging to security to workflow.

Domino Designer R5.0 includes a set of new features and tools for rapid application development:

- Outline Designer: The outline designer is a visual toolset for simplifying Domino Web site design. It enables developers to design an entire site, link content to the site design, manage the links, and create a UI site navigation map component that can be used in site frames or on Web pages.
- Frameset Designer: The frameset designer provides visual tools and wizards to easily create multi-paned interfaces for Domino applications.

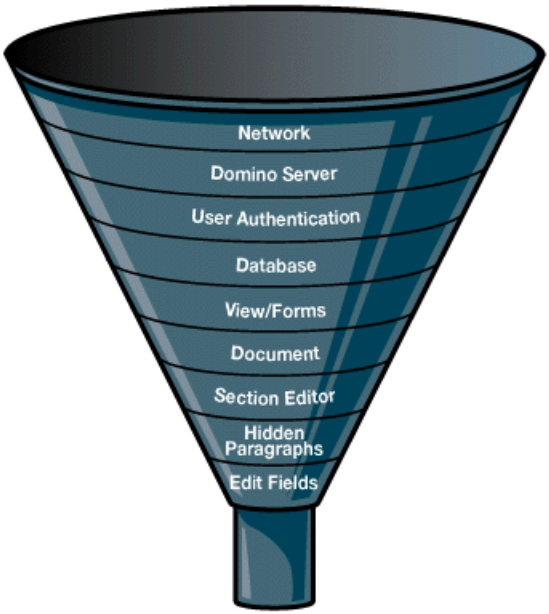
- Page Designer: Page designer, a visual HTML authoring tool, supports a broad range of browser technologies. A number of improvements to tables and graphics file support provide complete control over page design and layout.
- Domino UI Applets: Three popular Notes user interface components are now available as Java applets. The Java applets provide the capability to quickly add these full-featured Notes design elements to browser applications.
- Programmer's Pane Enhancements: An improved programmer's pane provides a consistent programming environment regardless of the script or language used. The programmer's pane includes tools to better access, use, and reuse objects.
- New Rapid Development Capabilities in IDE: Domino Designer now enables multiple work sessions to be open within tiled windows, provides a "movable" properties box for rapid manipulation of an object's properties, and offers a new Design Synopsis that provides access to all information about your application, including application source code and administrative information.



Domino offers you a lot of features, services, facilities and tools; but foremost, it offers you a comprehensive security architecture, which we will introduce in the next section and discuss in further detail in the rest of this redbook.

Levels of Security Offered by Notes and Domino

When talking about security in Notes and Domino we can look at it as nine layers from the outside in. The upper layer is simply getting access to the network where the Domino server is placed. The lowest layer involves security at the field level within a Domino document. The figure below illustrates how access gets more restrictive the further down you go.



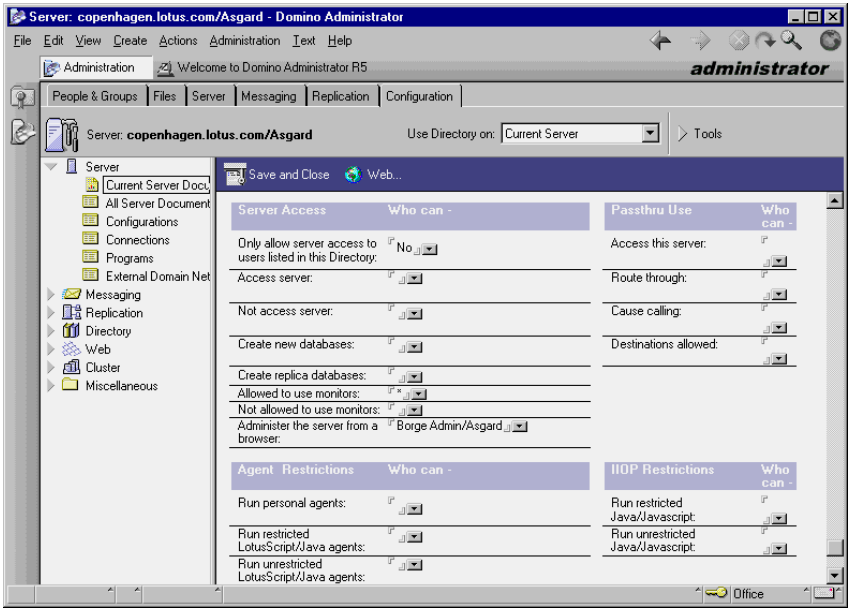
This redbook will focus on the upper three layers — that is, security in Notes and Domino from an infrastructure perspective. However, to give you a complete picture, we will briefly describe the other levels of access control as well. If you are interested in further information about security in Notes and Domino on the application level, please refer to the chapter, "Securing Your Application" in the redbook, *Lotus Domino R5.0: A Developer's Handbook*, IBM form number SG24-5331, Lotus part number CT6HPIE.

Note The figure above comes from the article "Designing a secure Domino app," which resides in "Iris Today," the technical Webzine located on the **`http://www.notes.net`** Web site produced by Iris Associates, the developers of Domino/Notes. Please visit Notes.net to obtain both the complete version of the article as well as other related articles.

Server Security

Server documents in the Domino Directory are used to control access to a Domino server. The implementation of security does not come for free, of course.

Applying server access restrictions will activate an additional security code that uses server processing cycles and can increase the time taken for a user to gain access to the server. Access restrictions are, however, an effective place to start with security controls. Using the restriction fields in the server documents you can quickly restrict several types of access to the server. You update the server access control parameters through the Domino Administrator. The server document can be selected from the navigator portion of the screen when you have selected the configuration tab in Domino Administrator as shown in the figure below.



Refer to the Domino Administrator documentation for an explanation of all the various security settings you can specify in the server document.

Data Access Security

After users or other servers gain access to a server, they will want some level of access to the data held by that server. The database layer of the Domino security model and the layers below that deal with data access, each layer providing more granular control than its predecessor. In fact, the access controls mirror the way that Domino stores and presents data:

- Database access

  At the heart of the system lie Domino databases. The records in a database are actually documents that have usually been entered by a user or administrator. Database access control facilities, therefore, provide the broadest control over who can do what to data on a Domino server.

- Form access

  Documents are not free-form text, but are in fact filled-in forms. When designing forms, you can use form access controls to specify who has access to the contents of a database in more detail than you can by using the database access controls.

- Document access

  Once a form has been filled in to create a document, the owner of the document can further restrict access to it. To what degree this is allowed depends on controls within the form from which the document is created.

- Section access

  Many documents contain data of varying sensitivity. In practice this means that you want to prevent certain users from updating or possibly even from reading parts of the document, but you want other users to have full access.

  One way to achieve this is to divide the form into sections and apply section access controls to it.

- Field access

  This is the most granular form of data access control. It allows you to control access to individual fields on a form or document. In addition to specifying user access, field access controls can limit the treatment that data receives when it is transmitted or stored.

Let us now consider each of these access control layers in more detail.

Database Access Control

Every database includes an Access Control List (ACL) which Domino uses to determine the level of access that users and servers have to that database. When a user opens a database, Domino classifies the user according to an access level that determines privileges. The access level for a user may vary in different databases.

The access level assigned to a user determines the tasks that the user can perform in the database. The access level assigned to a server determines what information the server can replicate within a particular database.

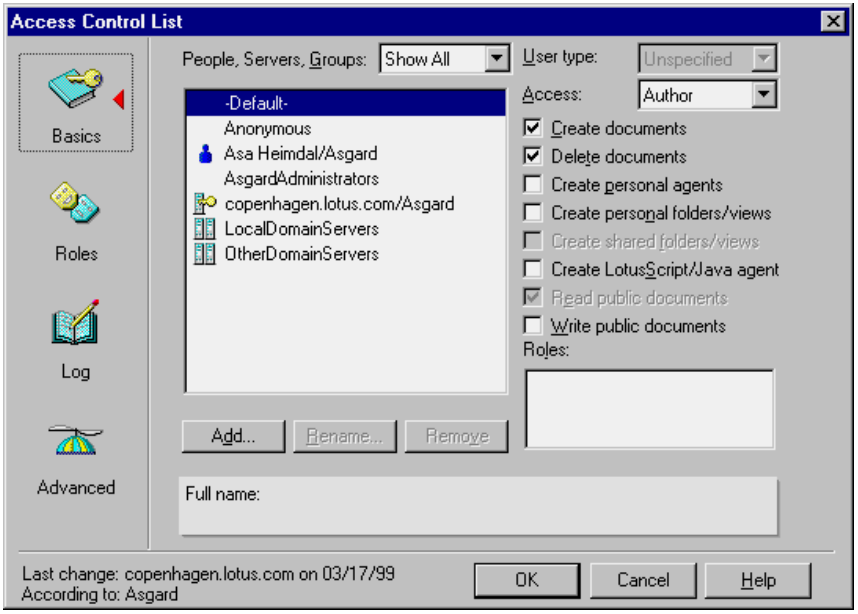
Only someone with Manager access can create or modify the ACL of a database located on a server.

Displaying the ACL

The access control list of a database shows all the servers, groups, and users who have access to the database.

To display the access control list of a database:

Choose File - Database - Access Control and the following panel will be displayed:



User and Server Access Levels

A database ACL determines the level of access that users, groups, and servers have. Someone with Manager access to the database assigns levels to the users, groups, and servers listed in the ACL.

With Domino Release 5.0 there are seven main levels of access that a database administrator can assign to a person, server, or group:

Level	*Users with this access can…*	*Servers with this access can…*
No Access	Not access the database at all.	Not access the replica at all.
Depositor	Create documents, but cannot read, edit, or delete documents, including those they create.	Not receive changes; not relevant for servers.
Reader	Read documents, but cannot create, edit, or delete them.	Pull changes from the replica but not send changes to it.
Author	Create and read documents, but can only edit their own documents if they are listed in an Authors field on that document.	Replicate new documents.
Editor	Create, read, and edit all documents unless there are restrictions on specific documents.	Replicate all new and changed documents.
Designer	Have Editor access to documents, except where restrictions exist for specific documents, and they can modify the database design, but they cannot delete the database or modify the ACL.	Replicate design changes as well as all new and changed documents, but not ACL changes.
Manager	Perform all operations on the database, including modifying ACLs and deleting the database.	Replicate all changes to the database and the ACL.

Setting Up and Refining the ACL

When you set up the access control list, you can refine the access for users in several ways, beyond simply specifying an access level:

- **Select User Type to Specify Users, Groups, and Servers**

  When you enter users in the ACL, you can specify whether they are users, groups, or servers.

- **Access Options**

  Assigning access options allows you to further refine user access.

- **User Roles**

  Roles allow you to define responsibilities in the application and refine access rights to database elements.

Users, Groups, and Servers

A group is a list of users and/or servers which have something in common. Using a group helps simplify many administration tasks. For example:

- A group of users can be given access to a database in the ACL.
- A group of servers can be designated as permitted to replicate with a database.
- A group of users can be denied access to a resource.

Note Groups you specify in the ACL must be listed in the Domino Directory.

There are two default server groups in the ACL:

- LocalDomainServers are servers in the local domain.
- OtherDomainServers are servers in other domains. These are usually servers in other companies with whom users in your company need to communicate.

User Types

The ability to specify user types lets you clearly indicate whether a name is that of a person, server, or group. See the table below for descriptions of the available user types:

User type	*Assign for this type of user*	*Allows you to . . .*
Person	An individual user; this includes a user on a server workstation.	Control access for an individual user.
Server	A single server; this includes a server console, and server workstation.	Prevent someone from accessing the database from a Notes workstation using the server ID.
Server Group	A group of servers.	Identify a group of servers that will host replicas of the database.
Person Group	A group of individual users.	Grant the same access to all users in a group without listing each user name in the access control list.

continued

User type	*Assign for this type of user*	*Allows you to . . .*
Mixed Group	A group of servers and individual users.	Grant the same access to a group of users and servers.
Unspecified	In the Advanced Access Control List window, click Lookup User Types for "Unspecified Users." Notes looks up an unspecified user type in the Address Book.	If you leave type as Unspecified, Domino will not check whether the access is given to a user or a server.

Assigning User Types for Additional Security

Assigning user types can provide additional security. Specifying names in the ACL as a person, server, or server group prevents someone from either:

- Creating a group in the Domino Directory with the same name and adding his or her name to it to access the database through the group name.
- Accessing the database from a Notes workstation using the server ID.

Note Designating a name as a server or server group is not a foolproof security method. It is possible to create a Domino add-in program that gains access to the database from a workstation through the server ID, since the add-in program behaves like a server.

Access Options

When you add users and groups you can specify individual options that further refine user access. For each ACL entry, you can specify slightly different options:

Enable this option…	*To allow…*	*This option is assigned by default to…*
Create documents	Authors to create documents.	Managers, Designers, Editors, and Depositors
Delete documents	Managers, Designers, Editors, and Authors to delete documents. Authors can delete only documents they created.	No one
Create personal agents	Designers, Editors, Authors, or Readers to create personal agents.	Managers
Create private folders/views	Editors, Authors, and Readers to create personal folders and views in a database on a server.	Managers and Designers
Create shared folders/views	Editors to create shared folders and views.	Managers and Designers

continued

Enable this option…	*To allow…*	*This option is assigned by default to…*
Create LotusScript/ Java agents	Readers, Authors, Editors, and Designers to create LotusScript and Java agents.	Managers
Read public documents*	Users to read documents created with forms, and use views and folders, designated as "available for public access user."	Readers and above
Write public documents*	Users to create and modify documents with forms designated as "available for public access user."	Authors and above

* Enabling users to read and write public documents lets you give users with No Access or Depositor access the ability to access specific forms, views, and documents without giving them Reader or Author access in the database. Public documents are useful for calendar applications in which one user might delegate the ability to read or create appointments on his or her behalf to another user.

Anonymous Access to Databases

You can handle anonymous users in one of the following two ways:

- Define an anonymous entry in the ACL and specifically define access privileges for anonymous users.
- Allow anonymous users the same access as the Default entry in the ACL.

Note Any application that will be deployed on the Web should have an Anonymous entry in the ACL.

If you allow anonymous access to a server, you can still control access to databases. To control database access for anonymous users, follow these steps:

1. Add a user with the name Anonymous in the Add User dialog box of the ACL.
2. Click OK.
3. In the Access drop-down box, select either:
   - No Access to prevent access by anonymous users.
   - Reader to allow access to an information database.
   - Author to allow access to an interactive database.

Caution If the database ACL does not contain an Anonymous entry, all anonymous users receive the Default access.

To protect the databases from unregistered users, you can establish the Default as No Access. If Default access needs to be higher, create an Anonymous entry in the database ACL and grant it No Access.

When granting access to unauthenticated Web clients, you will want to grant anonymous users the least access that still allows them to use the database effectively. For example, you might grant anonymous users:

- Reader access for an information database
- Author access for an interactive database

Differentiating Default and Anonymous Access

If Anonymous is not listed in the ACL, Domino grants the user access based on the default database access level. This may be a higher access level than you want for anonymous users.

Access level definitions:

- Default: a user not specified in the ACL
- Anonymous: a user without a valid Notes ID for that organization

Roles in the ACL

When a group you want to add to the ACL does not exist in the Domino Directory, you may want to create a special group or role for users of the database. Roles let you define responsibilities in the application and further define access to database elements.

What Is a Role?

A role is a subset of the ACL that is controlled by the database manager. A role can be used anywhere that a group or user name can be used. Users and groups are assigned roles to refine access to particular views, forms, sections, or fields of a database. Instead of assigning access to a design element to users and groups, you assign access to the role.

Some advantages of using roles are that they:

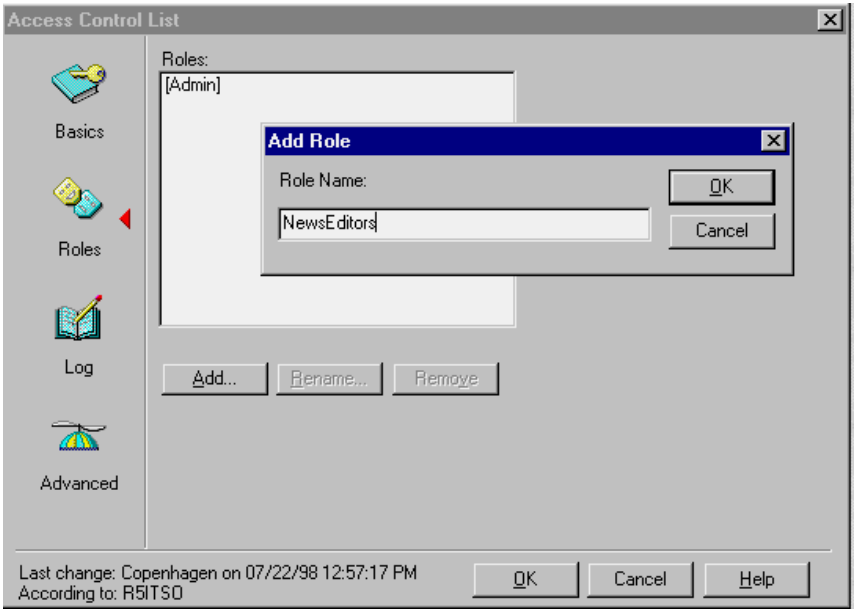
- Provide a flexible method of restricting document access to a specific set of users
- Can be used in formulas
- Provide group control if you do not have the authority to create groups in the Domino Directory, or if you want to create groups just for the database
- Make it easier for you to modify access when users leave or new users join

To use a role in an application, assign roles to users and groups in the ACL. Include the role in access lists, just as you do with users and groups (or actually *instead* of adding specific users and groups).

Adding Roles to the ACL

To add roles to an ACL, follow these steps:

1. Open the database ACL.
2. Click Roles in the Contents pane.
3. Click Add. The Add Role dialog box appears:



4. Enter a role name no longer than 15 characters and click OK. The role name appears in brackets in the Role list.

Assigning Roles to Users

To assign a role to a user:

1. Open the database ACL.
2. Select the user name in the list of people, servers, and groups.
3. Click one or more role names in the Roles list.
4. Confirm roles by highlighting a user. A checkmark appears next to the user role or roles.

Enforce Consistent ACL

You can ensure that the ACL of a database remains the same on all replicas. You do this by selecting the advanced access control list option "Enforce a consistent Access Control List across all replicas of this database." Selecting this option ensures not only that the ACL remains consistent across server replicas, but also that the ACL is enforced on replicas of the database made on workstations or laptops; if you do not select this option, users have Manager access to local replicas of server databases, which allows them to make changes to their access levels on the server replica, although they can't replicate such changes back to the server.

Enforcing a consistent access control list as it applies to ACLs on workstation or laptop replicas is not a security feature. Data in the local replica is not secure unless you physically secure the workstation or laptop or you encrypt the database using the local security feature. Also, a Domino add-in program can bypass an ACL enforced on local workstations.

To keep the ACL the same across all server replicas of a database, you must select this setting on a replica whose server has Manager access to the other replicas; otherwise replication will fail because the server has inadequate access to replicate the ACL.

Maximum Internet Name and Password Access

When working with advanced ACL options, you can also specify a maximum access level for users that have been authenticated with the Internet name and password setting (browser users). This setting overrides individual settings in the ACL. No browser user can get higher access than specified for Maximum Internet Name and Password Access.

Check this setting if you are experiencing problems with Web users not getting the access they have been granted in the ACL.

Form Access Control

Default read and create access to forms can be specified at the form level using the security section of the forms property box. Using the forms security will refine the database ACL, allowing more flexibility in database security designs. You can restrict people who have read access to the database from reading documents created with the form. In addition you can restrict people from creating documents with this form even though they have author (or above) access to the database. You can see in the table below which security options are available under form properties.

Option	*Use*
Default read access for documents created with this form	Allow only a subset of users in the database ACL to read documents created with a specific form. Documents created with this form will have this subset of users as the default document reader access list.
Who can create documents with this form	Allow only a subset of users with author access or above in a database ACL to use this form to create documents.
Default encryption keys	Encryption keys are generally created by database managers and distributed to the appropriate users. Specifying this will cause all encryptable fields to be encrypted when a document is saved with this form.
Disable printing/ forwarding/ copying to clipboard	This feature is an aid to users to keep them from accidentally including sensitive data when reproducing a document. It is not a very secure measure so it should not be relied on for true security.

Document Access Control

The creator of a document can determine who can read the document by using the security section of the document properties box. They can choose to allow all people with reader access or above to read the document or restrict read access to a limited number of people.

Documents inherit their read access property from the read access property in the form used to create the document. Anyone allowed to edit the document can change the document read access property.

Controlling Access to Documents Using Field-Based Access Controls

Each document can have special fields defined within it that control access to it. These fields are reader fields, author fields, and signed fields.

Reader Fields

Reader fields can be created on a form to allow read access restrictions to documents. A reader field consists of a list of names to be allowed read access to the document. It can be used instead of or in conjunction with form and document security read access lists. If both read access lists and reader fields are present, the users who can read the document are the addition of both lists. A reader field cannot allow access to a user that does not have read access to the database. It can, however, prevent read access from someone in the database ACL that could normally read the document.

Designating a field as a reader field and as editable allows the designer to choose options for presenting name lists to the author of the document. The field could present a list of names from the address dialogs, database ACL, a view dialog, or no list so that the author could manually enter names.

A useful way to use a reader field would be to create the field so the author of a document is presented with a list of people or groups to choose from, allowing the author to determine the reader list on a document-by-document basis.

Author Fields
Author fields can be created on a form to give users with author access edit capability to a document they didn't create. Author fields contain names of users and are created in the same way as reader fields. This is applicable only to users with author access to the database.

Signed Fields
Signed fields can be created on a form to allow a digital signature to be attached when a document is saved or mailed. Digital signatures verify that authors are who they say they are and guarantee that the data in the document has not been tampered with. The private key in a user ID file generates the signature. When a user opens the document containing the signed field, Notes verifies the signature by comparing it with the author's public key in the Public Address Book.

Restricting Access to Sections within a Document

Standard sections are used in a form to collapse or expand information. They can be hidden based on whether a document is in read or edit mode, or based on a formula. This will only hide a section from view. It does not protect it from update by agents, actions, or access from another form.

Access-controlled sections are used to group areas of a form and control edit access to objects in that area. However, like the standard sections, these fields may be edited from other forms, actions, or agents.

Layout regions are similar to sections. They are areas of grouped objects that can be easily moved and displayed in ways not available with forms and subforms. As with sections, they can be hidden based on whether a document is in read or edit mode, or based on a formula. This will only hide a layout region from view. It does not protect it from update by agents, actions, or access from another form.

Field Access Control

Database designers can design fields that can be encrypted with an encryption key. To decrypt and read the document, users must have the same key. Fields may also be protected during form design from update by authors after the initial document is created. Field property security options include an option specifying that a user must have at least editor access to use the field.

Summary

We've seen security basics in the first chapter, covered the basics of Notes and Domino in this chapter and are now ready to move to a more in-depth review of security features and facilities.

Chapter 3
Notes Public Key Infrastructure Revealed

In this chapter, we will describe the security infrastructure used by the "classical" side of Lotus Notes.

The base on which all Notes security is built is user authentication. Authentication is important because it allows you to differentiate one user from another; without it, you could not identify whether users are who they claim to be. It is, therefore, the key to providing restricted access to Notes resources.

The Notes authentication procedure depends on a certificate, an electronic stamp that indicates a trust relationship between the user and the server. The certificate is stored in a Notes ID file. Lotus Notes certification and authentication is a fairly complex process. Notes makes use of a number of cryptographic techniques, such as public key and symmetric key encryption, digital signatures, and public key certificates.

Confidentiality and integrity, as they apply to database replication and mail items in transit through the network, ensure that what arrives is identical to what was sent, and only the intended recipients can access the contents. All these features taken together constitute the security infrastructure of Notes.

Lotus Notes Certification Hierarchies

Lotus Notes authentication is based on *Notes certificates*, which are stored in Notes IDs. When a Lotus Notes user attempts to connect to a Lotus Domino server, the client and server present their certificates to each other. By examining certificates, the client will identify and authenticate the server, and the server will identify and authenticate the user.

Notes Certificates

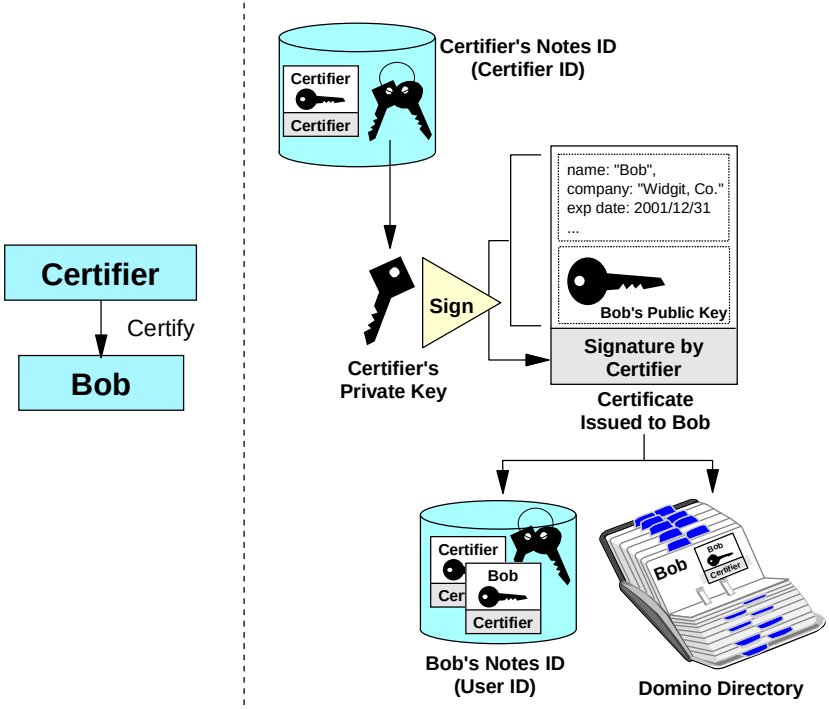
Casually speaking, a certificate is an electronic "stamp" which indicates a trust relationship among the entities in the Notes world. In actuality, a certificate is a digitally signed message added by a certifier to a Notes ID file.

The certificate contains:

- The certificate owner's name and details
- The certificate owner's public key
- The certifier's name and details
- The certifier's public key
- The certificate expiration date

The certificate is then certified, that is, digitally signed by the certifier using the certifier's private key, in order to prove its authenticity.

The certificate is then stored in a Notes ID file and the Domino Directory. A Notes ID file is a database that stores Lotus Notes certificates and private/public key pairs. The certificates registered to the Domino Directory are referred to by all users/servers belonging to the Domino domain, when they attempt to encrypt or digitally sign mail messages or document data. Note that the certificate itself does not contain any private information; it is therefore open to the public and can be distributed anywhere.



Using X.509 Certificates

The Domino Server and Notes client in R5.0 adds support for X.509 v3 certificates as well. The Notes client has the ability to request a certificate from any certificate authority, including a Domino R5.0 certificate authority, and store the X.509 v3 certificate in the Notes ID file. Please see the next chapter for details on X.509 and related topics.

Certification Hierarchies

Lotus Notes used to provide two types of certification: flat and hierarchical. Organizations using flat names may use several certifier IDs. Each user ID and server ID can include separate certificates generated by each flat certifier ID. Organizations using hierarchical certification have one organization certifier and optionally up to four layers of organizational unit certifiers below.

Flat Certificates

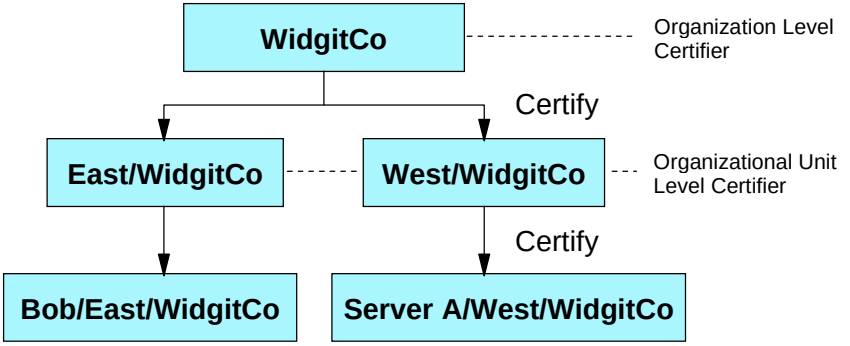
Lotus Notes/Domino R5.0 cannot create new flat IDs. However, flat ID maintenance is still supported by Notes/Domino R5.0 for compatibility with earlier versions. If your organization wants to use a flat ID with Lotus Notes/Domino R5.0, you must retain at least a prior version of Lotus Notes client to create a new flat ID.

New installations are encouraged to start with hierarchical names, and existing flat installations are encouraged to convert to hierarchical, because of the increased security and flexibility of access control, ID file generation and certification, and maintenance.

Hierarchical Certificates

In hierarchical certification, an organization may be layered with the organization certifier at the top and up to four layers of organizational unit certifiers below. When users or servers are registered with a certifier, they receive a certificate signed by that certifier and inherit the certification hierarchy of the layers above.

For example, consider the certification hierarchy shown in the figure below. This shows an organization named WidgitCo, subdivided into two organizational units, East and West. When Bob Smith shows up on his first day of work, the administrator of East/WidgitCo defines him as a new user. One of the results of this process is a new, randomly-generated, RSA private/public key pair. The administrator then creates a certificate for Bob, by signing his new public key using the East/WidgitCo certifier private key. As a result, Bob Smith's user ID inherits the certification hierarchy of the East/WidgitCo certifier.



Users and servers in the organization have fully distinguished names based on their certifiers. Each layer in the certification hierarchy inherits the fully distinguished name of the certifier used to create it and is in turn an ancestor to the layers below it. In this example, the organization level certifier WidgitCo has the fully distinguished name "O=WidgitCo," an organizational unit level certifier East has "OU=East/O=WidgitCo," and Bob has "CN=Bob/OU=East/O=WidgetCo."

Users and servers may authenticate with each other if they have at least one common ancestral certificate. In our example, this means that all users in the organization can authenticate with each other because they have the WidgitCo certifier in common. Entities that don't share at least one common ancestor can still authenticate by going through a cross-certification process.

Notes IDs and Domino Directory

Now let us see how this certification hierarchy is reflected in the Notes ID files and Domino Directory. When an administrator registers a user, he or she specifies the user name, password, expiration date, and other default options. The registration process creates an ID for the user or server and places it in the Domain Directory and/or in a file which must be given to the user to reside on the user's workstation.

User ID, Server ID, and Certifier ID

Essentially Notes ID stores certificates and encryption keys. There exist three different types of Notes ID: User ID, Server ID, and Certifier ID. User ID is the Notes ID for a Lotus Notes user, and Server ID is the Notes ID for a Lotus Domino server. Certifier ID is treated differently, because it is only used for certifying other Notes IDs. Therefore, it is much more important than other IDs for the organization. The Certifier ID should be stored on a floppy disk in a safe place. It should not reside on a server hard disk where unauthorized users may be able to access it.

Contents of Notes ID

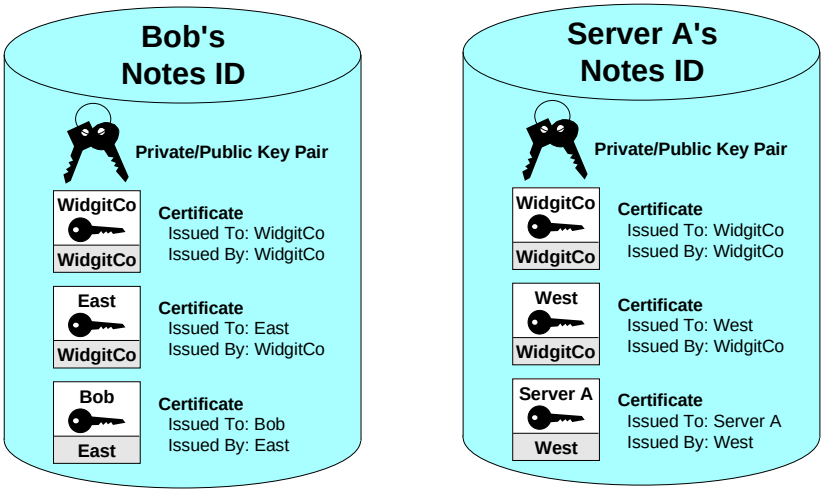
When an administrator attempts to register a new user or server, the Lotus Domino Administrator workstation generates two RSA key pairs for that entity. One, 512-bit key length pair, is used for data encryption in non-North American countries. Another, 630-bit key length pair, is used for data encryption within US and Canada, and for signature and authentication worldwide. The Domino Administrator then builds a certificate using the certifier's private key to sign the certificate. The signed certificate is then placed in the Notes ID file.

After the registration process, the ID file contains:

- The user's name and Notes license number
- Two public and private key pairs
- Two certificates for the user
- A certificate for each ancestor certifier

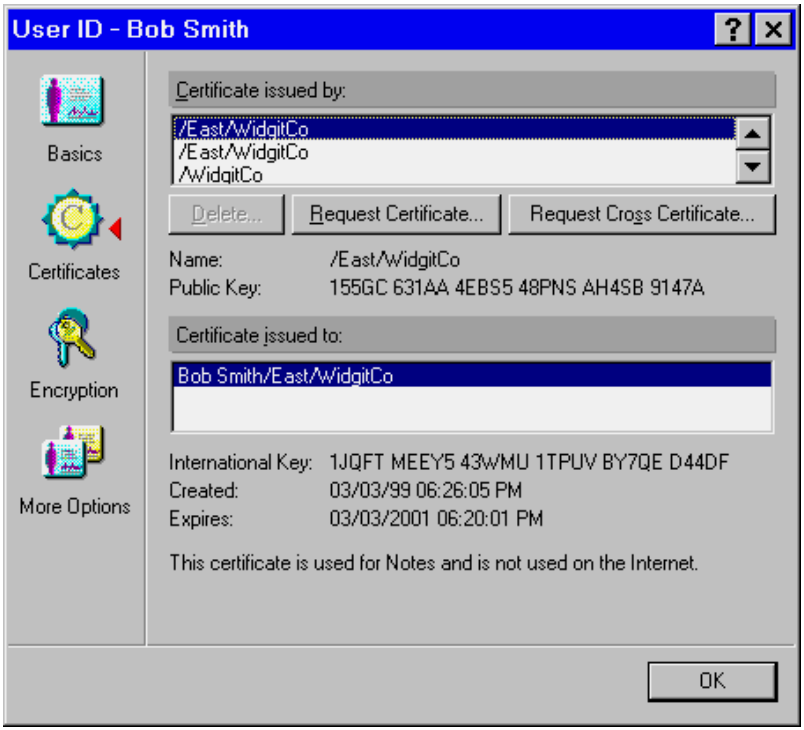
After the registration, encryption keys distributed by application developers may be added to allow encryption and decryption of fields in documents. The private key and the encryption keys in the ID file are encrypted using a key computed from the user's password, so that only the owner can access it. Public information such as the user's name and public key are not encrypted.

Again using the organization shown in the previous section as an example, the figure below shows the ID files for user Bob Smith in the East organization and for a server in the West organization.



Displaying the Contents of an ID File

Users can check the certificates available in their ID file by selecting File - Tools - User ID from the Notes client menu. The next figure shows one of the certificates from the ID file for Bob Smith.



Notice that the entries in the list at the top of the panel show the certifier, not the owner of the certificate. The entry labeled /WidgitCo is the certificate of the East certifier, issued by the top-level certifier in the domain. The two entries labeled /East/WidgitCo are two certificates for Bob Smith, both signed by the East certifier. The reason there are two is that one is an international key with a restricted key size, whereas the other is a full-strength North American key.

Alternate Naming

In Domino R5.0 it is possible to add an *alternate name* or alias by which the Notes entity may be referred to in the Notes ID file. This feature allows a user to be referenced by either their primary name or their alternate name. This may be desirable in an international organization where users are registered using a standard name format but prefer to be addressed by a more convenient name in their native country. The alternate name can then be used for mail addressing and in database ACLs. An alternate name like a

primary name, is hierarchical in format and must not be the same as any existing primary name or alternate name. During network authentication, both the primary name and the alternate name are authenticated. This means that users may be listed in ACLs, or in groups listed in ACLs, using either their primary name or their alternate name.

Alternate names are not compatible with Notes versions earlier than R5.0. In particular:

- Earlier versions of Notes/Domino servers are not able to authenticate with a user assigned an alias.
- Earlier versions of Notes/Domino servers and workstations are not able to validate a signature from a user assigned an alias.
- Earlier versions of Notes workstations are not able to use an ID file that contains an alternate name.

User Passwords

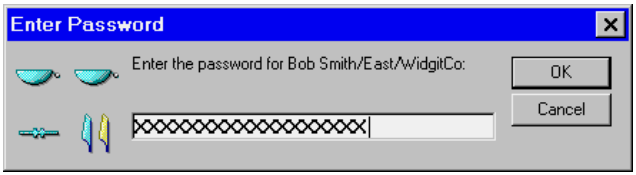
The password assigned to a user during registration is a mechanism to protect access to the Notes ID file. A Notes user attempting to use the ID file will be required to enter the password for that ID file. The password is normally used only to unlock the user ID file itself; the key pair contained in the ID is actually used to identify the user.

Users may have more than one copy of their ID file and different copies can have different passwords. This means that to change the password you must know the existing password for each copy of the ID. Even though Notes/Domino R5.0 has introduced a lost password recovery feature, we still strongly suggest that you back up your ID files and passwords.

Anti-Spoofing Password Dialog Box

To defeat dictionary or brute force attacks on ID file passwords and to reduce the risk of password capture, Notes employs an anti-spoofing password dialog box.

If users enter an incorrect password, Notes waits for several seconds before allowing them to try again. This delay increases with each incorrect attempt to a maximum of thirty seconds. The delay feature makes it difficult to try rapidly many passwords in succession in the hope of guessing the right combination. Also, the dialog box has a series of hieroglyphic symbols on the left side that change as users enter their password. The figure below shows the password dialog box with the hieroglyphic symbols on the left side.



These dynamic symbols make it more difficult to substitute a false dialog box that captures passwords in place of the Notes dialog box. Tell users to be alert to the symbols as they enter their passwords - if they notice that the symbols do not change or are not present, they should stop entering their password and click Cancel.

Multiple Passwords

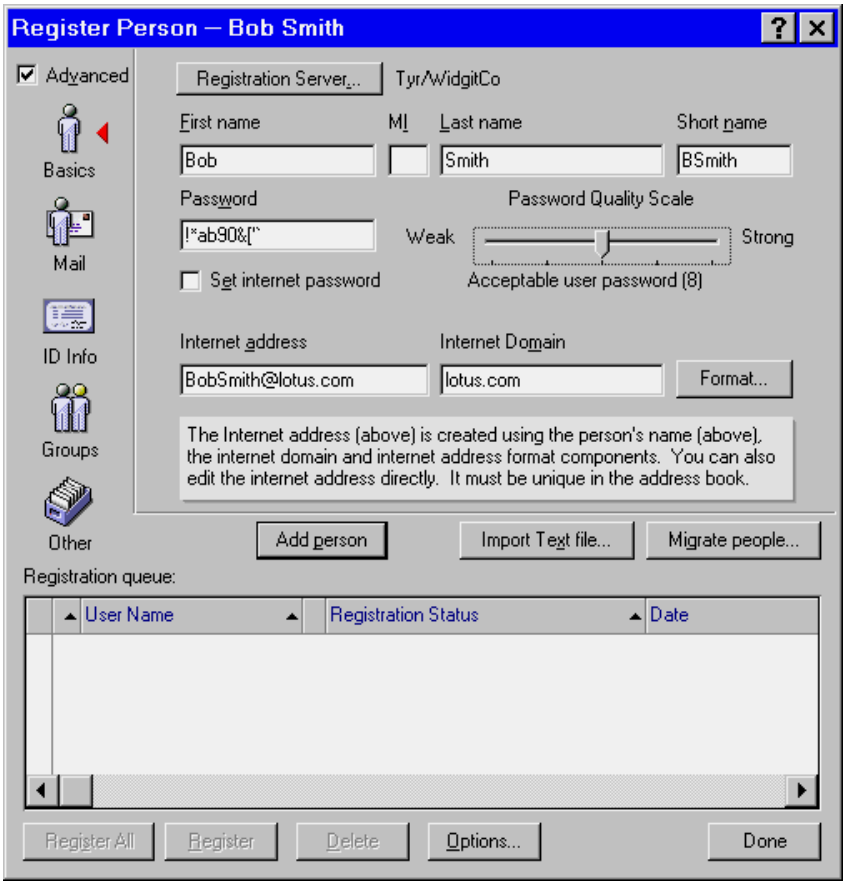
To provide tighter security for certifier and server ID files, you can assign multiple passwords to an existing ID. Doing this lets you require that more than one person act together when using the ID. You can specify that only a subset of the assigned passwords be required to access the ID. For example, you can assign four passwords to access the ID, but require only any two of the four passwords to access the ID. This feature is useful when you want to avoid giving authority for a certifier ID to one person.

Password Quality

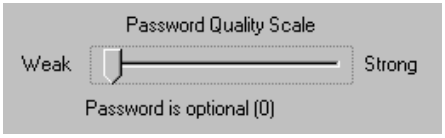
In previous releases of Notes and Domino, an administrator could, when creating or recertifying ID files, specify a minimum number of characters for passwords. However, not all passphrases of equal length are equal in strength; some are more vulnerable to passphrase guessing attacks than others. Unfortunately, choosing good passphrases can be difficult. A completely random collection of uppercase and lowercase alphabetic characters combined with numbers and punctuation marks (for example, “5P/#4Fwi=!”) would be ideal, but such a passphrase is not easily remembered and may need to be written down. In contrast, a passphrase consisting of one lone word (for example, “password”) provides little security.

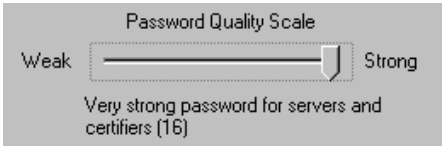
Mixed-case passphrases and passphrases containing numbers and punctuation are generally stronger per character than passwords consisting entirely of lowercase characters. Passwords that contain words found in the Notes spell check dictionaries are generally much weaker per character than any other kind of password.

Domino R5.0 includes a new feature that builds on the minimum password length feature. Administrators can now specify a password quality scale when registering a Notes user, as shown below.



Administrators can choose the quality scale from a range of 0 (weak) to 16 (strong). The higher the scale, the more complex the passphrase needs to be and the more difficult it is for an unauthorized user to guess the ID's passphrase.



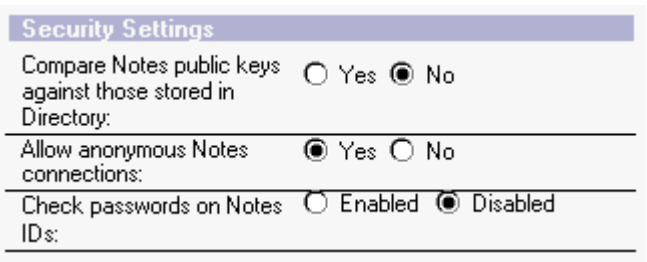


When a user changes his or her password using Domino R5.0, the quality of that password is estimated and then compared against the minimum password length specified for that ID file. If the password you specified is not sufficiently complex, your attempt is rejected and you will see the error message "Your Password is Insufficiently Complex."

The quality scales are stored in the ID file as equivalent password lengths, so that an ID file created by a Notes R5.0 client can be used in a previous release of Notes. ID files created in a previous release of Notes can also benefit from the password quality checking when used with R5.0. A random lowercase alphabetic password will receive equivalent ratings under both R4.x and R5.0; passwords containing words may pass the minimum password check under R4.x of Notes, but not pass a password quality check using R5.0. Complex and difficult to guess passwords may pass a password quality check using R5.0, but not pass the minimum character check using R4.x.

Password Checking

Starting with R4.5, Notes added a *password checking* process to the server. When password checking is enabled, information dependent on the user's password and the date the password was provided is kept on the server in the Person document. The user must enter the password corresponding to the information stored in the Person document to gain access to the server. The password checking facility adds the capability to require user password change intervals and to keep the previous 50 old passwords from being reused.



Lotus Notes is using the RSA key pair for authentication, so even if someone guessed the user's password they would still need to steal the user ID file to be able to impersonate the user. The information stored in the Domino Directory is not subject to dictionary attacks unless the attacker also has the ID file.

Password checking during authentication requires that both Notes clients and Domino servers run R4.5 or later. If you enable password checking on a server running a release prior to R4.5, authentication occurs without password checking. If you enable password checking on a client running

a previous release, authentication fails when the client attempts to connect to a server that requires password checking. The first time a user for whom password checking is required authenticates with a server, the User ID is altered and it cannot be used with a previous release.

ID File and Password Recovery

R5.0 includes a new feature that allows administrators to recover an ID file if a user loses, damages, or forgets the password for the ID. ID file and password recovery is a mechanism that allows a quorum of authorized administrators at a Notes site to gain access to the ID files of users within their domain. Recovery information is stored inside each ID file; encrypted backup copies of each ID file, that do not expose any private information, user passwords or bulk keys, are stored in a centralized location.

The certifier for a site can choose up to eight *Recovery Authorities* (RAs), those who are authorized for ID file recovery, and require between one and all of the Recovery Authorities to work together to access an ID file. For example, a site could be configured with five Recovery Authorities, three of whom are needed to unlock any given ID file. No single Recovery Authority could illicitly gain access to ID files so employee and job turnover would not lead to a breach of security. User passwords, which could be used to attack other accounts outside of the Domino servers, are not exposed. Any recovery ID file will not have the same password as the original ID file. Therefore, an attacker would have difficulty using a recovered ID file without the legitimate user noticing a loss of service on servers that have password checking enabled.

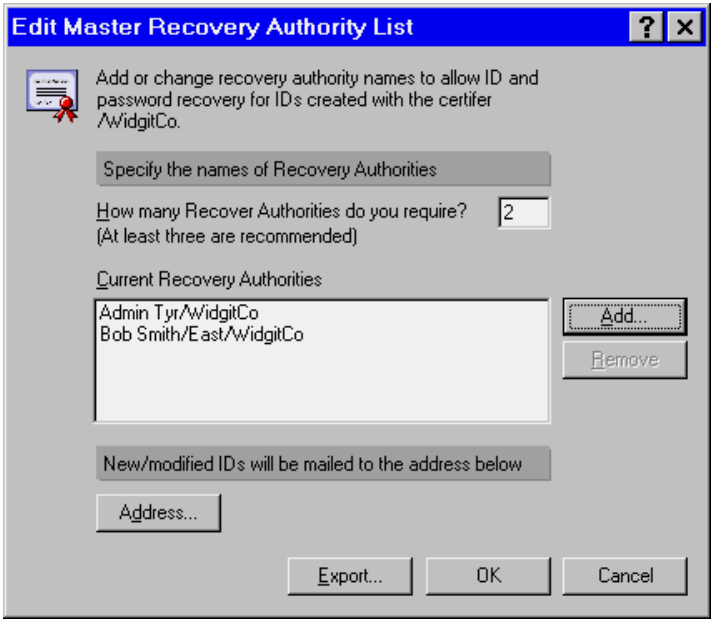
This feature can help users who have forgotten their passwords to gain access to their ID files, even if they are disconnected from the corporate network. Corrupted or lost ID files can be replaced as long as an out-of-band transmission channel for physical media (such as mailing a diskette) exists and corporations can gain access to the encryption keys used by employees who are no longer with the company.

Note The R5.0 password recovery feature replaces the “Escrow Agent” that exists in R4.x.

Setting Up ID File and Password Recovery

You should complete this process before registering users, because you can never recover IDs certified by a Certifier ID that do not contain recovery information.

1. To set up the ID file recovery, first you have to create a database on a server. This database will hold backup copies of Notes ID files. This "backup ID database" can also be set up as a mail-in database, so that Notes users can store a copy of their Notes ID via Notes mail. Any template can be used to create this database.
2. Configure the Access Control List of that database. The default must be No Access, and all Recovery Authorities must have at least Reader privilege. This ACL is the key to protecting the backup copies of IDs, therefore it is essential that you do this carefully.
3. Using Domino Administrator, define the recovery information in the Certifier ID. Open the Configuration Tab, select Tools - Certification - Edit Recovery Information. Enter the names of Recovery Authorities, and click OK. This information will be stored inside the Certifier ID.

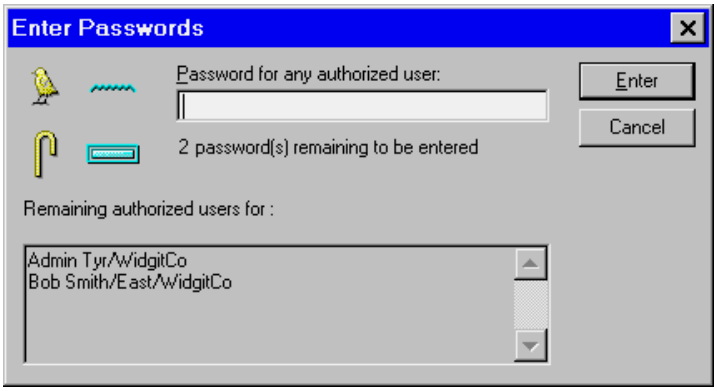


4. After registering users, you have to send the recovery information to them. To do this, click Export on the dialog box shown above, and supply the users' mail addresses.
5. The users will receive a Notes mail containing recovery information. Each user have to accept it, by choosing Action - Accept Recovery Information, and store the information in their own Notes ID file. At the same time, a backup of the recovery information will be sent to the backup ID database.

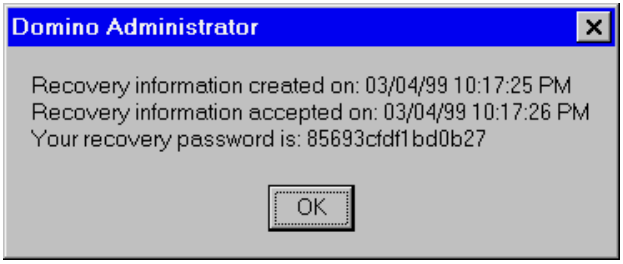
When an administrator registers a new Notes user after completing this operation, a backup copy of the Notes ID file is stored automatically in the backup ID database.

Performing Password Recovery

1. If a Notes ID file is lost or damaged, the RAs can retrieve the backup copy of the ID from the backup ID database. If the backup copy does not exist, you can never recover the ID.
2. To recover a password, the user selects File - Tools - Recover ID from the Notes client menu. You will be asked to enter several passwords. The user has to collect the series of passwords in order to unlock the ID file.



3. The user asks the RAs to extract the recovery password from that ID. The ID, issued by a Certifier ID which contains recovery information, also contains recovery information derived from the Certifier ID. Each RA uses Domino Administrator, and selects Configuration - Certification - Extract Recovery Password to show the recovery password assigned to them. Those passwords embedded in the recovery information are encrypted by each RA's public key, therefore each RA's Notes ID is required to decrypt the password.



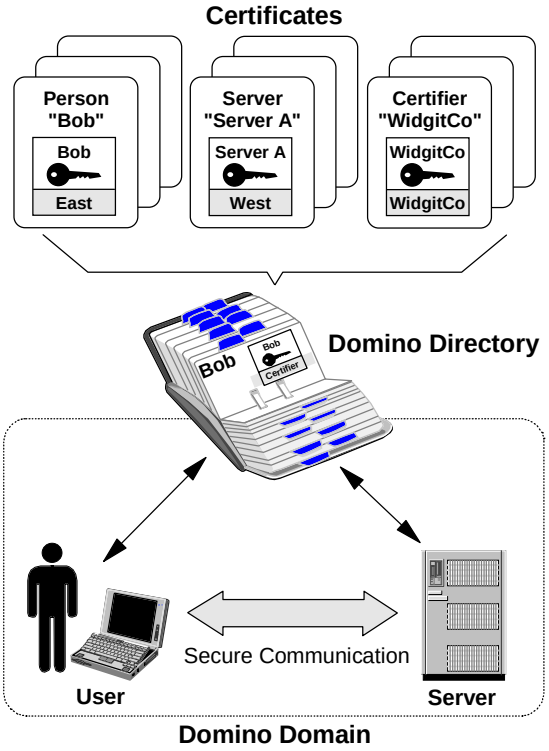
4. After collecting a series of recovery passwords, the user enters them in the dialog box. The user's ID will be unlocked and the user can change the password. That is, the lost password is recovered.

Domino Directory

Information about each ID is also maintained on the Domino server. The Domino Directory contains a Person document for each user, with a lot of information about the user including:

- The user's name and domain
- The user's public key/certificate
- An attachment with the user's ID file to be distributed the first time the user contacts the server, if it was chosen to store it here at user registration time

If a server has been registered, a Server document is put in the Domain Directory with similar information about the server. Certifiers are also represented in the Domino Directory by Server Certificate documents. The next figure illustrates how the Domino Directory works to maintain or distribute the certificates.

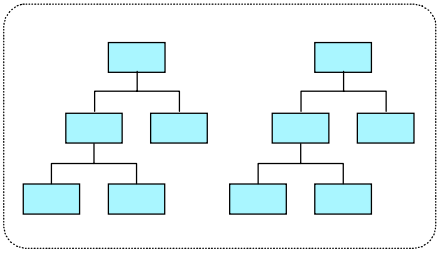


Domino Domain and Certification Hierarchies

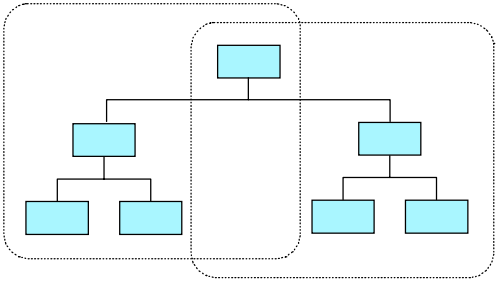
You may confuse the certification hierarchies with domains. However, they are totally independent of one another.

Briefly, a Domino domain corresponds to a Domino Directory. A Domino domain is a collection of Domino servers and users that share a common Domino Directory. A Domino Directory is a directory of users, servers, groups, and other entities. The primary function of the Domino domain is mail routing: users' domains are determined by the location of their server-based mail files.

Since certification hierarchies and domains are independent, you can manage two or more certification hierarchies within a Domino domain, as shown in the figure below. You may have to consider this kind of configuration when two organizations or companies merge, for example. You may also have to consider the cross-certification configuration between the certification hierarchies.



In addition, you can manage one certification hierarchy with several Domino domains as seen in the next figure. This is not a very likely scenario, however. With earlier releases of Lotus Notes you might consider this configuration in order to divide your domain into several domains. This could become necessary when the domain (or Name and Address Book) got so large that it affected performance.



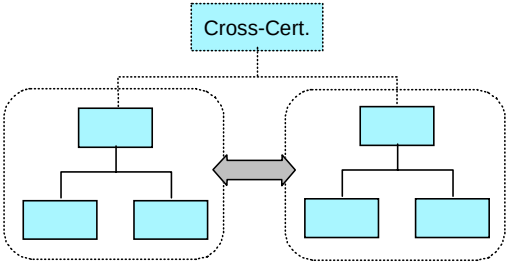
Cross-Certification

Under a certification hierarchies model, you cannot authenticate a server or a user that belongs to another naming tree. However, since company mergers occur frequently, how can we merge several certification trees satisfactorily? Even though several existing certification trees cannot be merged exactly, Notes provides a way to make it possible to communicate with other trees. This is achieved using *cross-certification*, a kind of peer-to-peer trust (certification) model.

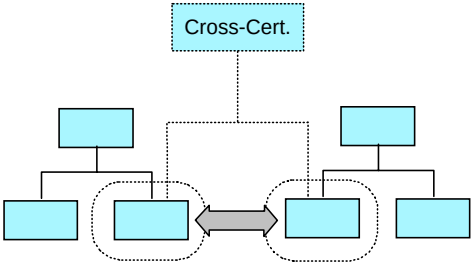
Cross-certificates are used to allow users and servers from the different hierarchically-certified organizations to access servers in other organizations. Each organization cross-certifies an ID from the other organization and then stores the cross-certificate it issues in the Personal Address Book or Domino Directory. For example, if Alan Jones/Sales/East/AcmeCo wants to access the Support/WidgitCo server, he needs a cross-certificate from /WidgitCo, and the Support/WidgitCo server needs a cross-certificate for /Sales/East/AcmeCo. When Alan tries to authenticate with the server, it checks for the cross-certificate. If the server finds a valid cross-certificate, it then checks whether Alan is allowed to access the server.

You also use cross-certificates to verify the digital signature of a user from another organization. Note that to verify signatures, cross-certification is necessary in only one direction; the verifier needs a cross-certificate for the signer.

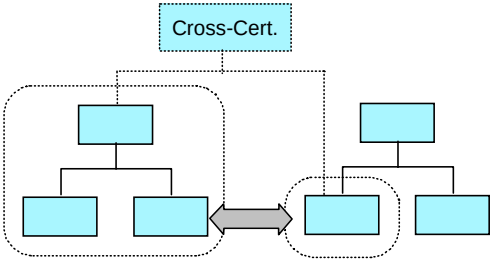
There are three style of cross-certification: between two organizations (or organizational units), between two users/servers, and between an organization and a user/server.



Organization - Organization

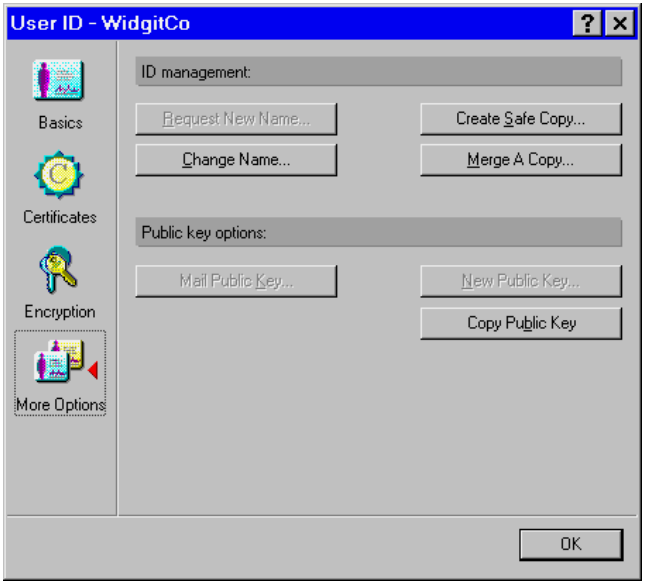


User / Server - User / Server



Organization - User / Server

To request a cross-certificate, first create a safe copy of the ID. Using Domino Administrator, select Configuration - Certification - ID Properties, choose the ID file to be cross-certified, go to More Options and click Create Safe Copy. Then you can bring the safe ID file to the other certifier.



To certify a cross-certification request, choose Configuration - Certification - Cross-Certify from Domino Administrator.

Please note that this operation results in a cross-certificate document. A user who attempts to connect to the other hierarchy must have this document in their Personal Address Book. The server being connected to must have a cross-certificate for the user or the user's organization in its Domino Directory.

Notes Authentication

In this section, we discuss how Lotus Notes and Domino authenticate each other in a session using Notes Remote Procedure Calls (NRPC). As we described already, Notes authentication is based on certificates, and it uses some public key based cryptographic techniques.

Please note that this authentication technique is similar to the one specified by X.509. Please refer to the following chapter for details on X.509-related techniques.

Validation and Authentication

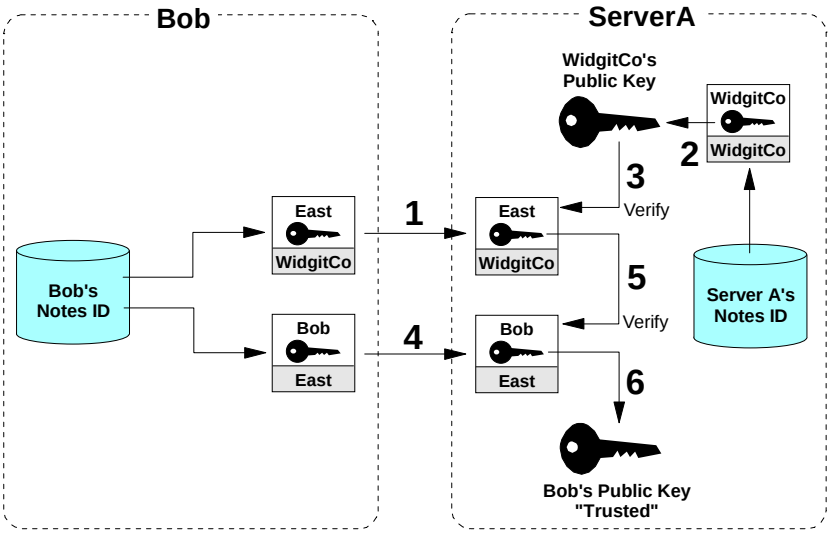
There are two phases in verifying a user's or server's identity in Notes. The first phase, called validation, is the process of reliably determining the sender's public key. In other words, the validation is the preparation phase for the actual authentication.

Notes uses the following rules when deciding to trust a public key:

1. Trust the public key of any of your ancestors in the hierarchical name tree because they are stored in your ID file.
2. Trust any public key obtained from a valid certificate issued by any of your ancestors in the hierarchical name tree.
3. Trust any public key certified by any trusted certifier and belonging to one of the certifier's descendants.

Phase 1: Validation

Let us now see how these rules are applied in the validation process. The user ID file for Bob Smith contains everything he needs to identify himself and establish his credentials. When he requests a session with a server the first step is to send to the server all of the certificates from the ID file (both the user's own certificate and the chain of certifiers' certificates that support it). The figure below illustrates the validation process that follows.



The numbered steps in the figure are described as follows:

1. ServerA reads the East certificate that Bob Smith sent from its ID file. This was signed by WidgitCo. ServerA is interested in it because East is the certifier of Bob's certificate.
2. ServerA reads the WidgitCo public key from its own ID file. (According to rule 1, ServerA will trust the public key of any ancestor that is stored in its ID file.)
3. ServerA uses the public key of WidgitCo (which is trusted because it is in the server's ID file) to verify that the certificate of East/WidgitCo is valid. (According to the rule 2, if you trust the public key of the ancestor, you will trust any public key obtained from certificates issued by the ancestor.)
4. ServerA reads the certificate that was sent from Bob Smith's ID file. This was signed by East.
5. ServerA uses the public key of East/WidgitCo, which now is trusted, to verify that the Bob Smith/East/WidgitCo certificate is valid. (According to rule 3, trust any public key certified by any trusted certifier and belonging to one of the certifier's descendants.)
6. ServerA has now reliably learned Bob Smith's public key.

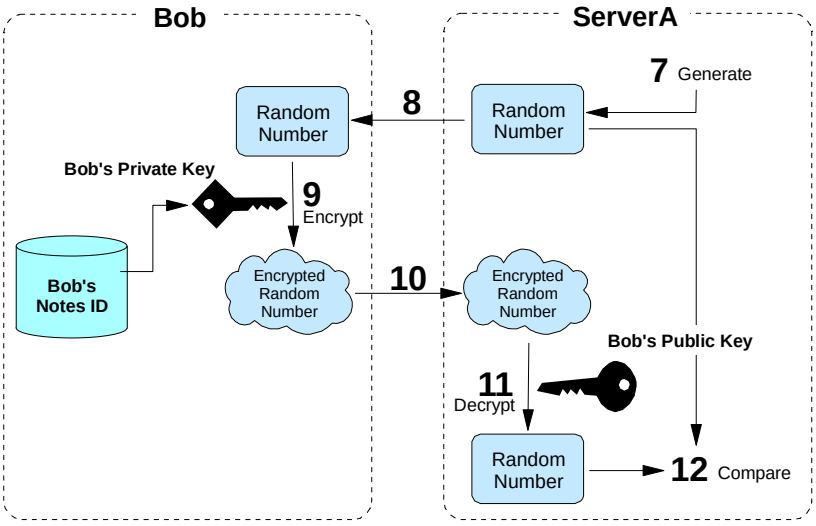
The same process is followed in reverse so that Bob can reliably learn ServerA's public key.

Phase 2: Authentication

After the validation process finishes the authentication process begins.

Authentication is a proof of identity. The validation process described above has not completely proved who each of the session partners is, because all they have presented so far is certificates. A certificate associates the user with a public key and tells the recipient that the public key can be trusted, but in order to prove that users really are who they claim to be they must show that they hold the private key that matches the public key in the certificate. The authentication process achieves this with a challenge/response dialog between a workstation and a server, or between two servers when either is running database replication or mail routing.

To continue the previous example of Bob Smith accessing ServerA, see the next figure below. The following is a simplification of the actual process, and is intended to illustrate what happens in a manner that's easy to understand.



1. ServerA generates a random number and a session key, and encrypts both with Bob's public key.
2. ServerA sends the encrypted random number to Bob Smith.
3. Bob receives the challenge and decrypts it with his private key.
4. Bob Smith sends back the decrypted number to ServerA.
5. ServerA compares Bob's response to the original random number.
6. If the result is the same as the original random number, ServerA can trust that Bob Smith really is who he claims to be.

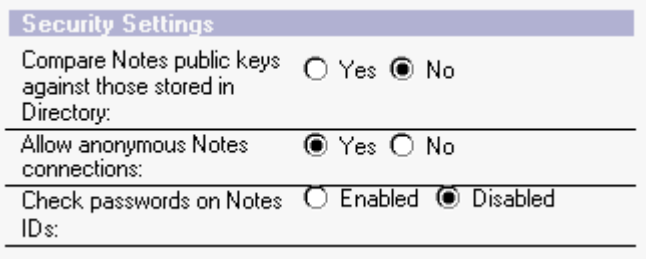
As with validation, authentication is a two-way procedure. Bob now authenticates ServerA using the same challenge/response process in reverse.

The actual algorithm is complex but efficient. It avoids any RSA operations on subsequent authentications between the same client-server pair. It also establishes a session key that can be used to optimally encrypt the messages that follow authentication.

Switching Off Certificate-Based Authentication

You can give Notes users anonymous access to a Domino server or Domino servers anonymous access to another Domino server. This lets users and servers access the server without that server authenticating them. This is most useful for providing the general public access to servers for which they are not cross-certified.

To allow anonymous access, open the Server document in the Domino Directory, select Security tab, and modify "Allow anonymous Notes connection" of the security settings as appropriate.



When you set up anonymous server access, Domino does not record the name of the person or server from the ID file, for example, in the Log file or in the User Activity dialog box.

If you are in a hierarchical certification environment and attempt to connect to a server which is set for anonymous access and the server can't authenticate you, you will see the following message in the status bar:

```
Server X cannot authenticate you because: the server's
Address Book does not contain any cross-certificates
capable of authenticating you. You are now accessing that
server anonymously.
```

Facilities for Data Integrity

When databases are replicated or mail messages are routed through the network, there is the risk they could be modified. We must be able to tell if the data that was received is the same as the data that was sent.

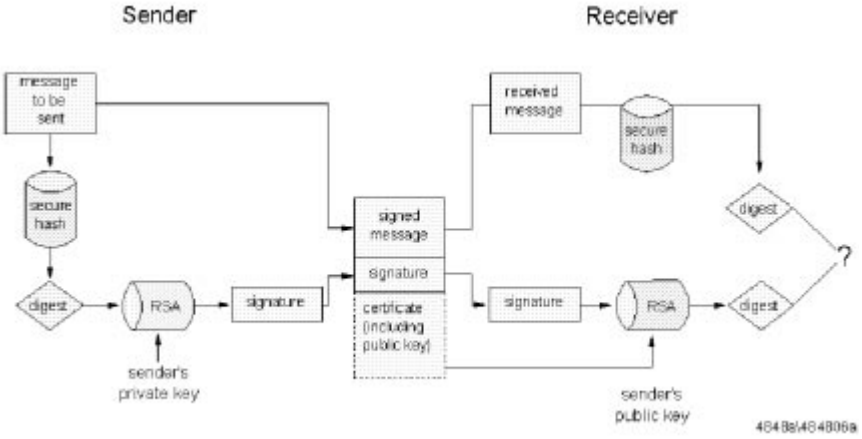
In order to detect any changes, we use digital signatures. Data integrity implies the current condition of the data is equal to the original "pure" condition. It guarantees that information is not changed in transit. A digital signature can verify that the person who originated the data is the author and that no one has tampered with the data.

Originators can add their digital signature to mail messages or fields, and to sections of Notes documents.

Note A database designer controls whether or not fields and sections of a database are signable; individual users can choose to sign mail messages.

Digital signatures use the same RSA key pair that was used in the verification and authentication process.

The figure below illustrates how Notes uses electronic signatures.



When a user adds a digital signature to a mail message, Notes uses a secure hash algorithm to generate a message digest (or "fingerprint") of the data being signed and encrypts the digest with the author's private key. This signature is attached to the data, along with the author's public key and certificates.

When the receiver accesses the signed data, Notes authenticates the sender's identity and decrypts the signed data using the public key in the user's certificate.

Notes indicates who signed the message if decryption of the signature is successful. Otherwise, Notes indicates that it cannot verify the signature. Two things are guaranteed by this signature process: the sender is authenticated (because the digest must have been encrypted in the sender's private key), and the message arrived unmodified (because the digests are identical). Otherwise the receiver knows the data has been tampered with or that the sender does not have a certificate trusted by the reader.

Facilities for Confidentiality

When you send data through the network, including mail messages, anyone who can intercept network packets, by tracing or electronic sniffing techniques, may read your data without authentication. This lack of privacy is a serious problem: it is likely that 90% of mail traffic is not sensitive, but to solve the problem you either have to persuade users to take security seriously or you have to treat all mail as sensitive and encrypt everything. Experience shows that upgrading computer systems is easier than modifying human nature, so often the latter approach is applied.

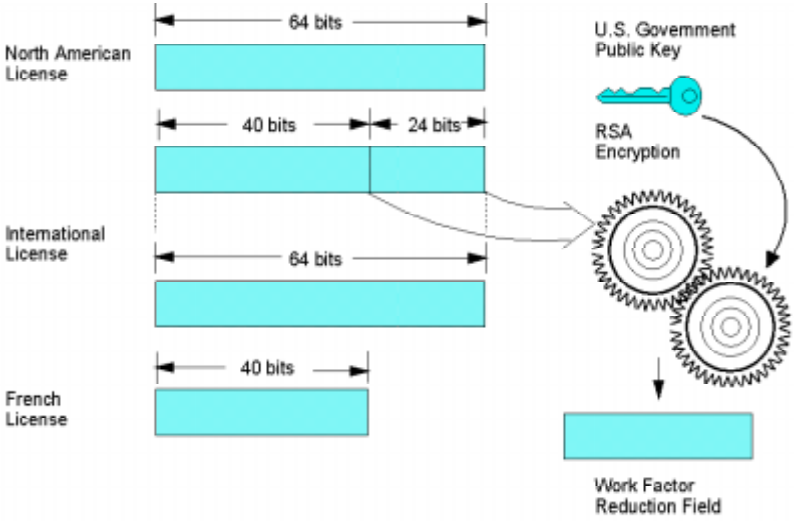
Fortunately, Notes provides a solution for either approach. Sensitive data can be encrypted into an unreadable format before transit. After it arrives at the destination, decryption can be used to read it. This method protects data from unauthorized access. Notes uses a bulk encryption mechanism, based on a secret key, to encrypt and decrypt data and to confirm that the data you received hasn't been read by others.

After authentication, the client begins to exchange data with the server. A lot of data will pass through the network, so it is important that the algorithm used be efficient. Notes uses the RC2 or RC4 algorithms for bulk encryption of data.

One variation is introduced by the type of Notes license a user has. There are three types of licenses in Notes: North American, International, and French. North American and International licenses are compatible in almost all aspects. Certification, authentication, electronic signatures, and communication between servers and users are the same except for one significant difference.

Because of US government export restrictions on encryption technology, a North American license uses a more secure encryption key than an International license. Notes will automatically figure out whether the sender or any of the recipients of a message is international and if so will use the less secure key.

The figure below illustrates the difference between a North American, International, and French license.



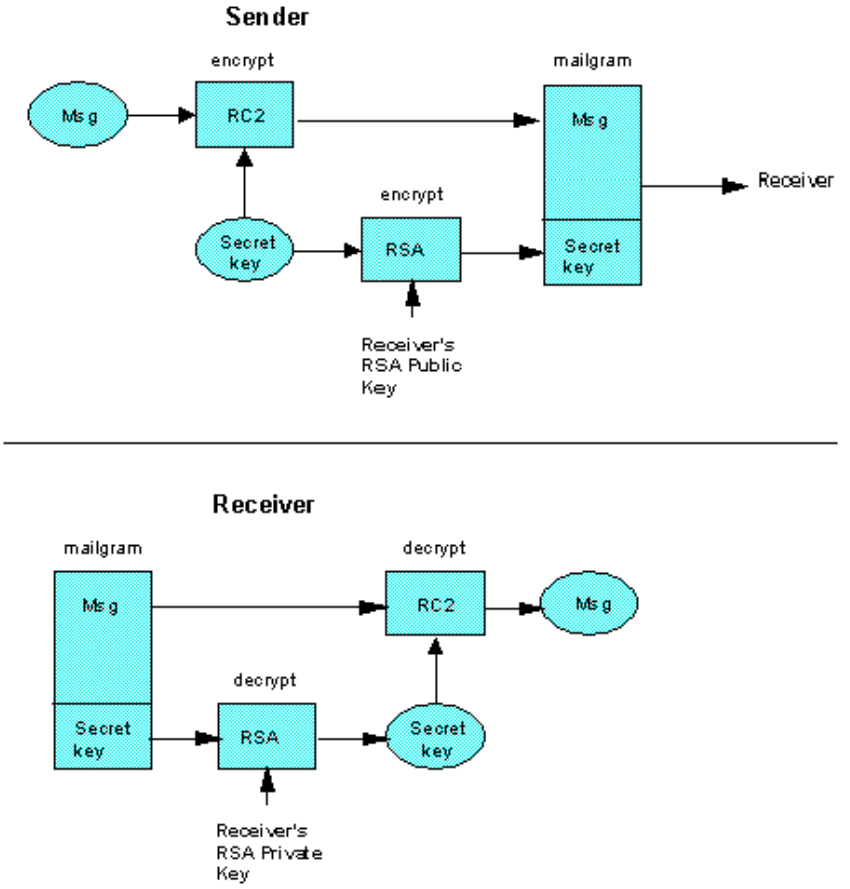
Both the North American and International versions of Notes use the full 64-bit key size. However, the International edition takes 24 bits of the key and encrypts it using an RSA public key for which the US government holds the matching private key. This encrypted portion of the key is then sent with each message as an additional field, the work factor reduction field. The net result of this is that attackers must break 64-bit encryption, which is at or beyond the practical limit for current decryption technology and hardware. The US government, on the other hand, only has to break a 40-bit key space, which is much easier (2 to the power of 24 times easier, to be precise).

The unique Notes encryption technique allows the International edition of Notes to use an encryption key equal in strength to the 64-bit key in the North American edition. This significantly increases international customers' information security, while neither increasing nor decreasing government access to encrypted information. Key length in the French version is limited to 40 bits due to restrictions imposed by French law. The three editions of Notes are fully interoperable.

Remember two things. If users are no longer able to use their ID file, either because they forgot their password or physically lost the file, any mail encrypted under their public key is permanently lost. The private key is located in the ID file. Also, anybody who has a copy of your private key can read your encrypted mail. Protect your private key carefully.

Mail Message Encryption

The following shows how Notes uses secret keys to encrypt and decrypt mail messages:



1. When the sender encrypts a message, Notes generates a random encryption key (the secret key) and encrypts the message with it. The random encryption key is itself encrypted with the recipient's public key and attached to the message. The RC2 algorithm is used to encrypt and decrypt the data.
2. After the mail with the encrypted key arrives, the recipient uses his private key to decrypt it. Secrecy is guaranteed, because only the recipient's private key can be used to decrypt the secret key needed to decrypt the message.
3. A different random key is generated and sent each time mail is sent.

Note S/MIME, a secure e-mail standard in the Internet, works very similarly to the secured Notes mail described here. For more information about S/MIME, please check the next chapter.

Other Notes Encryption Features

The above description applies to Notes mail, but Notes also provides other methods for encrypting information. Databases, documents, fields and transmission of data over the network can be protected using various methods of encryption:

- Databases can be encrypted with a user or server ID by using the Local Security option. This protects the database from being accessed by an unauthorized user who has gained access to the workstation the database is stored on, or who has made a copy of the database by using the operating system.
- Field encryption using special encryption keys created and distributed by the database designer can be used to limit access to fields by authorized users.
- Documents can be encrypted using private or public keys. Keys can be added to the form causing every document created with the form to be encrypted, or by letting users encrypt documents with their own encryption keys.
- Network port encryption allows unencrypted data to be encrypted at the port level for safe transport through the network. Network port encryption can be enabled for a user's workstation or at a server by selecting File - Preferences - Ports to modify the port definition to encrypt network data.

Summary

In this chapter we have shown how Notes and Domino security is built on a robust Public Key Infrastructure that allows for authentication, data integrity and confidentiality among Notes users. In the next chapter we will explore how this Public Key Infrastructure has been expanded to support security on the Internet using X509 certificates.

Chapter 4
Domino Internet Security Revealed

Today's Internet security is based on public-key X.509 certificate technology. However, as we described in the previous chapter, Lotus Notes has offered certificate-based security for many years.

In Notes and Domino R5.0 there is now an integrated infrastructure for the existing native Notes-certificates and the Internet X.509-based certificates.

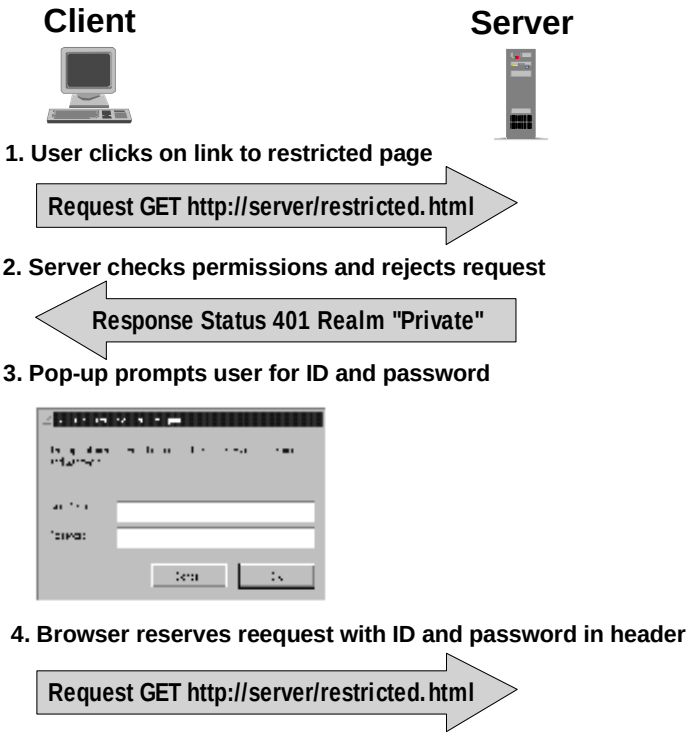
In this chapter we talk about the considerations for a security infrastructure where the need is to provide secure Web Browser and Internet messaging communication using Lotus Notes R5.0 and the Domino R5.0 Server.

HTTP Basic Authentication

Basic authentication employs user IDs and passwords. You have to configure the server to identify which parts of the document tree are protected. These zones of protection are known as realms. Each realm is associated with a set of user IDs and passwords that are allowed access. Realms can contain any kind of server object, such as CGI programs, as well as HTML pages.

How Basic Authentication Works

Basic authentication relies on a challenge mechanism to prompt users to authenticate themselves.



The figure above shows that when a client requests a URL, the server checks to see if the URL requires user authentication. If it does, the server rejects the request, with a status code of 401. The browser then pops up a dialog box on the user's screen, asking for a user ID and password. When the user has provided them, the browser resends the original request, but with the addition of the following MIME element within the HTTP header:

```
Authorization: Basic <userID and password block>
```

The user ID and password block is constructed by creating a string of the form userID:password, and then encoding it using the base64 algorithm.

You may wonder, given the above description, why you are not repeatedly prompted for a password every time you access a new restricted page. The reason is that the browser caches the user ID, password, server name, and realm name in memory, so that if it receives another 401 status code for the same server/realm combination it can reissue the request using the appropriate user ID and password. In fact, most browsers go one stage further than this and send a user ID and password for any URL that is likely to need it.

Netscape Navigator and Microsoft Internet Explorer sends the information with any URL that is in the same logical directory. For example, imagine that the user requests http://hostX/secret/foo.html and is prompted for a password with realm name PRIVATE. Netscape Navigator will store the details in memory. Later, if the user clicks on a link to URL http://hostX/secret/bar.html, the browser will send the user ID and password for realm PRIVATE in the request, without waiting for a 401 challenge from the server.

The objective of these tricks is to reduce network traffic and improve responsiveness, by eliminating a number of invalid requests and 401 status responses. They also, unfortunately, have the side effect of re-transmitting the user ID and password when it may not in fact be necessary.

Is Basic Authentication Secure?

There are two obvious loopholes in HTTP basic authentication:

- The user ID and password are included in the packet header, which means that they can be captured by anyone with a network sniffer or trace tool at any place in the session path.
- The user ID and password are cached in the browser, so if you leave the machine unattended, anyone can use your ID to access restricted information.

The second loophole is no different from any other situation where a machine is left unattended. The solution is one of user education: always lock the screen when not at your desk. Note that the caching is in memory, so the user information is lost once the Web browser has been shut down.

The first loophole is more significant. The user ID and password are not encrypted when they are placed in the packet header, but instead are encoded with base64. Base64 is an algorithm that forms part of the Multipurpose Internet Mail Extensions (MIME) protocol. It is a mechanism that turns any bit stream into printable ASCII characters. (It is described in RFC1521.) In fact, the objective of base64 is not for masking data at all, but to provide a method to send binary data through a mail gateway that can only handle character data.

The result of this is that by capturing the Authorization: Basic header from an HTTP request, an attacker can easily extract the user ID and password.

How serious is this exposure? Within a corporate network it may not be a big problem. In fact, base64 offers protection of user IDs and passwords that is superior to many older protocols that send them as clear text. In the Internet it is a different story. Here you have to assume that someone, somewhere is tracing everything you send. Clearly HTTP basic authentication should not be used as the sole method of protection for any critical resource.

So how can you solve the problem of sending confidential information like usernames, passwords, and credit details over the network securely? The solution is to ensure that the basic authentication, and subsequent communications are using an encrypted connection for it to operate in. SSL is a good example of a protocol that encapsulates HTTP data in this way. We describe the operation of SSL later in this chapter after a brief introduction to X.509 certificates.

X.509 Certificate

Although there have been several proposed formats for public key certificates, most commercial certificates available today are based on the international standard ITU-T Recommendation X.509 (formerly CCITT X.509).

X.509 certificates are used in secure Internet protocols like

- Secure Socket Layer (SSL)
- Secure Multi-purpose Internet Mail Extension (S/MIME)
- Secure Electronic Transaction (SET)

What Is the X.509 Standard?

Originally the X.509 standard was intended to specify the authentication service for X.500 directories. Directory authentication in X.509 can be done using either secret-key techniques or public-key techniques. The latter is based on public-key certificates. At present, the public-key certificate format defined in X.509 standard is widely used and supported by a number of protocols in the Internet world. X.509 standard does not specify a particular cryptographic algorithm; however, apparently RSA algorithm is the most broadly used one.

The initial version of X.509 was published in 1988. The public-key certificate format defined in this standard is called X.509 version 1 (X.509v1). When X.500 was revised in 1993, two more fields were added, resulting in the X.509 version 2 (X.509v2) format.

X.509 version 3 (X.509v3) was proposed in 1994. X.509v3 extends v2 in order to address some of the security concerns and limited flexibility that were issues in versions 1 and 2. The major difference between versions 2 and 3 is the addition of the extensions field. This field grants more flexibility as it can convey additional information beyond just the key and name binding. In June 1996, standardization of the basic v3 format was completed.

X.509 Certificate Content

An X.509 certificate consists of the following fields:

- Version of certificate format
- Certificate serial number
- Digital signature algorithm identifier (for issuer's digital signature)
- Issuer name (that is, the name of the Certification Authority)
- Validity period
- Subject (that is, user or server) name
- Subject public-key information: algorithm identifier and public-key value
- Issuer unique identifier - version 2 and 3 only (added by version 2)
- Subject unique identifier - version 2 and 3 only (added by version 2)
- Extensions - version 3 only (added by version 3)
- Digital signature by issuer on the above fields

Standard extensions include subject and issuer attributes, certification policy information, and key usage restrictions, among others.

After this introduction to X.509 certificates we will look at how they are used by Notes and Domino R5.0, first for SSL, and later in this chapter for S/MIME.

Secure Sockets Layer (SSL)

Overview

The SSL protocol was originally created by Netscape Inc., and is now widely implemented in most Internet-based client/server software. SSL makes use of a number of cryptographic techniques, such as public key and symmetric key encryption, digital signatures, and public key certificates.

SSL version 3.0, the current version, is a security protocol that:

- Encrypts information sent over the network from client and server. *(Confidentiality)*
- Validates that the message sent to a recipient was not tampered with. *(Data Integrity)*
- Authenticates the server, using RSA public key methods. (*Authentication)*
- Authenticates the client identity. *(Authentication - new in version 3.0)*

SSL version 3.0 also supports X.509 v3 certificate format, new key-exchange and encryption algorithms, and improved server performance through caching.

Normally SSL is considered as a way of securing HTTP traffic, however SSL has the advantage of being application protocol independent, which means that it can be run on top of any TCP/IP application protocol like (for example, NNTP, POP3, and LDAP). SSL connections use a different port number for each application protocol that it is encrypting, for example, when running SSL over HTTP, port 443 is used, as assigned by the Internet Assigned Numbers Authority.

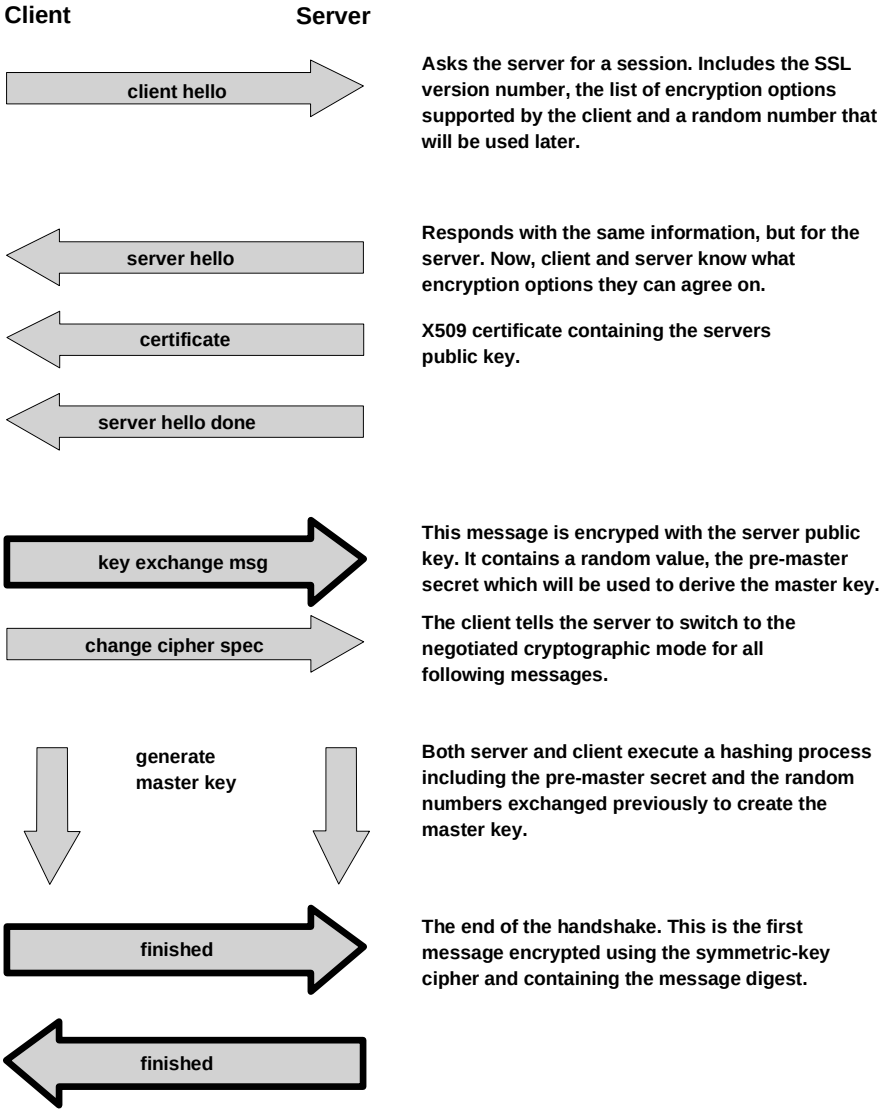
How SSL Operates

There are two parts to SSL:

- The *handshake*, in which the session partners introduce themselves and negotiate session characteristics.
- The *record protocol*, in which the session data is exchanged in an encrypted form.

The SSL Handshake

The following figure shows a simplified version of the SSL handshake process.



The two hello messages are used to exchange information about the capabilities of the client and server. This includes a list of cipher suites, combinations of crypto-algorithms and key sizes that the client and server will accept for the session.

Also the server provides a public key certificate. This is the method by which SSL checks the identity and authenticity of the session partner. In this example we only show the steps for server authentication, but if client authentication were required there would be another message exchange using the client public key.

Finally, the session partners separately generate an encryption key, the master key from which they derive the keys to use in the encrypted session that follows.

You can see from this example that there is significant additional overhead in starting up an SSL session compared with a normal HTTP connection. The protocol avoids some of this overhead by allowing the client and server to retain session key information and to resume that session without negotiating and authenticating a second time.

The SSL Record Protocol

Once the master key has been determined, the client and server can use it to encrypt application data. The SSL record protocol specifies a format for these messages. In general they include a message digest, using the MD5 algorithm, to ensure that they have not been altered and the whole message is then encrypted using a symmetric cipher.

Usually this uses the RC2 or RC4 algorithm, although DES, Triple-DES and IDEA are also supported by the specification.

Export Restrictions on Encryption Keys

The US National Security Agency (NSA), a department of the United States federal government, imposes restrictions on the size of the encryption key that may be used in software exported outside the US. Beginning about July, 1997, changes were made to these rules, whereby the US Government now allows the export of 128-bit encryption keys for any International banks and US companies with foreign offices.

Note VeriSign currently is authorized to issue these special certificates called *VeriSign Global Secure Server ID*. Please see "Applying for a Global Server ID" later in this chapter.

The RC2 and RC4 algorithms are able to achieve this by using a key in which all but 40 bits are set to a fixed value. International export versions of software products have this hobbled security built into them. SSL caters to mismatches between the export and non-export versions in the negotiation phase of the handshake.

For example, if a US browser tries to connect with SSL to an export server, they will agree on export-strength encryption.

SSL Deployment Considerations

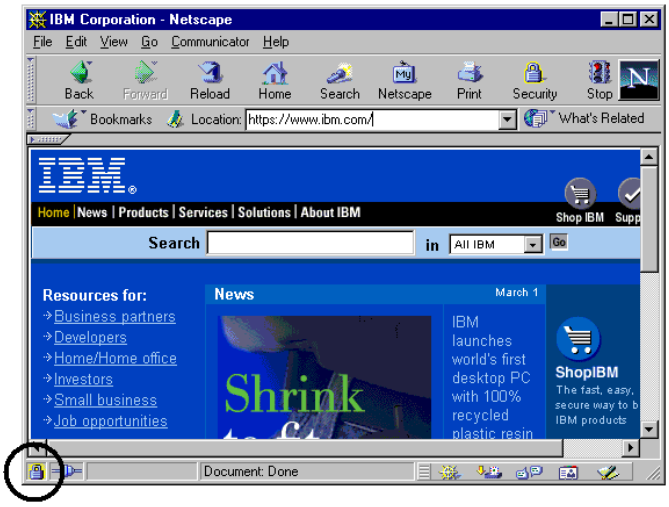
In practice, there are a few questions that you will need to have answers to before deploying security on your site, for example will you be using server authentication? Or client authentication? Who are your users and where are they connecting from? Are you in a private network/intranet, or will your server be on the Internet? Do your browser users trust you? And more importantly do you trust the people that are connecting to your server? These kinds of questions leads us into four broad areas of discussion.

- Server authentication
- Client authentication
- External Certificate Authorities
- Internal Certificate Authorities

Server Authentication

If the browser user that connects to your server has a trusted certificate or root certificate common with your server, then this provides the Web browser user with confidence that your server is who you say it is. If you want to control what the user can see on your secure site, then you will still need to manage user names and passwords for each user.

From the point of view of a browser user connecting to a Web site using SSL, the whole negotiation and authentication process can be quite transparent. To establish an SSL connection, the URL prefix must change from http:// to https://. Once the SSL connection has been established the browser gives the user a visual indication. In the case of Netscape Navigator this is a closed padlock symbol at the lower left of the screen (see figure below).



Client Authentication

With client authentication we take authentication one step further, and you as the server will want to know if you can trust the Web browser user. The Web user exchanges a client certificate which is signed by a Certificate Authority (CA) that is, a third party that you trust, or it may even be you (see "Internal Certificate Authority" later in this section). This provides you with confidence that the browser user represented by the certificate is the person you expect. You also have the added advantage of not having to manage passwords for these users, one less administrative burden, since their authenticity is now vouched for by the CA.

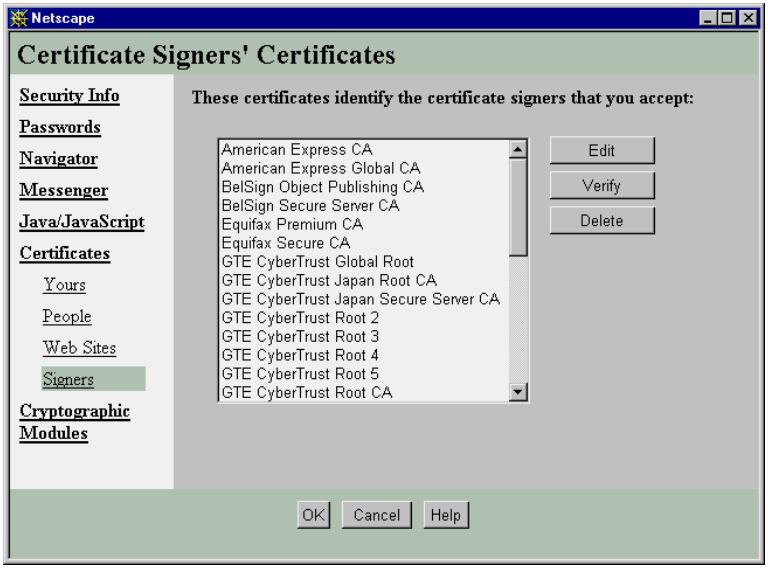
From the point of view of the webmaster, SSL is also quite simple. First the webmaster needs to generate a key pair for the server, and obtain a certificate for it. Normally this involves providing documentation to a certifying authority and paying an annual fee, although it is also possible to generate your own certificates for testing and intranet use. It is also possible to be your own Certificate Authority within your own organization. The difference between using an Internal CA or a third-party CA, is basically a matter of trust. Within an Intranet organization you may decide that it is reasonable for your employees to trust any server within the intranet, by the very nature that it is internal to your organization.

External Certificate Authority

If you are deploying your Web server onto the Internet, there is the issue of how and why a Web user on the Internet can believe you are who you say you are. This is particularly pertinent if they have to send credit details to you. Using an external CA (that is, a trusted third party) which your browser trusts, solves this problem.

You can see from the figure in the section "The SSL Handshake" that authentication in SSL depends on the client being able to trust the server's public key certificate. A certificate links the description of the owner of a key pair to the public part of the key. The validity of a certificate is guaranteed by the fact that it is signed by some trusted third party, the Certificate Authority (CA). But how does a certifying authority become trusted? In the case of an SSL-capable browser, the certificates of trusted authorities are kept in a key database, sometimes called a key ring file.

The list of top-level authorities, contains a set of certificates known as root certificates, that are pre-installed when you get the browser. The figure below shows part of the list of CA certificates provided by Netscape Navigator.



This approach has the benefit of being very simple to set up; a browser can authenticate any server that obtains a public key certificate from one of the CAs in the list, without any configuration or communication with the CA required. However, there are some problems arising from this method. The first is that a new CA will not automatically be recognized until the browser (wherever it may be in the world) has been updated. The second problem is that there is no way for certificate revocations to be processed. (For example, if a CA determines that a public key owner is fraudulent after a certificate is issued, the certificate will remain usable until it expires, without the end user being aware of any concern.)

The browser vendors have a two-part scheme to overcome the first problem (new CAs):

1. There is a special MIME format, application/x-x509-ca-cert, which allows a browser to receive a new CA certificate that has been signed by one of the known CAs. This format is specified in PKCS #7 (see http://www.rsa.com/rsalabs/pubs/PKCS/index.html). The browser will tell you that you are connecting to a secure server whose certificate is not from a known CA. You can then elect to trust just that server (that is, not the CA that signed the server's certificate).

For Internet applications you should purchase a certificate from one of the known CAs, such as VeriSign.

Internal Certificate Authority

There are instances when you may consider becoming an Internal Certificate Authority, this very much depends on who your browser users are. If they

are internal users, then you are dealing with a smaller risk and so it may be acceptable for you to be the CA, since your Web users will trust you. It will also means that you will not need to pay a yearly charge to the external CA.

As mentioned earlier, the browser comes pre-installed with a list of top-level authorities, this will leave you as the CA adminstrator with two options: either to install your new CA certificate in all your browsers, as described earlier, or prepare your users for the message that they will be connecting to a site that has been signed by a CA that the browser does not recognize.

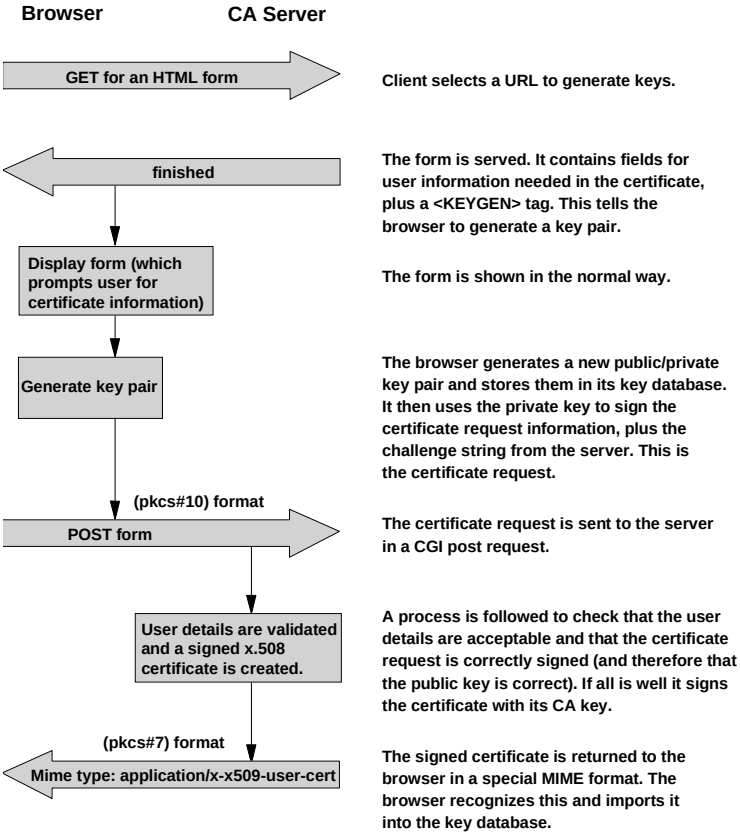
The following two figures show how Netscape Navigator warns you about a site that has a certificate from an unknown CA, and then allows you to trust the site anyway.

Note This will not make the CA trusted, so you will see the same warnings if you access a second server with a certificate signed by the same CA.

Serving Certificates to Browsers

Since a public key certificate provides proof of identity, it is reasonable to assume that the level of proof needed for a client is much lower than that needed for a server. Before providing a server certificate, the CA will require documentary proof of the legitimacy of the request. For a client, this proof can often be provided online, because a lower level of checking is needed. This is especially true of an intranet environment and certificate server products are initially intended for organizations that want to set up an internal authentication process.

Netscape and Microsoft use a different mechanism for initiating a client certificate request. The figure below shows how it works for Netscape. The mechanism uses the <KEYGEN> tag, an HTML extension that only applies within a form. When the browser sees this tag, it generates a key pair and returns a certificate request (in PKCS #10 format), for the key pair with the form to the CA. The CA processes the certificate request and sends it back as a signed X.509 v3 certificate in a special MIME format (known as PKCS #7 format), which the browser can accept.



With Microsoft Internet Explorer, the only change to the certificate request process is that Internet Explorer requires the Certificate Enrollment ActiveX™ control (*CERTENR3.DLL* for IE 3.0 and *XENROLL.DLL* for IE 4.0) to be installed. This ActiveX control generates the public/private key and encodes it in the PKCS #7 format for the CA, in exactly the same way that Netscape does.

Comparisons Between Notes Security and SSL

In the preceding section and the previous chapter, we have covered a lot of ground. The table below attempts to summarize some of this, comparing security functions at the application level in the Notes and Web environments.

Function	*Notes Implementation*	*Web+SSL Implementation*
Access Control	Multi-level/role-based access control, based on user authenticated Common Name.	Exclusive access to specific HTML files, directories, executables: that is, view or not view, execute or not execute.
	Full access control over what the client can see and do at the field, section, document, form, view, database, and server level.	Access control methods based on X.509 client authentication not yet developed.
User Management	Distributed directory of user information (Domino Directory) includes personal details and Notes and X.509 certificates.	Basic authentication: user ID and password file maintained separately on each server. Ability to create simple group of users.
	Ability to group users using distributed group profiles.	SSL Client authentication: directory of user information includes personal details and public-key certificate information. LDAP allows for cross directory access.
	LDAP allows for cross directory access.	
Certification scheme	Public-key Certificate Authority function provided by domain and subdomain servers. Integrated into the Domino directory.	Current X.509v3 certification: small number of well-known certification authorities allow server to be recognized.
	Cross-certification across multiple domains, in a controlled fashion. Manual registration by domain administrator.	Ability to become a Certificate Authority and distribute to browsers. LDAP will allow cross-directory access.

continued

Function	*Notes Implementation*	*Web+SSL Implementation*
Authentication	RSA public-key technology used to digitally sign a challenge string.	RSA public-key technology used to digitally sign a challenge string.
	Both Server/Client authentication.	Both Server/Client authentication by SSLv3.
Encryption methods	RSA technology (630/512-bit) used to encrypt key material which is subsequently used in a symmetric cipher for bulk data encryption.	SSL: RSA technology (key length varies) used to encrypt key material that is subsequently used for bulk data encryption.
	Encryption integrated with access controls.	RC2/4 or other technologies for bulk encryption. 128 bit key usually used for US and some categories of companies (such as banks) for non-US, and 40 bit for others.
	Bulk data encryption uses RC2/64 bit with 24 bits accessible by the US intelligence agencies in non-US version of Notes.	

The Domino Certificate Authority

So far we have described the role of SSL and certificates for secure Web communication. We have also mentioned the considerations for generating your server and client certificates, and whether you want to issue your certificates from a third-party provider or be your own CA. Below we describe the X.509 certificate services offered by Domino.

Setting Up the Domino Certificate Authority

A Certificate Authority (CA) is the common trusted third party that allows a server and a client to use SSL, as we described above. A CA is also the common trusted third party for exchanging secure e-mail messages using S/MIME, as we will describe later in this chapter. You can use a third-party, commercial company such as VeriSign, as the external CA. Or you can use a *Domino Certificate Authority* as an internal CA in your organization.

The Domino Certificate Authority application template is shipped with Lotus Domino R5.0. You can perform CA management tasks through the Domino Certificate Authority database, such as accept certificate requests, issue certificates, and manage issued certificates.

Key Ring File

Before you attempt to become a CA or request a certificate from a CA as a webmaster, you must create a *key ring*, which is the storage for X.509 SSL certificates for a Domino server. A key ring file is a binary file that is password-protected, it is very similar to the password protection for Notes ID files.

CA Key Ring

If you want to establish a Domino CA, you first have to create the CA key ring file. The CA key ring contains the CA's public/private key pair and the CA's certificate. You will use the CA key ring to accept and digitally sign server and client certificate requests.

You can use a Domino Certificate Authority application (CERTCA.NSF) to manage a CA key ring. Initially, you have to create the database from the Domino Certificate Authority template (CERTCA.NTF). Usually, the key ring file is named with the extention "kyr." The default filename for a CA key ring is "CAKey.kyr." You should keep the CA key ring file and the Certifier's Notes ID in a secure location. You should not store them on the server machine's hard drive.

Server Key Ring

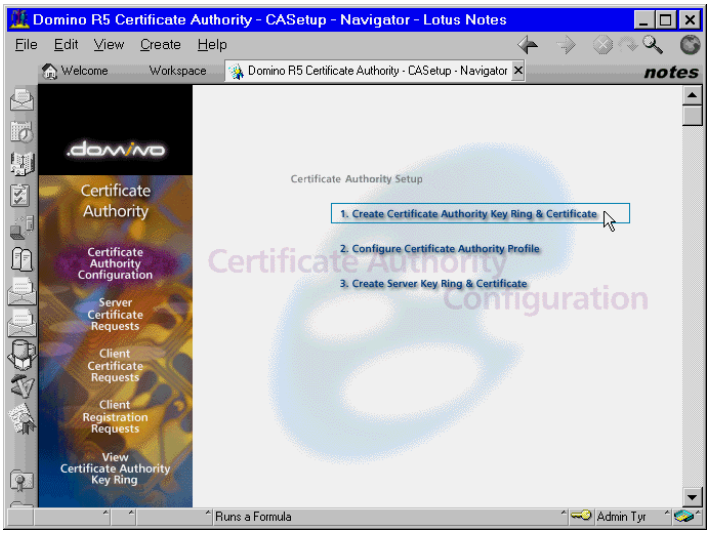
A server key ring file is stored on the server's hard drive. When you create a server key ring file, Domino generates an unsigned server certificate and automatically includes several trusted root certificates (issued by VeriSign, RSA, Netscape, and so on). The unsigned server certificate is not valid until the CA signs it.

You can use the Server Certificate Admin application (CERTSRV.NSF) to manage a server key ring. The default file name for a server key ring is "KEYFILE.KYR."

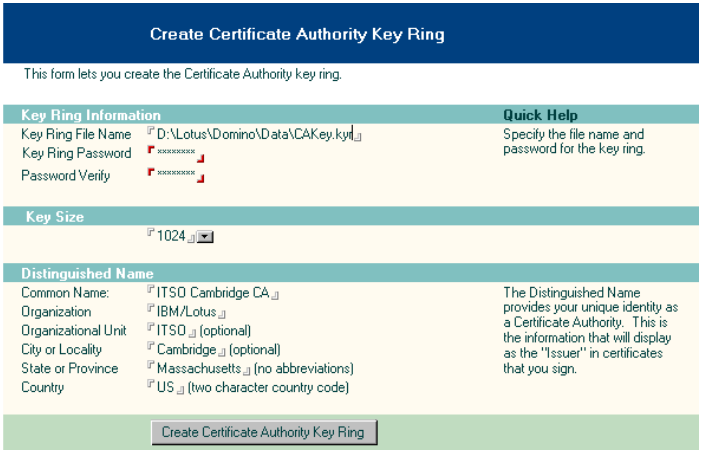
Setting Up Your Domino Server as an Internal Root CA

This section describes the procedure for setting up your Domino server as the root CA in your organization. The result of this operation is that a special password protected key ring file is created for the CA.

1. Using a Notes client, create a Domino Certificate Authority database from the Domino Certificate Authority template (CCA50.NTF). Name the database "CERTCA.NSF." The figure below is the initial screen for the Certificate Authority application.



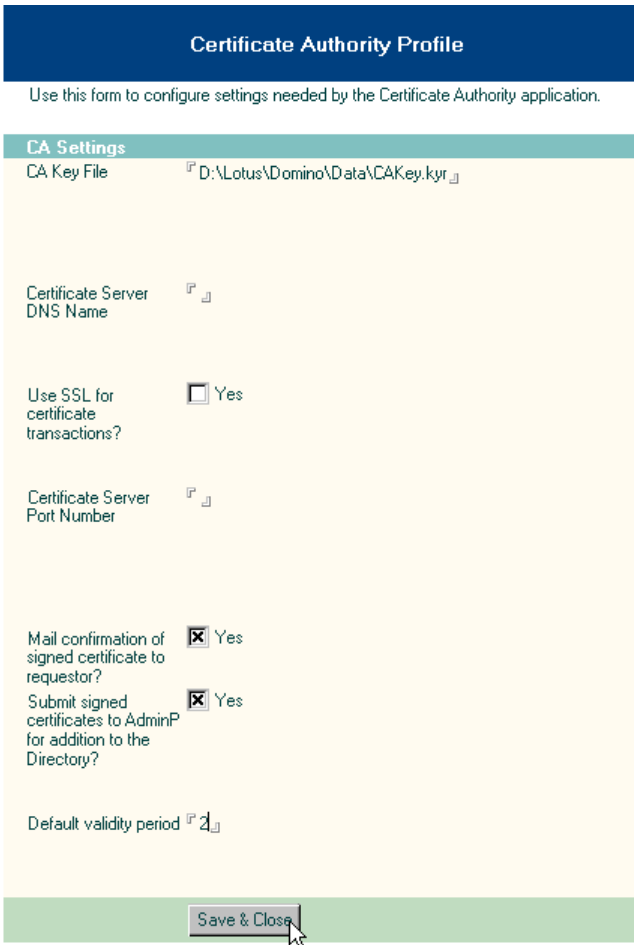
2. Choose Create Certificate Authority Key Ring & Certificate. Specify the required information as shown in the following figure. Key ring file name is the file and pathname relative to your Notes client's data directory. You may change this to any appropriate file and pathname.



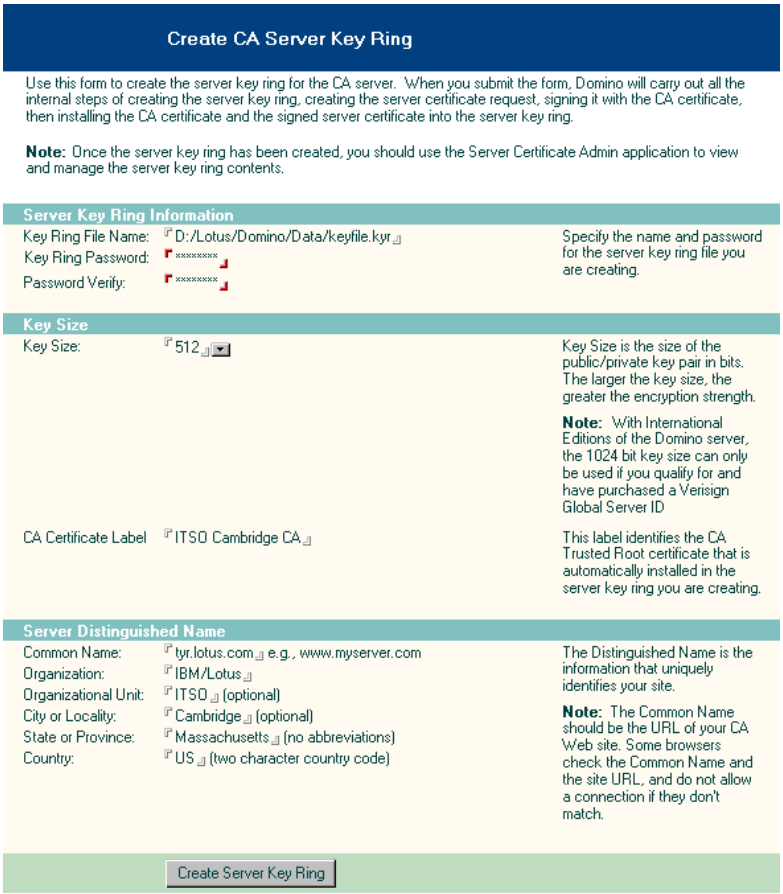
3. Press Create Certificate Authority Key Ring to generate the key ring file. You will see the confirmation screen. Press OK to proceed.

4. Check whether the CA key ring is generated properly. Select View Certificate Authority Key Ring from the navigator, on the left side of the screen, and press Display CA Key Ring. You will be asked to provide the CA key ring password, and the database will load the contents of the key ring. As the following figure shows, you can browse the certificate information from this view.

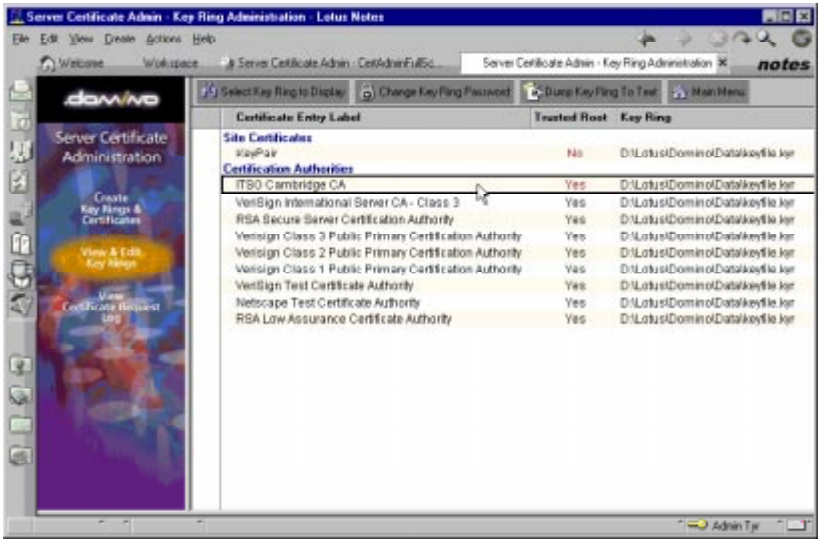
5. Choose Configure Certificate Authority Profile from the initial menu. At this time we do not want to use the Certificate Server; therefore, we will leave several entries blank.



6. Choose Create Server Key Ring & Certificate from the initial menu to create the server key ring. Since this key ring is used by the Domino HTTP service, the common name should be the URL of the Web server. Press Create Server Key Ring to proceed. You will be asked to enter the password for the CA key ring.



7. Let's take a look inside the server key ring. Open the Server Certificate Administration database (CERTSRV.NSF) and select View & Edit Key Rings from the navigator. You will see that the key pair is installed. In addition, you will find that several CA certificates, including your Domino CA certificate, are already installed and marked "Trusted Root."



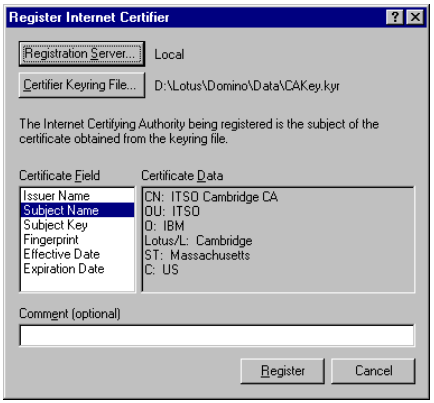
8. Enable the SSL port on your Web server, and restart the HTTP server task. Please see the following section on how to set up SSL with your Domino Web server.
9. Now your CA Web site can be reached via a Domino URL like this:

```
http://your.web.server/certca.nsf?OpenDatabase
```

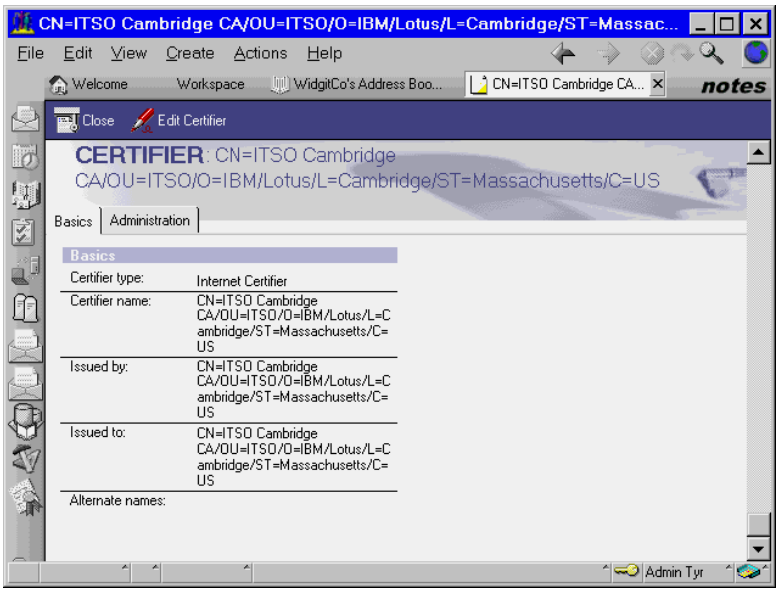
Registering Your CA to Domino Directory

If you plan to issue X.509 certificates to Notes users, your CA must be registered to the Domino Directory first. A Certifier entry in the Domino Directory contains the CA certificate, and those Notes users who belong to the Domain will fetch the CA certificate from the directory.

1. On the Configuration tab in Domino Administrator, choose Tools - Registration - Internet Certifier. Select the CA key ring file and enter the password.
2. A dialog box appears as shown below. Choose Registration Server for the new Certifier entry, and click Register.



3. You can use the Certificate Field options to select which information from the certificate is to be displayed in the Certificate Data field. You can also enter text in the Comment field, which is stored in the Certifier entry.
4. Click the Registration Server button and make sure that the server holding Domino Directory is specified. Click OK to save the registration server information.
5. When you are satisfied with the information in the key ring file and with the choice of Registration Server, click Register to create the Certifier entry in the Address Book. An Internet Certifier entry is immediately created in the Domino Directory containing the CA's name and self-signed certificate.



Invoking SSL on Your Domino Server

To invoke SSL on your Domino server with Internet protocol services such as HTTP, you must take two steps:

1. Prepare a key ring which contains a public/private key pair for the server and several X.509 certificates.

   There are several ways to obtain a key ring file:

   - Create a self-certified certificate and key ring
   - Create a key ring and request a certificate issued by an external CA

2. Configure the Server document in the Domino Directory to enable the SSL communication.

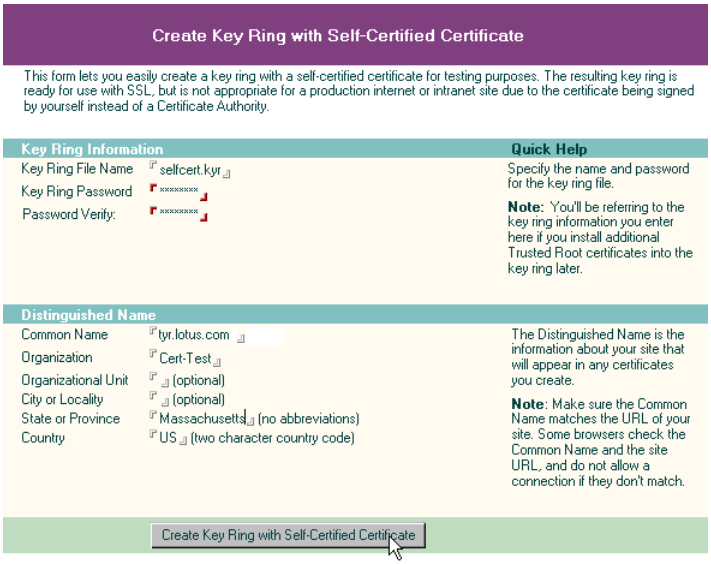
Using a Self-Certified Certificate

You can create a self-certified (or self-signed) certificate to test the certificate procedure at your organization. Because this certificate is not certified by a CA, you should use it only for testing purposes. You can create a self-certified key ring using the Server Certification Administration database (CERTSRV.NSF).

This is an adequate option if you don't need to work with any external or internal CAs. With the self-certified certificate on server-side, clients cannot check the validity of the server certificate, because clients and servers can never share any signer in common. When you connect to a server that has a self-certified certificate, your Web browser will tell you that the sign cannot be recognized automatically, and will ask you to permit the connection at your own risk.

To create a self-certified key certificate, do the following:

1. Open the Server Certification Administration database (CERTSRV.NSF) and choose Create Key Ring with Self-Certified Certificate from the initial menu.
2. Fill in the required information and click Create Key Ring with Self-Certified Certificate.



The self-certified key ring is created immediately. To use this key ring file on your server, you have to specify the name of key ring in the Server document. Please see "Configuring the Server Document" later in this chapter.

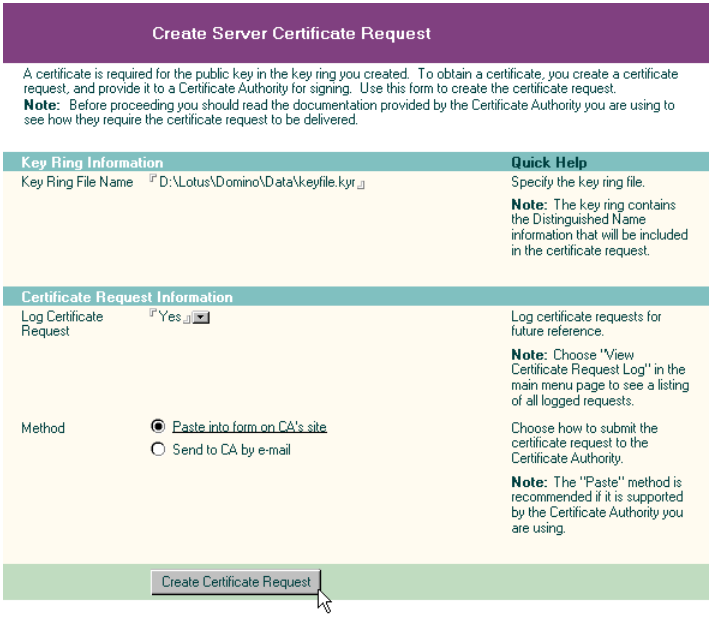
Applying for a Server Certificate to an Internal/External CA

You can apply for a server certificate to an internal or external CA. To illustrate this, we show you the actual step-by-step procedure below. We tested the Domino CA as an example of an internal CA, and VeriSign as an example of using an external commercial CA.

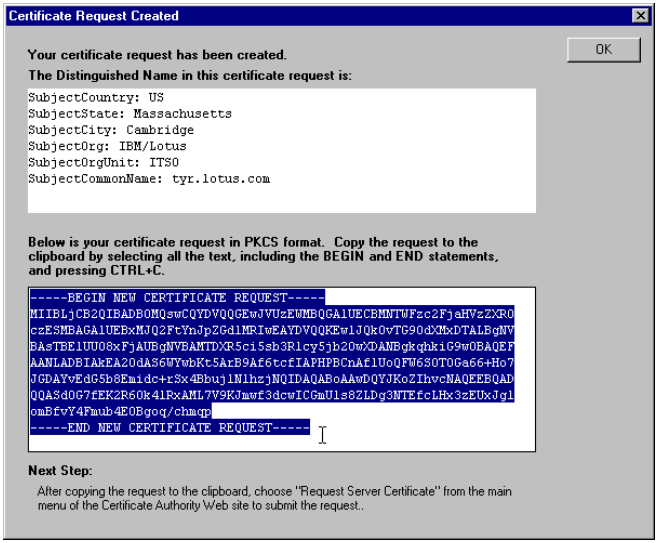
Please note that the actual procedure for obtaining a server certificate may vary depending on the CA. When you request an SSL server certificate, you use the standard Public-Key Cryptography Standards (PKCS) format, an industry-standard format that many CAs, including Domino, understand. Before you request a certificate from an external CA, make sure the CA uses the PKCS format, not some other format, such as Privacy-Enhanced Mail (PEM). If you are unsure of the format required by an external CA, check with the CA.

1. Make sure your server has a server key ring file. To create the key ring using the Notes client, open the Server Certificate Administration database (CERTSRV.NSF), and choose Create Key Ring.
2. Create certificate request. In the Server Certificate Administration database, choose Create Certificate Request.
3. Specify the pathname of the key ring file, and the method for passing the certificate request to the CA. Domino CA and some commercial CAs,

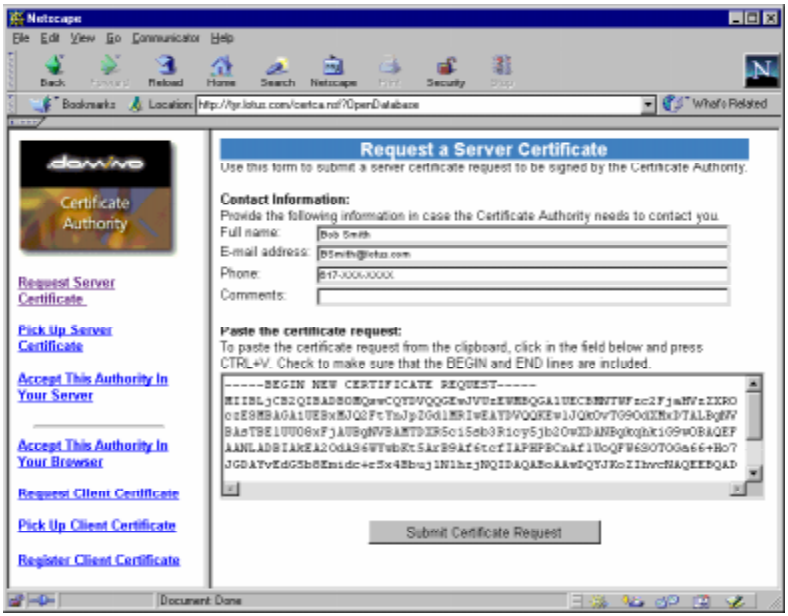
such as VeriSign, allow us to “paste” the request on the Web page. We selected Paste into form on CA’s site for this example as shown below. After entering the required information, click Create Certificate Request.



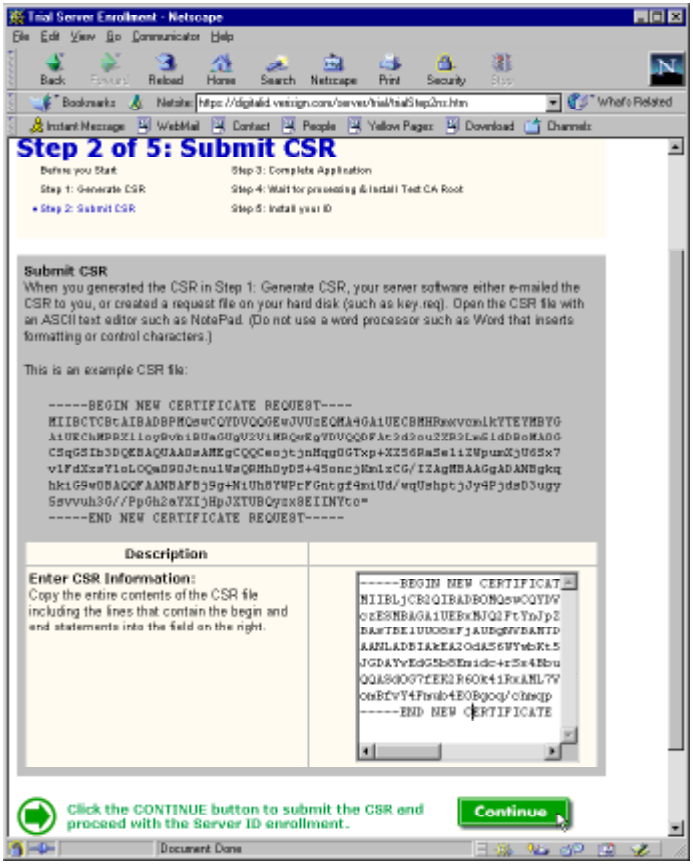
4. Your certificate request is immediately created in PKCS format, and it is displayed in a dialog box as shown below. Select the certificate, including the BEGIN line and the END line, and copy it onto the clipboard. Or, you may want to save it as a text file.



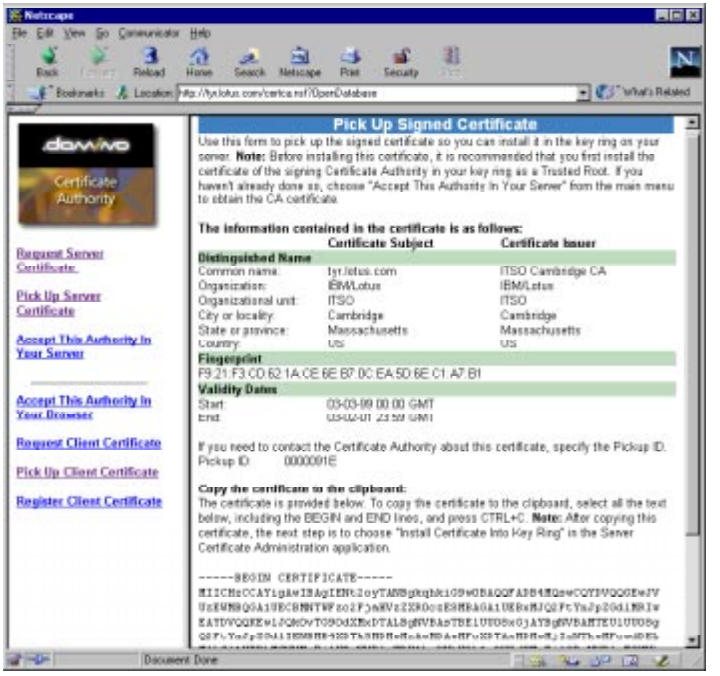
5. Go to the CA's Web site. If you attempt to use Domino CA, open the CA database and choose Request Server Certificate. Paste the certificate request in the appropiate field, as shown below.



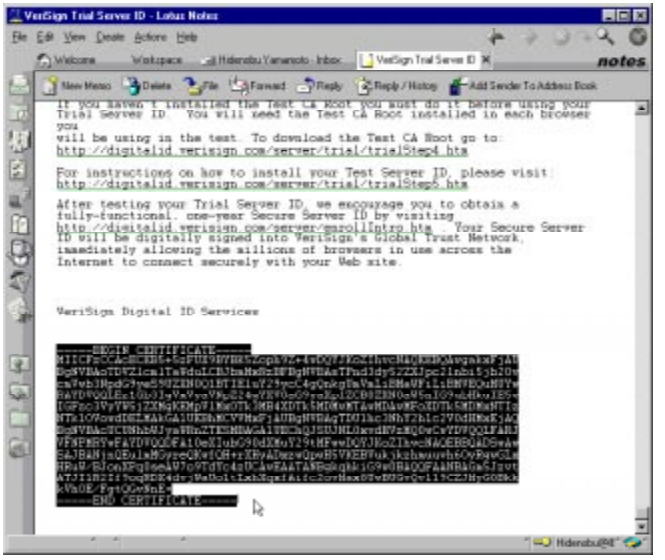
6. If you want to use a commercial CA, locate and open the appropriate page for applying for a server certificate. As an example of using commercial CA, we tried to obtain VeriSign's Free Trial Secure Server ID, as shown in the figure below (VeriSign calls a certificate request, "CSR"). Paste the certificate request on the Web page.



7. When you use Domino CA, the Domino CA administrator will advise you of a "Pickup ID" via e-mail. After receiving it, go to the Domino CA Web page, select Pick Up Server Certificate, and enter the Pickup ID. When the certificate is displayed on the Web page, copy it onto clipboard.

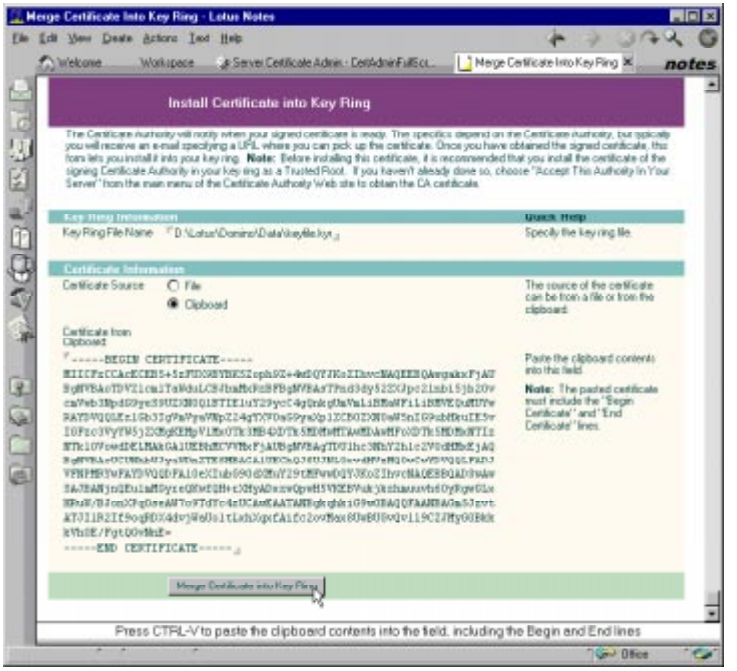


8. When you use VeriSign, you will receive an e-mail message containing the certified certificate. Copy it onto the clipboard.



9. Install the certificate into your server's key ring file. Open the Server Certificate Administration database from the Notes client, and select

Install Certificate into Key Ring. Paste the certificate into the document, and click Merge Certificate into Key Ring.



10. A dialog box to confirm the procedure will be displayed. Press OK.
11. Make sure all clients/servers have a root CA certificate. Without a shared root CA certificate among users and servers, your client cannot verify the validity of the server certificate, that is, your client cannot "trust" the server. To pick up the CA's certificate and install it to your server's key ring, click "Accept this Authority In Your Browser" in the initial Domino CA screen.

Applying for VeriSign Global Secure Site ID

Domino R5.0 supports VeriSign Global Secure Site IDs that negotiate a 128-bit encryption when communicating with international browsers and servers using SSL over HTTP, NNTP, LDAP, IMAP, and POP3. In previous releases, Domino supported the Global Server ID only on SSL over HTTP.

What Is VeriSign Global Secure Site ID?

VeriSign Global Secure Site IDs, formerly called VeriSign Global Server IDs, are a form of server certificate. Global Secure Site IDs enable the negotiation of SSL (or TLS) sessions using 128-bit RC2 or RC4 encryption. At the time of writing this redbook, VeriSign, Inc. is the only company that is approved by the US Government to sell certificates with this very secure encryption.

Until recently, due to US export regulations, it was virtually impossible for a browser and a server to communicate using key lengths longer than 40 bits unless both the server and browser were located within the US. Using ordinary server certificates, such as VeriSign Secure Site IDs, US-based companies with servers located in the US can communicate at 128-bit encryption within the US, and at 40-bit encryption outside the US. Non-US-based servers can communicate at 40-bit encryption with their clients.

The US Government determines the categories of companies that can use Global Secure Site IDs outside the US and across US borders. At present, according to VeriSign's Web site, the categories are defined as the following:

- Banks and Financial Institutions
- Insurance Companies
- Health and Medical Organizations
- Online Merchants
- US Subsidiaries

Applying for Global Secure Site ID

If you want to obtain VeriSign Global Secure Site ID, you will be requested to supply some information to VeriSign.

As part of the enrollment process of the ID, your organization will be asked to provide information that establishes its corporate identity and ensures that the institution meets the US Commerce Department definitions for those business categories. You can provide a Dun & Bradstreet D-U-N-S number for this purpose (please see **`http://www.dnb.com/`** for information about D&B D-U-N-S numbers). In addition, your organization will be requested to agree to the "Global Secure Site ID Subscriber Agreement."

If you want to read a "Global Secure Site ID Subscriber Agreement," or if you want to learn more about Global Secure IDs, please go to VeriSign's Web site at **`http://www.verisign.com/.`**

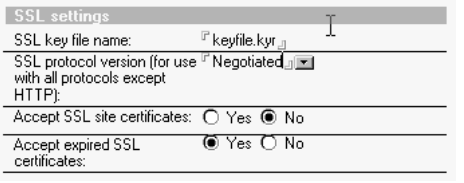
Installing and Using Global Secure Site IDs

No special configuration is needed to install Global Secure Site IDs onto Domino servers. That is, you just have to install the Global Server ID as well as an ordinary server certificate. And, now, the latest export and US domestic versions of Netscape Navigator and Microsoft Internet Explorer browsers can encrypt transactions with your site using strong encryption in 128-bit sessions. If you have set up SSL for the Internet protocol, Domino automatically initiates communication using a 128-bit encryption.

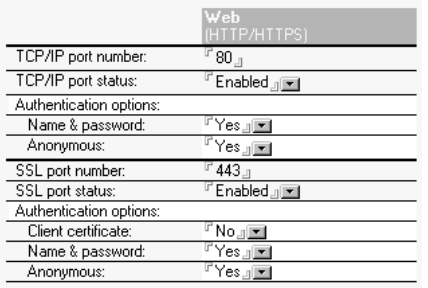
Configuring the Server Document to Enable SSL

To turn on the SSL port on your Domino server, you have to modify the Server entry of the Domino Directory. We configured SSL over HTTP. You can use a similar procedure for other protocols (NNTP, LDAP, IMAP, POP3, SMTP).

1. Open your Server's entry on the Domino Directory. Select Ports - Internet Ports - Web.
2. Modify the SSL settings. Specify your key ring file name (or path name relative to the Domino data directory).



3. Change SSL port status to Enabled.



4. You must re-start the HTTP server task so that HTTP over SSL (HTTPS) is activated on your server.

Issuing X.509 Certificates by Domino CA

Typically, a three-step process is needed for a user to acquire an Internet X.509 certificate:

1. The user creates a public/private key pair, the keys are stored locally, a copy of the public key is sent to a Certificate Authority (CA) in the form of a certificate request.
2. The CA processes the request, takes some action to verify that the request is valid, for example, that the public key does indeed belong to the user making the request, issues the certificate, and either posts it in a public place for the user to retrieve, or mails it to the user.

3. The user retrieves the certificate, either from a public place or from a mail message, and stores the certificate with the previously stored private key.

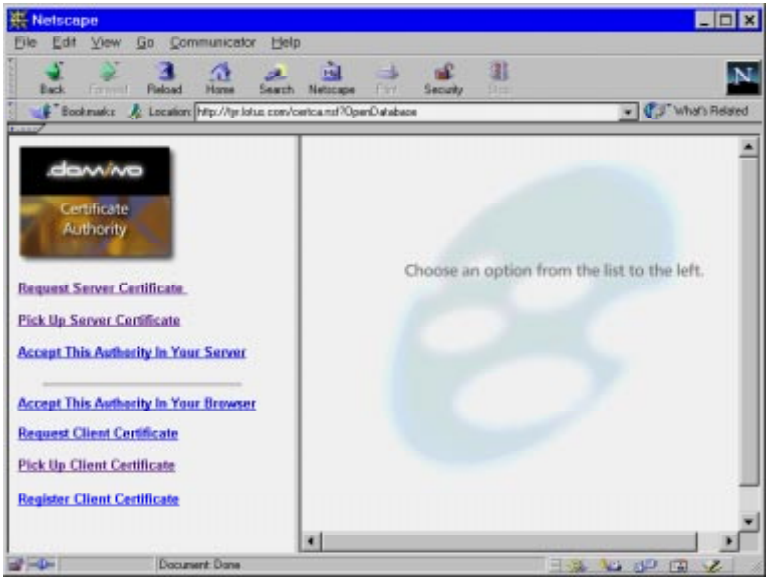
Issuing X.509 Certificates to a Web Browser

This section describes the steps taken by an administrator of the Domino CA. We show this example using Netscape Communicator 4.5. If you want to install a certificate issued by the Web CA into your Notes ID, you must use the Notes Browser (Web Navigator) — see "Obtaining X.509 Client Certificates using Notes Browser" later in this chapter.

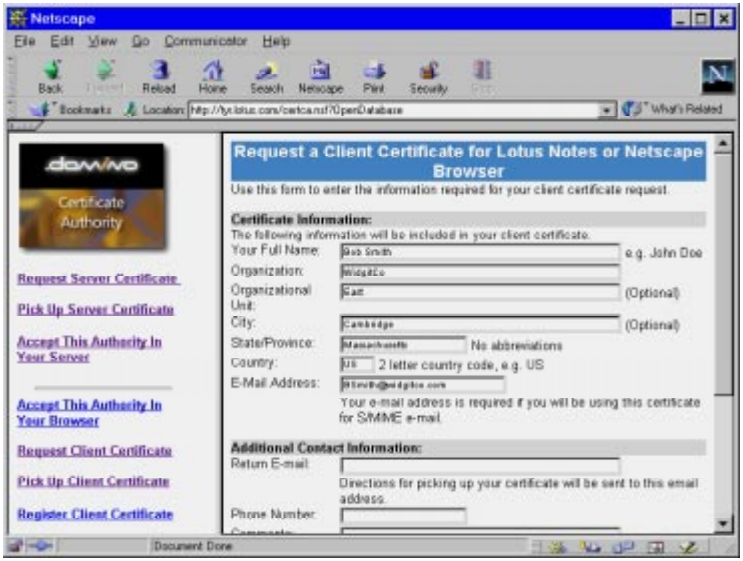
Note The AdminP process for the Domino CA server must be running before you are able to issue certificates.

Step 1: User Request Client Certificate

1. Go to the Domino CA Web page. The Domino CA database is named CERTCA.NSF; therefore, the URL would look something like this: http://yourserver/certca.nsf?OpenDatabase. Select Request Client Certificate.



2. Fill in the form, and click Submit Certifier Request.



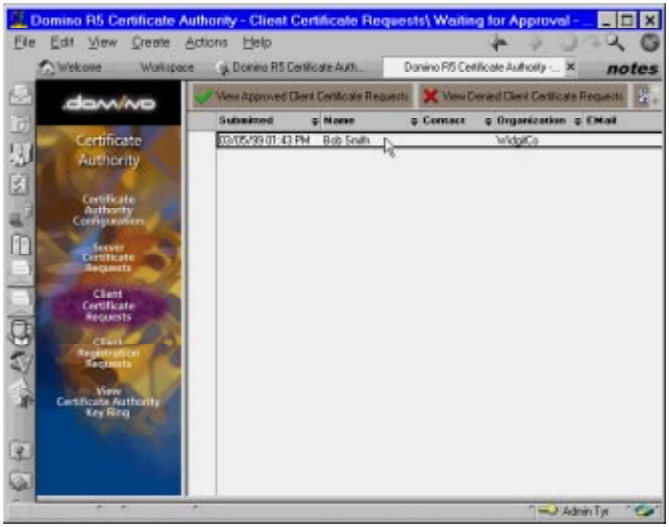
3. The users's browser will generate a private/public key pair.

4. The key pair is stored in your local key ring file. In Netscape terminology, a key ring is called "Communicator Certificate DB."

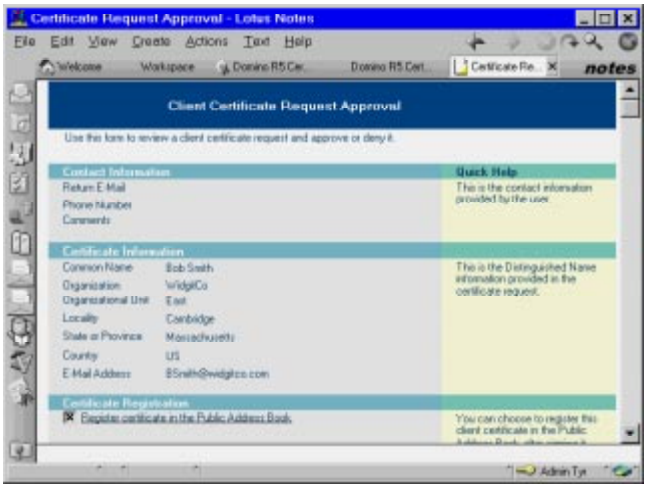
5. The Certificate request is sent via the HTTP session.

Step 2: CA Accepts Certificate Request and Issues Certificate

1. The Domino CA administrator will see the certificate request received in the Certificate Authority database.



2. Open the request document, confirm the contents, and click Accept or Deny.



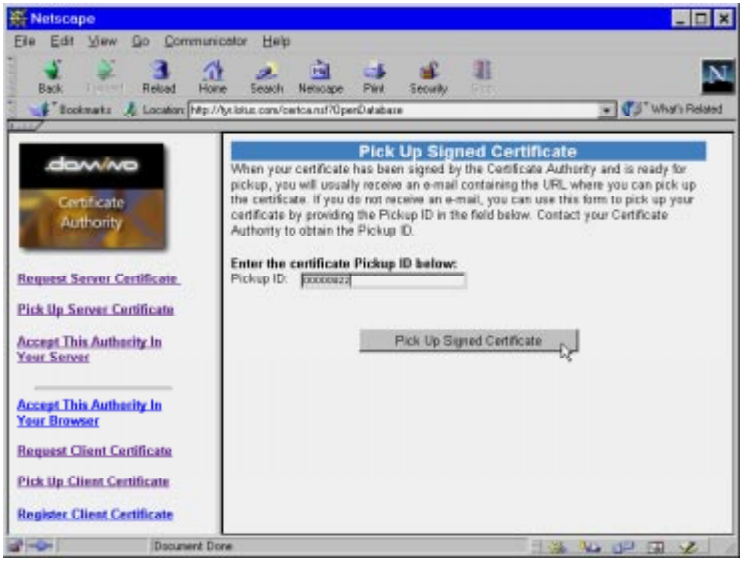
In this registration process, you can register the client certificate in a Domino Directory. The client certificate will be stored in the specified person's entry by the AdminP process if you check the box in the figure below.

Note This operation will submit an administration request to the Administration Requests Database on your administration server. The AdminP server task will then do the actual work of registering the certificate in the person document in the Domino Directory.

3. Send an e-mail to the user and include the Pickup ID as well as information on where the user can pick up the certificate.

Step 3: User Obtains and Installs Client Certificate

1. After receiving the Pickup ID (or PIN), the user visits the CA Web page again to pick up the signed certificate. Choose Pick Up Client Certificate on the navigator, enter the Pickup ID, and click Pick Up Signed Certificate.



2. The user's browser will ask you to name the certificate you are about to receive. Specify the name and click OK. Your client certificate will be installed in your key ring file automatically.

3. To confirm if your certificate is installed properly, choose Security - Certificates - Yours.

4. If the CA is an internal CA, or you have not obtained the CA's root certificate, you should request the root certificate as well. A user can choose Accept This Authority in Your Browser to install the root CA certificate.

Registering the Client Certificate to Domino Directory

You can use Domino Directory as a directory for X.509 client certificates. There are a couple of ways to register certificates to Domino Directory, such as:

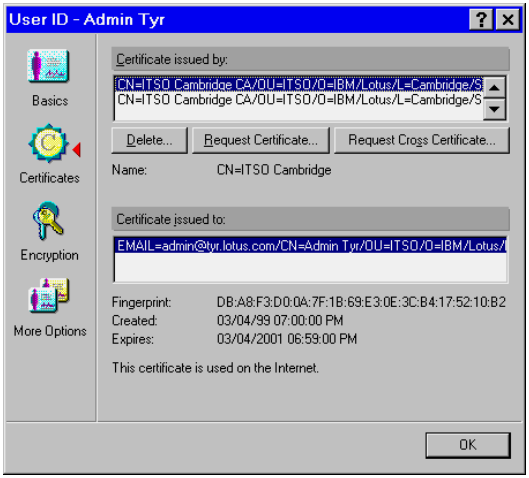
- At the certificate request approval step, you can specify the option of registering the certificate to the Domino Directory.
- A user can request certificate registration through "Client Registration Requests" menu of Domino CA.

Issuing X.509 Certificates for Notes

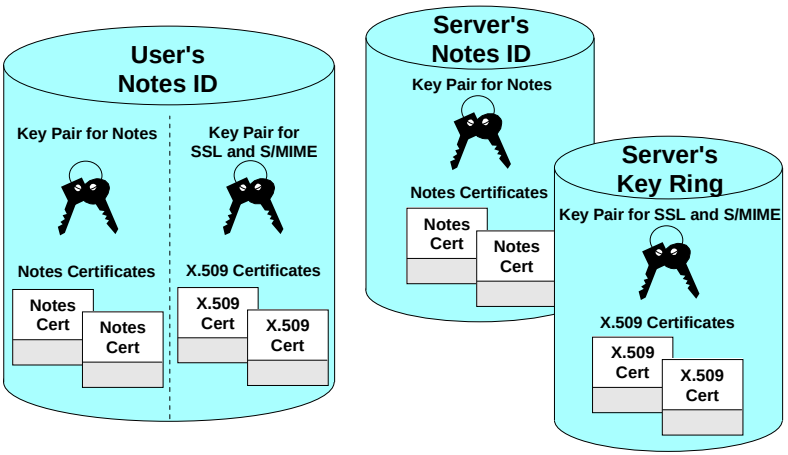
Notes R5.0 clients are able to use X.509 certificates on Internet communications such as HTTP over SSL, S/MIME, and so on. This section describes how and a Notes ID handles X.509 certificates, and how you can obtain X.509 certificates for use by the Notes client.

Notes/Domino R5 and X.509 Certificates

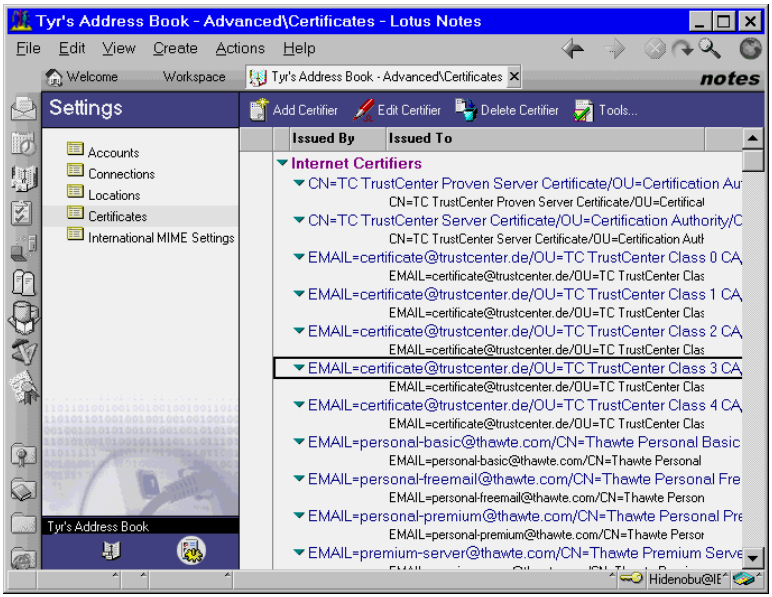
With Domino R5.0, the Notes ID file for client authentication is now able to hold X.509 certificates as well as Notes certificates.



Note You still need key ring files to maintain X.509 certificates on the Domino Server side.



Your client's Notes ID only holds X.509 certificates required to prove your identity, that is, your own certificate and the CA certificates (or chain of certificates) that you trust. Other X.509 certificates are stored in your Personal Address Book or Domino Directory. The figure below is an example of a Personal Address Book. You will see the X.509 certificates under the Certificates view.

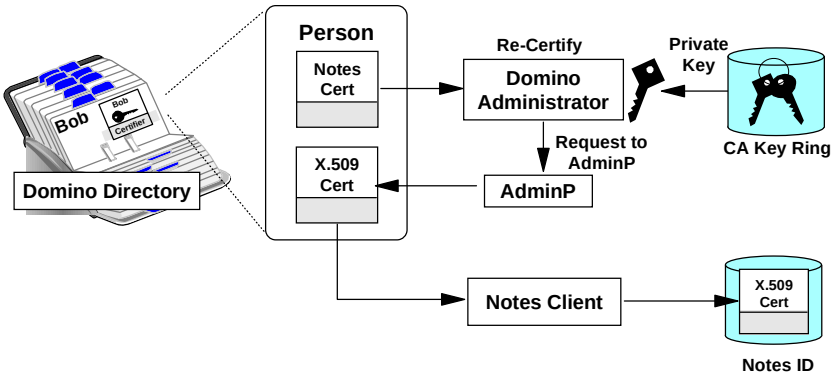


Note As you see in the figure above, several major CA certificates are "pre-installed" in your Domino Directory, because the Domino Directory template database contains such certificates in Domino R5.0. This means that if you upgraded from a previous version of Notes, it is probable that the certificates have not been copied to your Domino Directory. In that case, you have to copy them into your Domino Directory manually.

Generating and Distributing X.509 Client Certificates for Notes

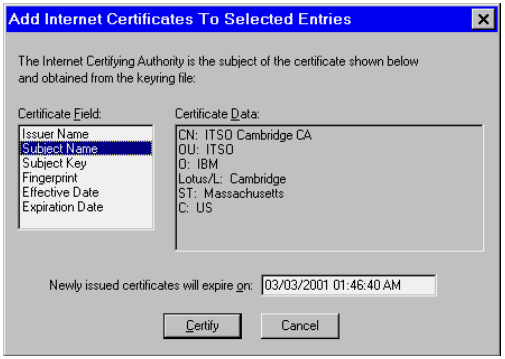
With R5.0, you are now able to generate X.509 certificates for Notes and distribute them through the Domino Directory.

Notes clients are able to use X.509 certificates on several Internet communications such as HTTP over SSL, S/MIME, and so on. The Domino Certificate Authority application and Domino Administrator client enable administrators to issue X.509 certificates to Lotus Notes R5.0 clients. In this process, Notes certificates are encapsulated and re-certified, while X.509 certificates are generated. Administration Process (AdminP) will distribute the certified X.509 certificates using Domino Directory. This process never removes the existing Notes certificate, that is, as the result of this operation each Notes certificate and X.509 certificate will "share" the same public/private key pair.

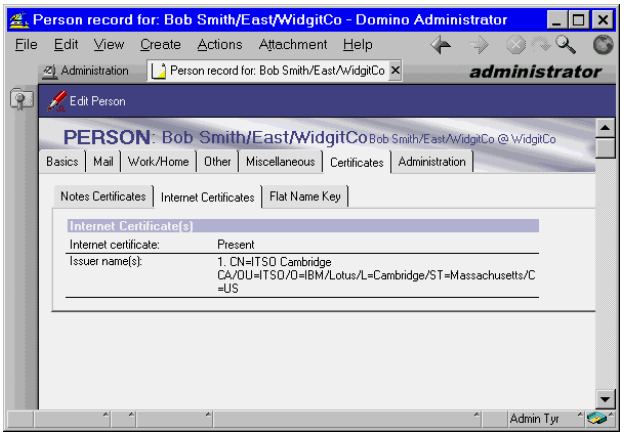


The actual procedure follows. This procedure is not only shorter, it is automatic. The certificate is automatically entered into the user's ID file in the normal course of using Notes.

1. First, make sure your Domino CA is registered to Domino Directory (see "Registering Your CA to Domino Directory" for details).
2. Click the People & Groups tab in the Domino Administrator on any of your servers that contain a corresponding Administration Requests Database (ADMIN4.NSF). Open the People view and select the users to receive the certificates.
3. Choose Actions - Add Internet Cert to Selected People. You will be asked to specify the CA key ring file name and the password, and the desired certificate expiration date. In the dialog box shown, click Certify to start the actual certification operation.



4. For each selected user, Domino Administrator will pick up the Notes certificate from the Domino Directory, create X.509 certificates by re-certifying them with CA's private key, and submit an Administration Request for the AdminP task to add the X.509 certificate to the UserCertificate field of the corresponding user's Person document (see the figure below).



5. The next time the user accesses his or her home/mail server, Notes automatically inserts the Internet certificate into the ID file.

Obtaining X.509 Client Certificates Using the Notes Browser

You can obtain X.509 certificates from a Web-based CA site using a Notes Web browser (or Web Navigator), as well as Netscape or Microsoft Web browsers.

To obtain Certificates from either a Domino CA or a third-party Web CA using the Notes Web browser, it must emulate the Netscape browser. To emulate the Netscape browser, set the NOTES.INI variable:

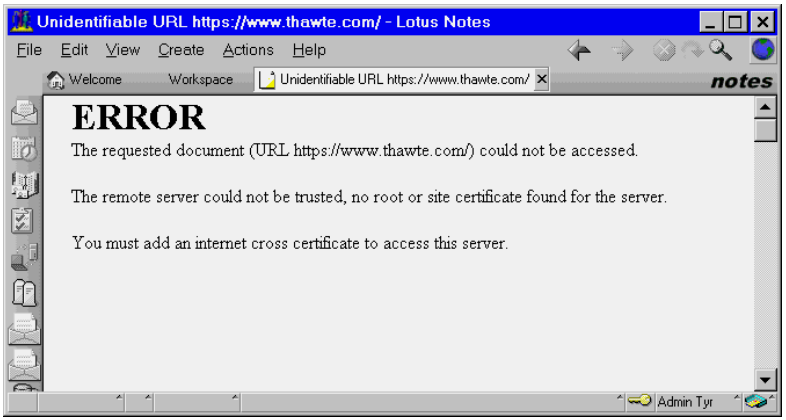
```
WebUserAgent=Mozilla/4.0
```

When obtaining a certificate using Web retriever, it is advisable to configure your location setting so that the Web retriever process updates the cache every time. Select Every time in the Update cache field in Web Retriever Configuration. You will find the Web Retriever Configuration section under the tab Advanced - Web Retriever in your location document.

When you attempt to retrieve an X.509 certficate using the Notes browser, you will be asked to click Yes to install the X.509 certificate into your Notes ID file.

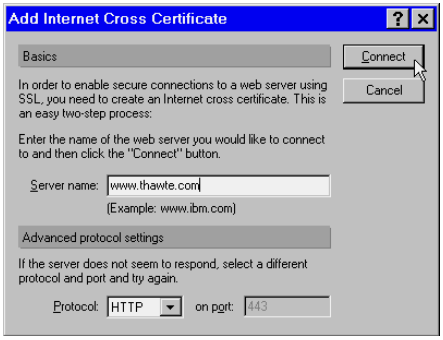
Internet Cross-Certificate

When you want to connect to a Web site using SSL, your Notes browser will try to check the validity of the certificate of the Web server. To verify a digital signature appended to the certificate, you need a root CA certificate that the Web server belongs to. Without this, the Notes browser will fail to open the Web page using SSL, and an error message will be displayed.

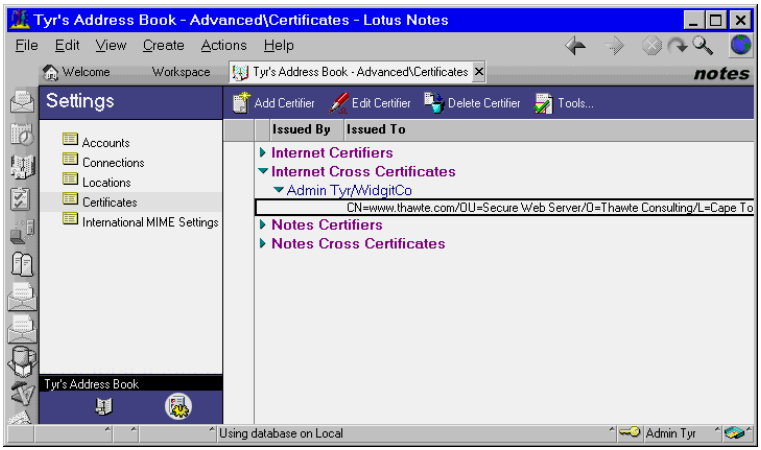


In order to "trust" the other party, you can install the CA's certificate as a trusted root, or, you can issue an *Internet Cross-Certificate* to the Web server or to the signer to the server. An Internet Cross-Certificate is the certificate of another party that is also signed by your private key. In other words, this means that you "trust" the other party at your own risk.

To create an Internet Cross-Certificate for a server, choose File - Tools - Add Internet Cross Certificate on your Notes client, and specify the target server's address.



Notes will generate the Internet cross-certificate, and store it in your Personal Address Book.



Cross certificates can be created directly from the Domino Directory. Open a certificate entry from the Internet Certifiers category in the Certificates view; then choose Actions - Create Cross Certificate.

Secure E-mail Messaging

Currently the X.509-based security infrastructure is used for the Web and other Internet applications. So far our discussion of X.509 Certificate-based security has been primarily focused on session encryption of the HTTP protocol for Web browsers. Using the same techniques, we can provide secure e-mail communications.

This section describes the facilities to provide secure messaging between e-mail clients over the Internet, and specifically how this security is implemented when Lotus Notes is used as the Internet client and Domino as the Internet mail server.

Let us begin with the basics of Internet protocols commonly used for e-mail messaging.

Commonly Used Mail Protocols

Much of the growth of e-mail on the Internet is due to the simplistic nature of the protocols used; this has proven to be a double-edged sword. Simplicity in operation scales well for millions and millions of users, but simplicity in design has left many open security loopholes.

Following is a look at the various Internet protocols that are typically used for sending and receiving Internet mail.

SMTP

SMTP (Simple Mail Transport Protocol) specifies a protocol for sending e-mail messages between hosts, although with the use of Domain Names Service (DNS) and Mail eXchange (MX) records, it can be thought of as sending e-mail messages to users between domains. Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another. In addition, SMTP is generally used to send messages from a mail client to a mail server. Any host that supports SMTP can also act as a SMTP relay, which can forward messages to another SMTP host.

MIME

Short for Multipurpose Internet Mail Extensions, a specification for formatting non-ASCII messages so that they can be sent over the Internet. One of the problems with the original SMTP specification was that it assumed e-mail messages consisted primarily of text; the format specifies the use of plain ASCII text. MIME extends the specification by allowing binary data to be repackaged in text form and transmitted over the Internet in mail messages that are compliant with the original specification. Typically, an e-mail message which supports MIME would have extra header messages after the Subject field.

```
From: skumar@cyberude.com
To: poojanayar@hotmail.com
Subject: Donuts... What will they think of next...
MIME-Version: 1.0
Content-Type: image/gif
Content-Transfer-Encoding: base64
Content-ID:
Content-Description:
[...GIF data...]
```

Most e-mail clients now support MIME, which enables them to send and receive graphics, audio, and video files via the Internet mail system. In addition, MIME supports messages in character sets other than ASCII.

POP and IMAP

Both POP and IMAP are protocols that specify protocols for accessing mail from an Internet mail box or post office.

The Post Office Protocol, Version 3 (POP3) is used to pick up e-mail across a network. Not all computer systems that use e-mail are connected to the Internet 24 hours a day, 7 days a week. Some users dial into a service provider on an as-needed basis, and others may be connected to a LAN with a permanent connection but may not always be powered on. In cases such as these, the e-mail addressed to the users on these systems is sent to a central e-mail post office system where it is held for the user until pickup. POP3

allows a user to log onto an e-mail post office system across the network, validates the user by ID and password, allows mail to be down-oaded, and optionally allows the user to delete the mail from the server.

IMAP (Internet Message Access Protocol) a newer protocol that allows for e-mail clients to retrieve e-mail messages from, and work with, the mailboxes on a mail server. The latest version, IMAP, is similar to POP3 but offers additional and more complex features. With IMAP, for example, you can work with your e-mail on the server, and sort and manage your e-mail on server-side folders. For more information about IMAP, see Stanford University's Web page at:

`http://www-camis.stanford.edu/projects/imap/ml/imap.html`

Problems with These Protocols

As mentioned earlier, the simplicity of these protocols has meant that they pose security issues for anyone sending and receiving mail across the Internet.

SMTP

The SMTP protocol does not use any authentication process when establishing communications with another SMTP host for relaying and delivering mail. The sending host basically sends a command to the receiving SMTP host saying who it is, and that it wants to communicate. The receiving host believes who it says it is, and readily awaits further commands. The sending host then sends another command saying who the mail is from, which the receiving SMTP host then accepts; the sending host then sends another command saying who the intended recipient of this mail is, once again the receiving SMTP host accepts. The sending host then sends a command, stating that what follows is the text message, with, finally, an end of message string advising the completion of the message.

As you can see, in this scenario anybody with a network sniffer could pick up this traffic over the network, since it is all sent in clear text. Even worse, it would be quite simple for anybody to spoof a message on any SMTP server. It would be easy to initiate the communication with an SMTP host, and pretend that the mail was sent by someone else.

The example below demonstrates how simple this is. By connecting to the SMTP host using TELNET on port 25, and sending the commands that the receiving SMTP host expects, we can spoof an e-mail message.

```
Telnet <SMTP Host> 25

        HELO foobar.com
        MAIL FROM: <reverse-path>
        RCPT TO: <forward-path>
        DATA
        SEND FROM:<whatever-address-you-like>
```

POP and IMAP

The original POP3 specification does not contain any authentication methods; similarly to SMTP, the communication between a POP3 client and a POP3 server is sent in clear text. In fact, the commands USER and PASS are used for passing the user name and password for authorization to connect to a POP3 server for receiving mail. For more information on this, read RFC 1725.

Changes to these specifications include newer and more secure authentication methods like S/KEY, GSSAPI, APOP, and Kerberos V4. Currently, however, these methods do not appear to have general widespread support across the Internet.

IMAP4 also provides additional authentication mechanisms like Kerberos V4.

SSL

It is possible to use SSL to encrypt the session when communicating using POP3 or IMAP4. This would resolve the problem of weak authentication schemes that are used by POP3 and IMAP4.

Improvements to These Protocols

SASL

SASL, which stands for Simple Authentication and Security Layer, is specified in RFC 2222. It describes a method of adding authentication support to connection-based protocols. Each protocol that uses SASL includes a command for identifying and authorizing a user to a server and for optionally negotiating a security layer for subsequent protocol interactions.

Protocol designers who want to use the SASL specification to support authentication in their protocol define a SASL "profile" that describes how it is used in that protocol (e.g., the SMTP Extension for Authentication is a profile of SASL).

Note At the time of writing, Domino R5.0 employs SASL only for LDAP services. Domino uses SASL automatically if SSL with client authentication is set up on the server and if the LDAP client supports the protocol. No additional configuration is necessary.

ESMTP

ESMTP stands for Extended Services for Simple Mail Transport Protocol. It defines a framework for extending the SMTP service by defining a means whereby a server SMTP can inform a client SMTP as to the service extensions it supports. Extensions to the SMTP service are registered with the IANA. Examples of extensions to SMTP include, SMTP over TLS/SSL, and Delivery Status notifications.

SMTP Service Extension for Authentication

When a client submits a message to an SMTP Server that supports the SMTP authentication extension (AUTH=LOGIN), it will allow the client to authenticate the user to the server. Also the extension preserves the authentication identity when a message gets transferred from one SMTP server to another (assuming that both SMTP servers support the extension). However, as earlier mentioned, this user name password combination is only base64 encoded.

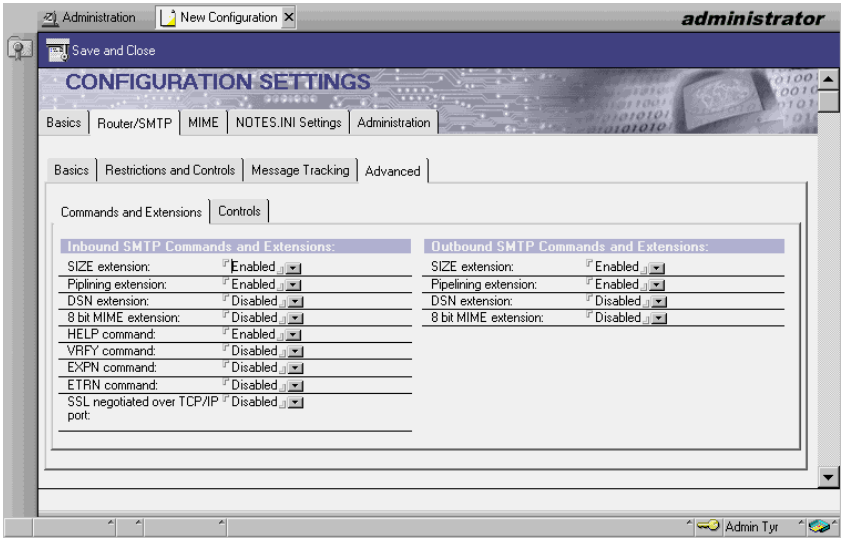
SMTP Service Extension for Secure SMTP over SSL/TLS

SSL or TLS is a popular mechanism for enhancing TCP communications with privacy and authentication. SSL/TLS is in wide use with the HTTP protocol, and is also being used for adding security to many other common protocols that run over TCP. TLS is very similar to SSL 3.0 with a few differences. Instead of using MD5, TLS uses the HMAC secure keyed message digest function. By securing STMP over SSL/TLS, only the communication between the two hosts is secure, it is not an end-to-end mechanism, and you are not securing the transport from the originating mail user agent to the recipient.

In fact, just because delivery of a single piece of mail may go between more than two SMTP servers, adding SSL/TLS privacy to one pair of servers does not mean that the entire SMTP chain has been made private.

Note The SMTP service in Domino R5.0 supports several SMTP extensions, including SSL negotiation over TCP/IP.

You can enable SMTP extensions using Domino Administrator or the Notes client as follows: From the Domino Administrator, select Configuration tab, and select Messaging - Configurations view. Click the Add Configuration or the Edit Configuration button to open the Configuration document. Click Router/SMTP tab - Advanced tab, and you will see the screen below, and be able to configure SMTP extension features on your Domino server.



Message Encryption

SMTP with support for the SMTP extensions can ensure that the initial client-to-server communication has been correctly authenticated. But that does not guarantee that the full SMTP hops will use that same authentication. Also we note that the message itself is not encrypted. However, this can be solved by using another SMTP extension which ensures the SMTP communications (Client/Server or Server/Server) are encrypted using Public/Private key pairs. Again, this does not guarantee that during its complete series of SMTP hops the message will be encrypted all the way to the recipient. Even if you could guarantee that your e-mail message was correctly authenticated with trusted SMTP servers, and fully encrypted during its hops to the client, there is always the possibility that the message was spoofed from someone else.

Thus, the only sure way to provide confidentiality, authentication and integrity of your e-mail messages is to make sure that the MIME content of your messages is encrypted. Until recently, there were two competing standards for achieving this: PGP and S/MIME.

PGP

PGP, which stands for Pretty Good Privacy, is a highly secure public key cryptographic system designed for sending secure mail anywhere around the world. It was developed by Mike Zimmerman, and is available for free on the Internet. PGP does not have key management capabilities, in fact its certificate structure is a very loose one. Instead of having authorities issue certificates to individuals, it works on a “web of trust” model, where certificates gain authority by being signed by people you know. A newer standard called OpenPGP, will permit a hierarchical approach to

accommodate Certificate Authorities (CAs), X.509 certificates, and other accepted standards. It is unclear how much broad support OpenPGP will gain since it uses the Diffie-Hellman algorithm. This will immediately make it incompatible with the 20 millions users currently using PGP, which employs RSA patented encryption algorithms.

S/MIME

S/MIME, Secure Multi-purpose Internet Mail Extension, is an e-mail security technology developed by RSA for encrypting and digitally signing e-mail messages. Lotus Notes and Domino R5.0 now support S/MIME. It is an IETF proposed standard which builds security on top of the industry-standard MIME protocol and a set of Public-Key Cryptographic Standards (PKCS).

A message is encrypted by taking the entire content of a message or just certain MIME parts and running it through an encryption algorithm that uses the public key of the recipient.

S/MIME v2 is slowly becoming the de facto standard across the Internet for sending secure mail, and currently S/MIME v3 is being ratified by a working group for publication and final ownership of the IETF. Although S/MIME v2 is widely implemented by vendors, it is not an IETF standard, and quite likely will not become one. It requires the use of RSA key exchange, which is a US-based patent, and uses weak cryptography (RC2/40). S/MIME v3 will resolve these problems by using stronger cryptography and a choice of encryption algorithms. At the moment, Notes R5.0 is implemented using S/MIME v2.

S/MIME uses a public-key algorithm for key exchange and for digital signatures. S/MIME recommends three symmetric encryption algorithms: DES, Triple-DES, and RC2. The adjustable key size of the RC2 algorithm makes it especially useful for applications intended for export outside the US where RSA is the required public-key algorithm.

How S/MIME Works

Let's look in a closer detail at how S/MIME works. Throughout this and the following sections, you will understand how Notes R5.0 implements and supports S/MIME.

S/MIME offers users the following basic features:

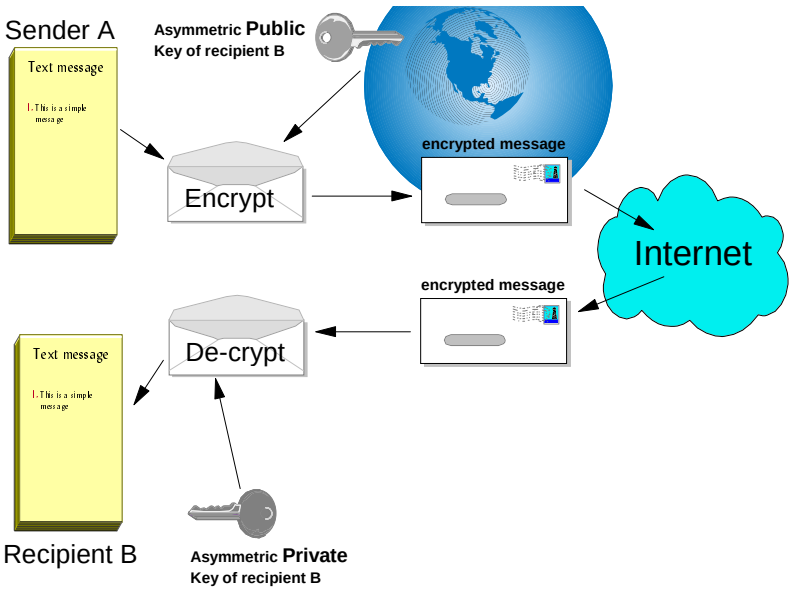
- Encryption for message privacy
- Tamper detection
- Signing - Authentication sender with digital signatures
- Interoperability with other S/MIME-compliant software
- Cross-platform messaging

With these features, you can be sure that:

- from the moment the message is sent to the moment that it arrives for the intended recipient, no one can see the contents of your message.
- that the message came from the person that the recipient thinks it came from.
- that the message has not been tampered with or changed on route to delivery.

Encryption for Message Privacy

For message privacy, or confidentiality, S/MIME uses asymmetric keys (public/private keys) to encrypt messages, the same technique that has been employed in Notes for years. To send an encrypted message, you need to obtain the recipients public key, and encrypt the message using this key. Since the only person who has its associated private key is the recipient, the message can be sent with safe knowledge that only the recipient will be able to decrypt this message.



If you look at the process described in the figure above in more detail, you will find that S/MIME uses a technique often referred to as a digital envelope, whereby the message is actually encrypted using the shorter symmetric cipher, the symmetric cipher is then encrypted using the larger asymmetric key, and sent along with the encrypted message.

You may wonder why this approach is taken. It is simply that it is far quicker to encrypt the whole message using the shorter symmetric key, than

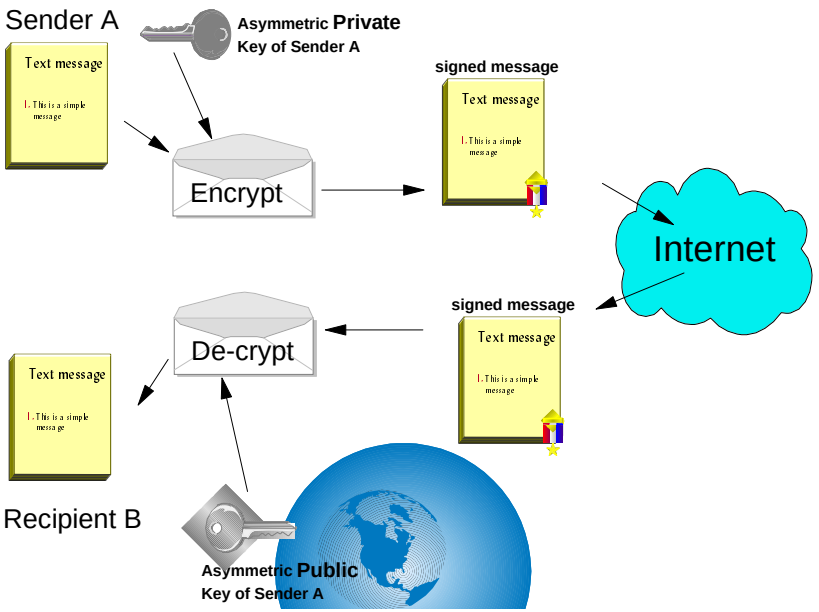
to encrypt the message using the longer asymmetric key. The message is still quite safe; this approach gives you the speed of symmetric encryption with the security of asymmetric encryption.

Tamper Detection

For tamper detection, or data integrity, S/MIME also can ensure that the message has not been tampered with. Again, it uses a technique that's already employed in Notes.

Signing - Authenticating Sender with Digital Signatures

S/MIME provides message signing by using digital signatures. By encrypting the message with the senders private key, and sending this encrypted message with a certificate (which vouches for the authenticity of the sender's public key). The recipient can decrypt the message with the sender's public key, which is freely available. Remember not to confuse this with message encryption, where the message is encrypted with the recipient's public key. In many cases, the sender may only want to sign the message, but it is also possible that the sender might want the message encrypted and signed, in which case the message goes through encryption with the recipient's public key and then goes through encryption with the sender's private key. The S/MIME specification does not specify the order in which the encryption must occur when both encrypting and signing a message.



In this figure we show only the process of signing the message; the message may also be encrypted as shown in the previous figure.

Details of Authentication

Let's look in closer detail at how the sender actually verifies that the message received is from whom it claims to be from.

As we said, the message is encrypted with the sender's private key. The sender also sends its X.509 certificate with the signed message, so now the certificate is nothing more than the signed public key of the sender. It is signed by another trusted party, a Certificate Authority (CA). What happens if you don't trust the CA that signed the sender's public key? Well S/MIME allows for that by employing what is known as a *chain of trust*. This means that when the sender sends the encrypted message and the sender's own certificate (which contains its public key signed by a third-party CA), it also sends the third-party CA's certificate. This may itself be signed by another CA or it may indeed be the root certificate. So as long as you can trust any of the CA certificates in that hierarchy, then obviously you can trust the CA that signed the sender's public key.

So how do you trust that CA in the first place? Well, held within your S/MIME client will be a list of CAs and their public keys that you trust; this is pre-built in the client to ease distribution of CA certificates. So you now have

- The signed message
- The sender's certificate
- You trust the CA that has signed the sender's certificate because you have its public key in your S/MIME client

You are now able to testify the validity of the sender, since the senders certificate that has been sent can be decrypted with the public key of the CA that are held in your S/MIME client.

If this is successful, then you can vouch for the authenticity of the certificate and its contents, the sender's name, the sender's public key, organization, country, and e-mail address. Now that you can trust the sender's public key, you can attempt to decrypt the message, to see if the message was signed by that same person.

However on the sender's certificate there is another piece of information, the e-mail address. This information is crucial in ensuring that e-mails are not spoofed, even if the message can be correctly decrypted with the sender's public key. If its associated certificate does not have a matching e-mail address then this would suggest that the message was sent from a different user. If this is the case, how trustworthy can the message or the sender really be?

As it happens, this may cause problems in the future as people tend to acquire multiple e-mail addresses. They may have a work address, a personal e-mail address, and perhaps a second work address if they are working temporarily at a customer location. Does this mean that we need to maintain three sets of public/private key pairs and certificates? This problem is further compounded by the fact that it is not easy to export S/MIME certificates from one client to another. S/MIME v3 hopefully will solve some of the problems regarding having multiple e-mail addresses associated with a common name. With regard to interoperability between clients from different vendors, this can be resolved only if customers always demand the highest level of interoperability.

Clear and Opaque Signing

If you attempt to send a signed message to a recipient that does not have an S/MIME client, there are two possible outcomes depending on the capabilities of the sending S/MIME client.

If the message is sent as *opaque* it means that the signature is sent as an application/pkcs7-signature MIME type. Thus, a non S/MIME-compliant client will not be able to read the pkcs7-signature type. The S/MIME client will first split the incoming message, and then check the validity of the signature.

If the message is sent as *clear* it means that the signature is inserted as part of a multipart/signed MIME object type. The signature is generated from the message by hashing it and the application/pkcs7-signature is inserted into the second part of the MIME type. This means that any receiving client will be able to receive both parts of the MIME type — the unsigned message and an attachment of the application/pkcs7-signature MIME type.

Interoperability with Other S/MIME-Compliant Software

The PKCS #12 standard specifies the format for certificate export and import. This is particularly important for ensuring that you have backup copies of your private key. Also if you need to send S/MIME e-mail from a different machine or from a different S/MIME client, you will need a simple way of taking your public/private key pair with you and then installing it in the new client.

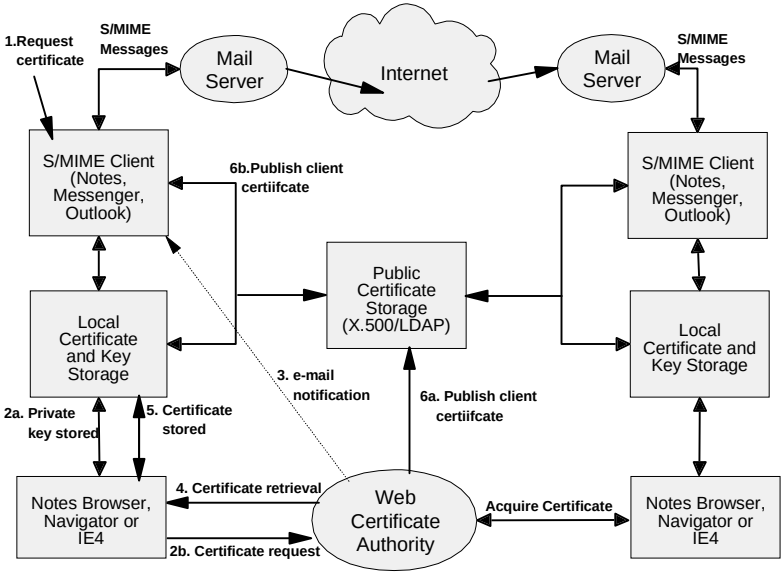
The purpose of the PKCS #12 standard is to provide interoperability of your private/public key pair and certificates with other S/MIME clients. This is quite important because if you request a certificate with Internet Explorer from a Web CA like VeriSign, you can only retrieve it into Outlook Express. Similarly, if you request a certificate with Netscape Navigator you can only retrieve it into Netscape Messenger.

Obtaining a Client Certificate for S/MIME

To be able to send signed mail using S/MIME, you will need an X.509 certificate for your client software. Current S/MIME-compliant clients, like Netscape Messenger, Lotus Notes, and Microsoft Outlook Express, provide the ability to generate a certificate request with a Web based CA. Once you have requested a client certificate, it will be installed in your S/MIME client so that you can sign any e-mail messages.

You will also need to make your certificate available to anybody who wants to send you encrypted mail. Encrypted mail messages addressed to you are encrypted with your private key.

The diagram below shows the typical process, at a high-level, for the flow of certificates and S/MIME mail messages as implemented by today's ubiquitous browser/e-mail clients.



Let's go through the steps needed to request a client certificate to be installed in an S/MIME client.

1. From within your S/MIME client (Lotus Notes, Netscape Messenger, or Microsoft Outlook Express), you request a client certificate. The Notes browser, Netscape Navigator or Internet Explorer 4 (depending on which S/MIME client is being used) will prompt the user to fill in a client certificate request form at the Web site of a trusted CA.

2a. As the request is being submitted, it will trigger the browser to generate and store a private key locally (this process differs if it is Internet Explorer; see "Serving Certificates to Browsers" earlier in this chapter).

2b. A corresponding public key is included in the HTTP header as part of the certificate request (in PKCS #10 format) to the Web CA.

3. The CA processes the request and returns instructions on how to pick up the certificate via e-mail. The instructions state a URL and a pickup ID (PIN) where the signed client certificate can be picked up.

4. You connect to the stated URL, enter the PIN, and pick up the signed certificate.

5. It is then installed in the S/MIME client.

6a. You may go one step further and publish your certificate by sending it to one of the public directory providers. Often the CA's themselves will have this facility available.

6b. Alternatively, you can also use your S/MIME client to publish your client certificate to one of the public directory providers (not available with Internet Explorer 4/Outlook).

Obtaining a Recipient's Certificate for S/MIME

In Netscape Messenger (4.x) and Microsoft Outlook Express there are a couple of mechanisms for obtaining a recipient's certificate.

The first is by having them send you a signed message. When you receive this message, these e-mail clients will automatically add the sender's certificate to the list of stored certificates. Similarly, if you send signed mail to another Netscape Messenger or Outlook Express user, they will obtain a copy of your certiticate.

The second method is by providing access via LDAP to search online directories such as Four11, Bigfoot, Switchboard, and so on. If the required certificate is stored in one of these directories, you will be able to add it to your address list.

Using Lotus Notes R5.0 as S/MIME Client

Once your Notes client is connected to a CA-based infrastructure, the actual operation of sending/receiving S/MIME messages is simple and as easy as using Lotus Notes mail. In this section, you will learn how Notes and Domino R5.0 have been seamlessly integrated with S/MIME.

How Notes R5.0 Implements S/MIME

For traditional Notes users who understand Notes certificates and Notes ID files, the concepts of encrypting and signing should be nothing new.

With R5.0, the Notes ID file which contains the native Notes certificates will also be used as the container for storing X.509 certificates. When a certificate

is requested from a Web CA, it is requested via the Notes browser (or Web Retriever). Once it has been approved, it is stored within the Notes ID file.

Notes has a facility for creating safe copies of ID files, this is basically the public key and the associated signed certificates. Currently, there is no facility for creating 'safe copies' of X.509 certificates, so you will not be able to import or export S/MIME client certificates in or out of Notes. The PKCS #12 standard will hopefully solve this issue.

To sign e-mail messages with S/MIME, you have to install your own X.509 certificate in your ID file. You can use either a certificate issued by Notes, a certificate issued by Domino CA, or a certificate issued by another commercial CA.

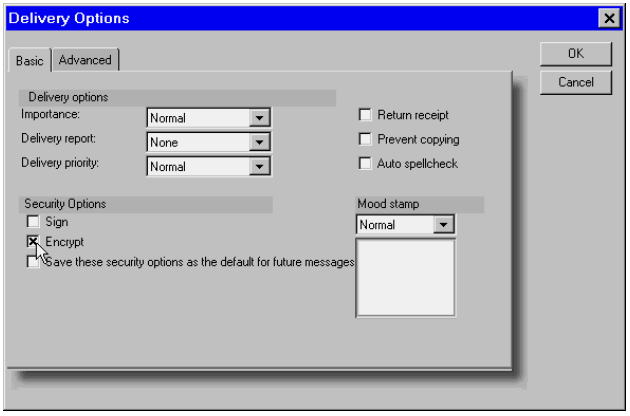
To be able to verify the signatures on an S/MIME signed mail you receive, you need a certificate from the trusted root CA for the signer, or a cross-certificate to the recipient's certificate in your Personal Address Book or Domino Directory.

Prior to encrypting a message, you will need to obtain the other party's certificate. Your S/MIME client will encrypt the message using the recipient's public key. In Notes R5.0, recipients' client certificates are stored in the Domino Directory.

Sending and Receiving Encrypted S/MIME Messages

When you attempt to send an encrypted message, the recipient's X.509 certificate is used. That is, you must have access to that certificate to send an encrypted message. The recipient's certificate must be registered in your Personal Address Book or Domino Directory.

To send an encrypted message, click Delivery Options and select Encrypt.



Or, if you want to encrypt all the mail messages you send, choose File - Preferences - User Preferences - Mail and News - Encrypt Send Mail.

If your mail is addressed to an Internet e-mail address or the message is in MIME format, then Notes will try to use S/MIME encryption for the message. Otherwise, Notes will encrypt it using the native Notes certificate. In either case, if Notes cannot find the recipient's certificate, you will see an error message.

Sending Signed S/MIME Messages

Users can sign individual mail messages or sign all mail messages that they send, including encrypted messages. Before signing a message, make sure you have obtained your own X.509 certificate in your Notes ID file.

To sign an individual mail message, when you finished composing the mail message, click Delivery Options and select Sign. Or, to sign all mail messages you send, choose File - Preferences - User Preferences - Mail and News - Sign Send Mail.

Send your message. If your message is addressed to an Internet mail address or the message is in MIME format, the message is automatically signed for S/MIME.

Receiving Signed S/MIME Messages

Upon receipt of a signed e-mail, Notes will try to verify the validity of the signature.

If you "trust" the signing certificate, that is, if you have the signer's certificate or an Internet Cross-Certificate to the sender, you will receive a message in the status bar indicating the validity of the signature. For example:

```
Signed By: Bob, at 10:52 AM, According To: TestCertAuthority
```

If you don't "trust" the signing certificate, you will receive a prompt to create an Internet Cross-Certificate on demand. You can select the subject name of the certificate in the message that you wish to trust.

Note Signed S/MIME messages contains the certificate chaing of sender and signers. The resulting Internet Cross-Certificate is stored in the receiver's Personal Address Book. By creating the cross certificate, you are asserting that you trust a certificate contained in the S/MIME signed message. Signature verification can then proceed.

Also you can manually store the sender's address and X.509 certificates in your Personal Address Book. When viewing S/MIME signed mail, choose Actions - Tools - Add Sender to Address Book. Note that this certificate is not an Internet Cross-Certificate, that is, it is not used when sending or receiving signed S/MIME mail. It is used to encrypt messages from you to the sender.

Summary

The Internet has enormously widened communication between different people in different organizations in different countries, but it has also brought with it the problem of "Who can you trust?" The use of X509 certificates in secure connections (SSL) and secure e-mail (S/MIME) addresses this problem of trust. For Notes and Domino R5.0, we have shown how support for X509 certificates is implemented. We also have described how X509 certificates are acquired and used in a Notes/Domino security infrastructure.

Chapter 5
Domino and Firewalls Revealed

At this point, you should have a sufficient understanding of the mechanisms in Notes and Domino to achieve the goals and objectives of your security policy. If your security plan is consistent and you have diligently applied the knowledge gained from the previous chapters, you have very few things to worry about.

If you now put your servers on the Internet, where any system may be the target for crackers, protection of those servers becomes all-important. Your security policy must now be augmented, since your servers will be vulnerable to new and different types of attacks. These attacks fall into two different categories: passive attacks and active attacks.

A passive attacker could be described as someone using a trace tool or protocol analyzer to look at the traffic to and from your Internet-connected systems. These can be hosts for your Web site or specialized communications hosts that aid in the transfer of electronic messages. A passive attack is similar to having someone eavesdrop on your phone conversations. Since the Internet is completely unregulated, you should assume that everything you communicate on the Internet is intercepted and read by someone.

An active attack could be described an attempt to compromise your server in some way that will exploit the vulnerabilities of the transport protocol and the software programs that enable the connectivity these provide. This kind of attack is generally mounted by crackers. Hackers are quick to point out that they do not "attack" systems, since they are merely using their technical prowess to test the defenses of your system. Hackers should be considered as much of a threat as crackers since as we noted in Chapter 2, they can open holes or back doors to your server or computer system and make it easier for crackers to do their damage. For this reason, both crackers and hackers will be referred to as attackers in this chapter.

Although the objective is to protect servers that are connected to the Internet, it is really the computer systems behind the server that need to be protected. An Internet server is, by its nature, exposed to some degree of abuse. However, by opening a connection to the Internet you want to make sure that you don't open your private network to attack.

The goal of this chapter is to explain how you can protect the points of entry, where your private network is connected to the Internet. Providing this kind of protection is the job of a number of tools and mechanisms, which are generally grouped together under the term "firewalls."

Before we go further, you should note four important things:

1. The goal of this chapter is to ensure that you have an understanding of firewalls and how they apply using Notes and Domino. If you want a "How To" book that goes into great detail about the specific installation and configuration of a firewall, we suggest that you consult another redbook titled, *Protect and Survive, Using IBM Firewall 3.1 for AIX*, SG24-2577-02.
2. Firewalls require a solid knowledge of IP protocols, as well as addressing and router configurations. If you do not know the fundamental differences between TCP and UDP, it would be advantageous to read another redbook titled, *TCP/IP Tutorial and Technical Overview*, GG24-3376-05. There is an overview of TCP/IP services as they pertain to firewalls in the section "TCP/IP services" later in this chapter.
3. It is important to read this chapter with your computer security policy in mind, because it will explain how to define your firewall and the security objectives that the configuration should meet.
4. A firewall is not a trivial exercise; if you find that the security policy requirements force you towards a complex firewall configuration, it may be wise to get some outside help to assist you. Consider outsourcing some of the work to a competent, trusted specialist. This will ensure that you will have the expertise you need to ensure that the firewall is properly designed and satisfies your stated security needs.

Firewall Basics

To understand how a firewall works, consider this example: Imagine a building to which you want to restrict access and exercise control over who will enter it. In designing the building, you specify one lobby as the only entrance point. In this lobby, you have receptionists to welcome, security guards to monitor, video cameras to record, and badge readers to authenticate all the people who enter the building.

This works very well to control a private building. But imagine that a non-authorized person succeeds in gaining entrance. To protect the building against any untoward actions by this person is extremely difficult. However, if you can limit this unauthorized person's movements you at least have a chance of detecting any suspicious behavior and repairing any damage.

When you are defining your firewall strategy, you may think it is sufficient to prohibit everything that presents a risk for the organization and allow the rest. However, because of new attack methods, you may not be able to prevent every attack and, as in the example of the building, you need to monitor for signs that somehow your defenses have been breached. Generally, it is much more difficult and costly to recover from a break-in than it is to prevent it in the first place.

Even with that in mind, a lot of people are misled by the term *firewall* because, in practice, an Internet firewall is not necessarily one device nor does it necessarily perform one function. It is true that the classic solution uses a single device called a screening router, but that is not sufficient today to ensure security. It is, nevertheless, the starting point for firewall defenses.

You can build upon this basic configuration to include a separating host which delineates where the Internet ends and where your internal network begins. A firewall can also be a proxy server, which is defined with a specific application protocol in mind.

The best strategy is to permit only the applications you have tested and in which you have confidence. If you follow this strategy, you need to exhaustively define the list of services you must run on your firewall. Each service is characterized by the direction of the connection (from in to out, or out to in), the list of users authorized, the list of machines where a connection can be issued, and perhaps the span of time per day you authorize this service.

This section will define firewalls and provide you with a basic explanation of what firewalls can and can't do.

What Is a Firewall?

A firewall is actually a system or a group of systems that provide some form of access control between two networks. A firewall can be seen as having two functions:

- Permit traffic flow
- Block traffic flow

Depending on what you want your firewall to do, the emphasis for your firewall will either be to permit traffic above all or block traffic above all. Most companies choose to do the latter.

Based on your security policy, you will be able to define the type and the extent of access control to your internal network that the firewall should enforce.

Ideally, the best firewall configuration is one where your computer system is connected to the Internet via one device. If you manage to achieve this, then you have an effective *choke point* where you can control and police the traffic going to and coming from the Internet.

A word of caution: You should design and implement your firewall very cautiously. If you do not pay close attention to the manner in which you restrict information going through your firewall, you will be, in fact, redefining the company's computer security policy and could create added security risks where there should be none. Consequently, your firewall architecture should be planned with extreme care, more than any other aspect of your security architecture.

Basic Functions of a Firewall

In practice, a firewall can perform a number of different functions. For example it can do the following:

- Limit access from the Internet so that only the services that you intend to offer can be accessed.
- Provide controlled access to Internet services for users within a private network.
- Perform a gateway role for applications that store and forward information, such as e-mail and news services.
- Provide an encrypted tunneling capability to pass application traffic securely across the Internet.
- Act as a screen to hide or disguise the real content of a private network.
- Act as a gateway between different IP network addressing schemes. (This is not really a security function, but it is something that a firewall is frequently called on to do, because of limitations on the number of available legal IP address ranges.)

This is a formidable list of roles and the result is that few firewall implementations are alike. However, these implementations share many things in common, which we will cover in this chapter.

Before going into a specific overview of firewall components and architecture, let's cover what level of protection a firewall can provide, as well as the protection a firewall cannot provide.

The Measure of Protection a Firewall Can Offer

Firewalls offer different types and levels of protection for your computer system. Simple firewalls permit specific traffic to go through them.

There are some firewalls that permit HTTP traffic to go through them, while other firewalls permit only e-mail traffic to pass. There are also firewalls that are not so specific and only aim to protect against known methods of attack or know vulnerabilities in the computer security architecture. Finally, still other, more sophisticated firewalls block traffic coming in from the outside and headed towards the inside, but permit users that are using the inside computer system to communicate freely with those people outside, on the Internet.

Generally, firewalls prevent unauthorized users from accessing machines on your computer network by permitting only authenticated access to the resources of your internal computer system.

No matter what type of firewall you put in place, a nice feature is that above the security services provided, you can also log and audit all traffic passing through the firewall. This permits you to determine if there were any attempts made to circumvent your firewall, and if so, helps you to determine the system of origin and the nature of the attack.

The Measure of Protection a Firewall Cannot Offer

A computer system is a collection of resources, the firewall being only one such resource. It is important to make sure that all resources, including dial-up modems, are examined for security holes and back doors. However, firewalls are not foolproof devices. As new, imaginative security services are offered, there are also new, imaginative ways to circumvent them.

Firewalls can only prevent attacks that are attempted against the firewall. If the attack circumvents the firewall altogether, there isn't much the firewall can do. In other words, if an attacker can find an alternative, unsecured, and uncontrolled access point, your whole firewall effort will have been wasted.

As an example, many companies go to great lengths to protect against sensitive information going through the Internet, while leaving gaping security holes in other areas, such as modem dial-up remote access, for example.

Firewalls cannot do anything to guard against the people working in your company. If you have an industrial spy in your midst, chances are that this person won't transmit information through the firewall, but via more traditional means, which can take the form of written letters, phone conversations, faxes, or by backing up the information on floppy disks or magnetic tapes.

Firewalls also cannot compensate for lack of training or for carelessness. Employees should be trained to ensure that they do not, unwittingly, provide sensitive information over the phone or by any other means that have nothing to do with the firewall.

Neither can firewalls protect you from other forms of attack, some of which are listed below:

- Virus — a program that propagates from one computer to the next by attaching itself to other files. Once the virus has arrived at its new location, it infects the new computer and tries to move on from there in the same manner as before.
- Trojan horse — a program that hides itself in another program that people tend to trust. When a user runs the trusted program, that user also activates the hidden Trojan horse program.
- Logic bomb — code that is inserted into an application or operating system that causes it to perform some destructive or security-compromising activity whenever specified conditions are met, such as on a Friday the 13th, for example.
- Worm — a program that actively propagates itself from one computer to another. These types of programs are very rare, but when they strike, they can have a major impact on the systems they infect, such as the Internet Worm of 1988 unleashed by Robert T. Morris, Jr.

As part of any computer security policy, virus-scanning software should be implemented so as to cover the above-mentioned forms of attack. Don't rely on your firewall to detect all known forms of virus attacks, since the vast majority of viruses come from floppy disks exchanged at work or brought in by employees from home.

Finally, if there are reasons to suspect that the firewall's security has been compromised, the fastest way to plug the hole is to unplug the firewall. This might cause some outage to the services you are providing through your firewall, but compared to sensitive information leaking out and potentially ruining your company, this is the lesser of two evils.

Access Needs Versus Security Requirements

In the end, a firewall will force you to make a tradeoff between *access needs* and *security requirements*.

Access needs can be thought of in terms such as *transparency* (there should be no security or added hoops that users need to jump through), *connectivity* (the communication system should not be complex to use) and *performance* (the system should be fast and responsive).

Security requirements can be thought of in such terms as *cost* (the system should not be expensive), *complexity* (the system should be simple) and *risk* (the system should be safe).

The idea here is that the more you make a system secure, the more you need to curtail access. Inversely, the more you make a system accessible, the more you need to curtail security. You might have a system that is transparent but costly, or you could have a system that is safe but slow.

In the end, it will be your security policy that will dictate what matters the most, access or security, and once that priority has been determined, whatever remains of the other will be what your users (and ultimately your company) will have to live with. This is yet another reason why your security policy is the most important step, as the rest of your security work flows from it.

Firewall Configuration and Architecture

Armed with your company's computer security policy and a good understanding of what a firewall can and cannot do, you are now ready to design and implement your company's firewall. If you already have a firewall in place (that you have installed or have inherited from someone else), you are ready to validate the design of this firewall and perhaps make any appropriate changes you feel are necessary.

TCP/IP Services

Before detailing firewall components and their characteristics, it is an opportune place to lay the groundwork for TCP/IP services as they pertain to firewalls. This assumes a good understanding of TCP/IP.

TCP/IP uses a number of services to assist in mapping IP addresses to host names. These must be well understood; otherwise these services might not be rendered properly when traversing a firewall and might result in an inability to access key servers.

TCP/IP Name Services

The Domain Name System (DNS) is a client/server mechanism which provides a number of services, including the following:

- Host name to IP address mapping
- Network address to host name mapping (optional, but typically used by firewall proxies)
- Mail routing information via Mail Exchange (MX) records (used by SMTP)

For example, when your computer needs to access a remote host (e.g. www.lotus.com), a client called a resolver, which is normally part of your TCP/IP setup, contacts a DNS server (also known as a name server). The resolver gets the IP address of that specific host, upon which the DNS server provides the related IP address of that host, if it is present in the DNS database. This database is the Internet distributed database, which allows for local updates while providing global consistency.

DNS is one of the most important TCP/IP services and thus it is essential to consider the implications of this service in a firewall configuration. For example, you should be extremely cautious about the manner in which people can access your internal DNS server from the Internet. The DNS server could provide valuable clues on servers or other devices containing key information that might become the target of active attacks that can cripple your entire computer system.

Finally, note that TCP/IP Name services include the Microsoft Windows Internet Name Service (WINS) and the Sun Network Information System (NIS), formerly known as the *Yellow Pages.* These are vendor specific implementations of Name services optimized for the vendor's brand of networking products.

Dynamic Host Configuration Protocol (DHCP)

DHCP, the Dynamic Host Configuration Protocol, is defined formally in the IETF-ratified RFC 1531. This protocol is an extension and improvement of the *bootp* protocol.

The main advantage of DHCP is that it does not require a network manager to set up a table linking MAC addresses to IP addresses, as the *bootp* protocol does. A "MAC address" is the unique hardware address assigned by the manufacturer of the token ring or ethernet adapter card. DHCP allows the network manager to specify a range of available IP addresses without having to tie each one to a specific MAC address. The DHCP server leases an IP address to each client, and dynamically maintains a table linking the client's MAC address to its leased IP address. For clients requiring a permanent IP address, DHCP can also allocate a fixed IP address based on each client's MAC address.

Thus, with IP address leasing, a client can lease an IP address for a certain period of time. The client can renew the lease if required or allow the address to return to the pool at the expiration of the lease. The server can then allocate the address to another host for the lease period. For clients with an infrequent need to communicate on the network, a short-term lease can reduce address constraints.

When using DHCP in a firewall configuration, it is important to consider how DHCP provides addresses for servers. Namely, that:

- Dynamic changes to the IP address of a server can cause Name Lookup failures.
- There is no standard mechanism for synchronizing DHCP addresses with DNS, so if you change an address in DHCP, you must manually update DNS.

Therefore you should be very careful and proceed with caution. DHCP is a great TCP/IP service, but it could cause many problems if not used properly.

Network TCP/IP Addressing

There is some key TCP/IP addressing information you should be aware of when planning a firewall configuration. For example, it is possible to have one host name map to multiple IP addresses. This is a handy way of configuring a hunt group of servers within your firewall to eliminate single points of failure. This feature is called a *multi-homed host*.

- It is possible to have one IP address that maps to multiple host names. This feature is called *aliasing*.
- There exists a special bank of addresses (which subnet starts with 10, namely 10.*.*.*) that offers key advantages, above and beyond being useful when registered addresses are unavailable. For instance:
  - They are configured within the TCP/IP addressing scheme as specifically being non-routable, which means that a router will not pass the packets to another subnet.
  - They are useful to protect/isolate the internal network.
  - They may be used with Network Address Translation (NAT) services for Internet access. NAT is explained below.

Network Address Translation

Even though there is a maximum of 4,294,967,296 IPv4 addresses that can exist on the Internet, there is a shortage already.

This shortage is due to the fact that addresses are granted in address spaces called *Classes* (referred to as Class A, B, and C). Each class of IP address has a network portion and a host portion. These classes determine the number of networks in relation to the number of hosts present per network. They are as follows:

- 126 Class A addresses
- 16,382 Class B addresses
- 2,097,150 Class C addresses

These classes of addresses were allocated to companies that use them on their internal TCP/IP networks, but in most cases, not on the Internet. This makes for a lot of addresses allocated, but not many used on the Internet, considering that the shortage exists with only 100 Million Internet users. These allocated addresses cannot be used by others, despite the fact that they are not used, because of the attribution system in place.

IPv6 aims to correct this situation, but it is expected to take some time before a transition is made from IPv4 to IPv6, mainly due to applications that communicate over TCP/IP. Notes and Domino have already been architected to port over to IPv6, but this is not the case for most applications that exist.

To combat this, Network Address Translation (NAT) was developed and became the IETF-ratified RFC1631 in May 1994. This specification details NAT as a mechanism used to circumvent the IPv4 address shortage.

NAT offers two methods of address translation: *static* and *dynamic*.

Static address translation is used when the networks are of equivalent size. For example, if you need to translate all IP addresses from network 2.24.118 to IP addresses in network 160.106.65 (the subnet mask would be 255.255.255.0 for both). In this example, the IP address 2.24.118.197 would be translated as 160.106.65.197.

Dynamic address translation is used when the networks are of different sizes (like translating from one class of network addresses to another). Another reason for using dynamic address translation is for security; with static address translation an attacker could easily determine the corresponding, translated IP address, whereas with dynamic address translation, connections using different and changing IP addresses are created.

It is this security feature of NAT that we will explore as part of a firewall configuration. We will cover an example of a firewall using NAT in a subsequent section of this chapter.

Firewall Components

There are specific types of firewalls and there are a number of elements that are common to most firewall designs. The three components are:

- Packet Filters
- Circuit Level Proxies
- Application Level Proxies.

The table below summarizes the key characteristics of each firewall component.

Firewall Component	*Services Description*
Packet Filter	Does not understand the application Cannot Proxy at all Will restrict at the network level
Circuit Level Proxy	Does not understand the application Will proxy at the session level
Application Proxy	Understands the application Will proxy at the application level

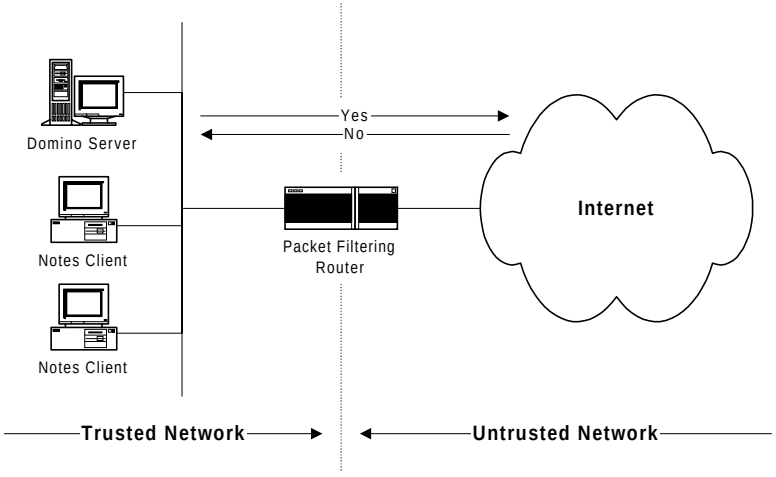
Each component will be covered in further detail below. We will provide additional explanations of the advantages and disadvantages of each configuration.

Packet Filters

Packet filters intercept IP packets in transit and test them against a set of filter rules. This forms the most basic protection mechanism. In fact, filtering is often performed by a router that analyzes the packets it routes (*a screening router*), rather than a dedicated firewall machine. Packet filters either allow a packet to pass on to its destination or block it, based on criteria such as the following:

- The source and destination IP addresses
- The origin of the packet (Did it come from the Internet or from a private network?)
- The client and server port numbers
- The session-level protocol carried by the packet (UDP, TCP, ICMP, etc.)

The figure below illustrates a simple packet filter.



The objective of placing a packet filter between a server and the Internet is to limit the scope for an attacker to probe the server system. As such, it is a good idea to employ a packet filter as a screen in front of a World Wide Web server, even in cases where the server is not connected to a private network.

Typically, the session will pass through a local IP router, so normally the filtering capability of the router is employed. The desirable filter characteristics to protect a Web server such as Domino are listed in the table below.

Filter Requirement	*Details*
1. Allow sessions only to the publicly accessible servers	The filters should reject session requests for any IP addresses other than the addresses of the servers that you want to be accessible from the Internet.
2. Allow sessions only to the publicly accessible IP services	In addition to restricting session requests to the publicly accessible servers, the filters should also reject requests for anything other than the IP services that you want to offer. For a Domino server these are: • TCP port 80 for HTTP (normal Web access) • TCP port 443 for SSL (secure Web access) • TCP port 1352 for Lotus Notes access (if you want Notes clients or servers from the Internet to have access to your server)
3. Prevent IP address spoofing	The basic filters described above rely on extracting the source and destination addresses from each IP packet. But an attacker can send packets that appear to come from one of your trusted IP networks and try to gain access. The filters should reject such packets if they appear on the Internet side of the screening router.
4. Prevent well-known port misuse	Attackers sometimes use a technique called source porting to try to evade IP filters. This means using a well-known port which is normally a server port as a client port (for example, using TCP/80 as a source port). Directional TCP filters can prevent this.
5. Log packets that are rejected by the filter rules	If someone is attacking your site you want to be able to detect it as early as possible. Thorough logging is one of the weapons that helps this detection.

Attackers will employ tools such as port scanners to probe a server for unsecured services or known application bugs. By placing a screening router to do packet filtering in front of the server, you can frustrate this kind of attack. For example, you could place a Web server behind a screening router which prevents access to all except the HTTP port (TCP port 80).

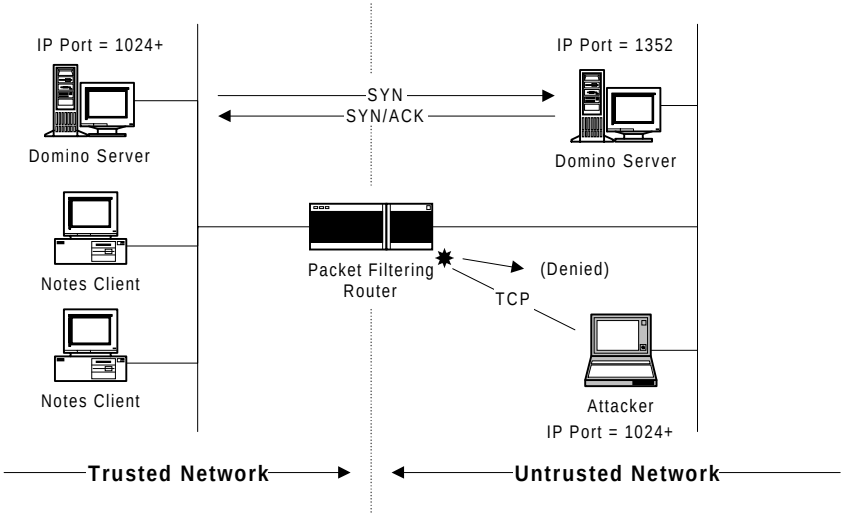
Note Different routers have more or less sophisticated filtering functions, so it may not be possible to implement everything shown in the table above. What level is acceptable to you depends on what the cost of a server break-in would be.

Directional TCP Filtering

To prevent an attacker from using a well-known port as a source port, filters exploit a feature of the TCP/IP session setup handshake.

When a client tries to establish a new TCP session it sends an SYN packet and the server responds with its own SYN packet, which has the acknowledgment (ACK) flag set. From that point in the session, each packet contains an ACK. To put it another way, the only packet in the session that does not have an ACK is the initial session request.

IP filters can therefore look for the ACK flag and use it as an indicator of the direction in which a session was initiated. The figure below illustrates how such a filter can frustrate an attacker using a well-known port (in this case the Notes server port) as a source port.



How Much Protection Do Packet Filters Offer?

Packet Filters make the attacker's life more difficult, but packet filtering only provides part of the solution in most cases. Attackers can use spoofing techniques to bypass some of the protection that a filter offers. Although this may not allow access to restricted services, it can give them a window into the private network that a firewall is supposed to hide.

Filters do nothing to prevent some common modes of attack, such as the following:

- Attacks that exploit some bug or administrative error in the server itself.

  The best way to counter this threat is to be vigilant in setting up the server and to monitor sources of information about common problems, such as the World Wide Web security FAQ, which can be found at the following URL:

`http://www.genome.wi.mit.edu/WWW/faqs/www-security-faq.html`

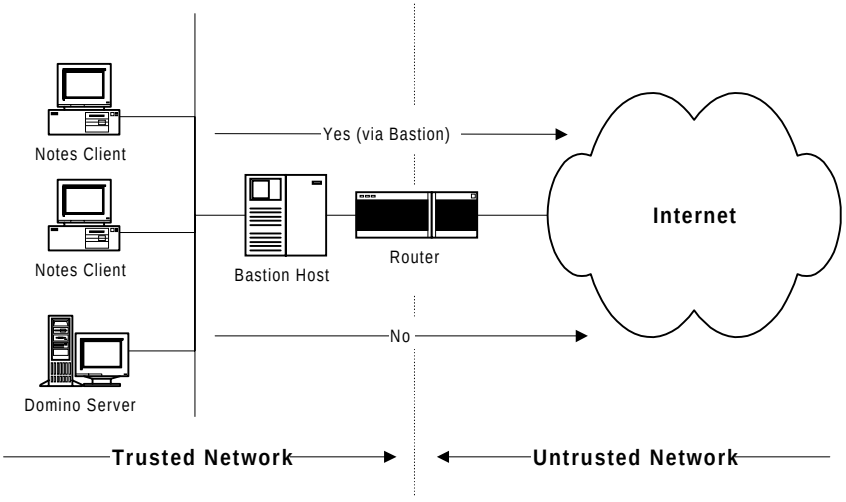
- Denial of service attacks, which usually intend to bring your server to its knees.

  These include bombarding it with unconsummated SYN packets and sending malformed IP packets that the server cannot handle (such as over length ping packets).

Consider packet filters as being only a part of your armor. In most cases, filters have to be combined with some kind of proxy service for a fully secure environment. We will explain proxy services later in this chapter.

Bastion Hosts

A Bastion Host is a system that has been hardened to resist attack by breaking the IP forwarding of packets (it is sometimes called a sacrificial host). A Domino passthru server can be a Bastion Host, for example. Unlike packet filters that run a special-purpose or ROM-based operating system, the Bastion Host generally runs a common operating system (AIX, Windows NT, and so on).



The Bastion Host is placed between the trusted and untrusted network where the IP forwarding is broken, meaning that no IP packet can go through this machine. As the routing is broken, the only place from which you can access both networks is the Bastion Host itself. Therefore, only users who have an account on the Bastion Host, with a double identification (one for the Bastion Host and one for the remote host), can use services on both the networks, as shown in the figure on the previous page.

This has some disadvantages, because it may become necessary for the Bastion Host to support many users. It is important to enforce good password control here. If an attacker manages to break into a user ID, that attacker can then impersonate the user and get into the private network. Besides this security point, supporting a great number of users will require a machine with a great amount of computing power. To avoid having users logged in to this machine and to reduce the load on the machine, more general-purpose bastion applications exist, such as the SOCKS server, which will be covered later in this chapter.

Proxy Services

A proxy service is a firewall function that accepts session requests from one side of the firewall, checks whether they are permissible and, if so, passes them on to the target system.

In addition to allowing effective authorization checking, a proxy also breaks the session at the firewall boundary. This means that an attacker cannot trace the session back to its source, thereby greatly reducing the size of their target.

Proxy services operate both at the circuit level and at the application level. The basic difference is that an Application level proxy understands the application data, whereas a Circuit level proxy doesn't. Both enable you to specify filtering rules for data, the criteria being as follows:

- Circuit level proxy: source/destination address and/or source/destination port;
- Application level proxy: source/destination address and/or source/destination port as well as an application-specific criteria.

We will look at configurations based on a proxy service in the next section.

Gateway Services

A firewall is often required to act as an application gateway for services that use the Internet. For example, a firewall may act as an SMTP mail gateway.

The objective of these services is to provide access to these Internet services for local users, but to only reveal the minimum necessary information about local users and systems to the rest of the world.

We will also look at configurations that provide gateway services in the next section.

System Protection, Logging and Auditing

A firewall must provide good logging and auditing facilities, so that attack attempts can be detected and repulsed.

Make sure that the audit logs are well-protected, otherwise it would defeat the whole purpose of having a log. The attacker could penetrate the system, effect the damage or steal the desired information and, upon exiting, could modify the log so as to cover their tracks.

Types of Firewalls

From a design point of view, there are two types of firewalls. There are *Circuit-level* firewalls and *Application-level* firewalls.

Circuit-Level Firewalls

Circuit-level firewalls offer security services based on the data exchanged at the protocol level (i.e., TCP/IP). The firewall analyzes each packet and determines the validity of the information contained within, namely the addresses and ports of both the source and destination.

Since all packets pass through a router, simple firewalls can be implemented using routing rules at the router. More advanced routers expand on this basic functionality and maintain additional information about the data passing through them, which allows them to apply security rules more knowledgeably.

The main advantage of circuit-level firewalls is that they are transparent to users, since they operate at a level lower than the application level. Consequently, they are very fast compared to application-level firewalls.

The disadvantage of circuit-level firewalls is that they route network traffic directly through, as per the security rules defined for it, so to connect one to the Internet requires an assigned set of IP addresses.

Besides a simple packet filtering firewall, which we have seen in the previous section, there are two more advanced types of circuit-level firewalls:

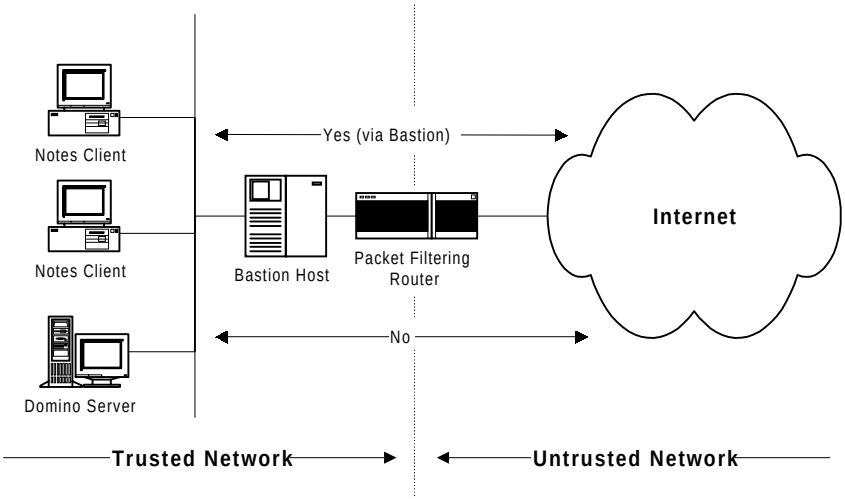
- Screened Host firewall
- Screened Subnet firewall

These use a Bastion Host as part of the firewall configuration.

Both the Screened Host firewall and the Screened Subnet firewall are explained in further detail below.

Screened Host Firewall

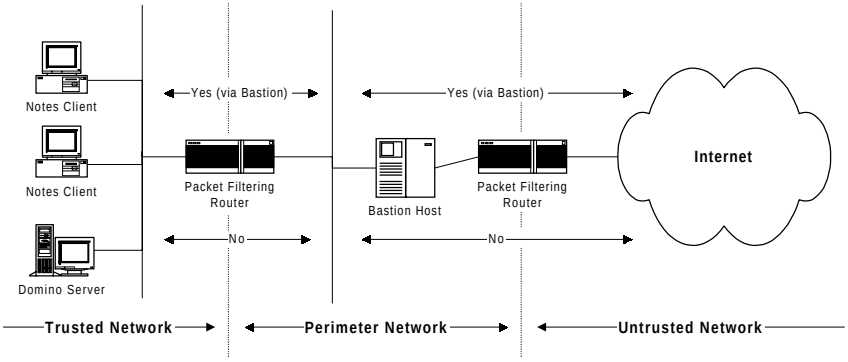
The figure below illustrates the configuration and the elements of a Screened Host firewall.



In this configuration, the router controls access to the Trusted Network by means of packet filtering. Only traffic to and from the Bastion Host is permitted. Direct traffic to and from the Domino server depicted here would not be authorized unless it involved the Bastion Host. This means that even though the PCs and server are on the same network they should, given the risks identified, be secure. The main disadvantage with this type of firewall is that if somehow an attacker manages to attack the Bastion Host and take control of it, your entire network is then compromised.

Screened Subnet Firewall

The figure below illustrates the configuration of a Screened Subnet firewall.



This configuration is roughly the same as the Screened Host firewall, except for the fact that this is effectively a combination of screened host firewalls.

In this configuration, the first router only permits traffic from the Internet to the Bastion Host located on the Perimeter Network. The second router only permits traffic from the Bastion Host to resources in the Trusted Network. Any other traffic to and from any other resources would be unauthorized.

This firewall is by far the more robust, since there are more devices and steps that a person outside must go through in order to mount a successful attack. Additionally, there are ways to distribute firewall services within the perimeter network in order to eliminate any single point of failure. However, the disadvantage with this solution is that it is difficult to implement properly and is complex to configure.

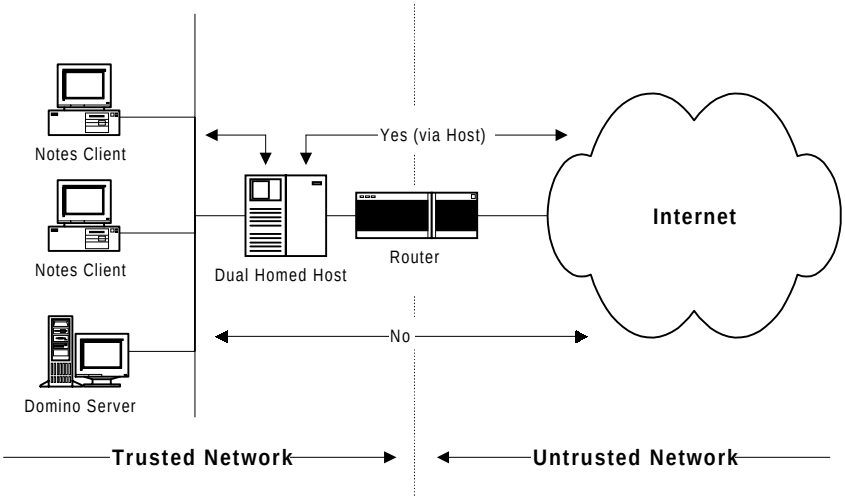
Note The Perimeter Network is also called a "demilitarized zone" or DMZ. We will cover this in further detail in the next section.

Application-Level Firewalls

Application-level firewalls are more elaborate types of firewalls, which prevent any kind of traffic between networks. These Firewalls run Proxy services and offer logging and auditing of traffic passing through them.

Dual Homed Host Firewall

An example of an application-level firewall is a dual homed host firewall. The figure below illustrates the configuration of this type of firewall.



In this configuration, the application-level firewall is a highly secure host (somewhat like a Bastion Host) that basically runs Proxy Services. The reason it is called a *dual* host is because the host contains two network interface cards (NICs) and blocks traffic passing through it — IP routing/forwarding has been disabled. For data to pass from one NIC to the other, it must be analyzed by the application-level firewall and matched against one of the rules defined by the firewall administrator. Otherwise, the data is blocked and not passed on to the other NIC, which is connected to the Trusted network.

The advantages of a dual homed host firewall is that it is cost-effective and relatively robust. The disadvantage is that it provides a single point of failure. If the dual homed host fails, it prevents authorized people on the outside from connecting with the services provided on the inside and prevents any people on the inside from communicating with authorized servers on the Internet.

The Demilitarized Zone (DMZ)

We have said that there are many possible firewall designs, depending on the requirements being addressed. However, most mid-to-large-sized enterprises have common requirements which lead to firewall configurations that are generally similar to each other.

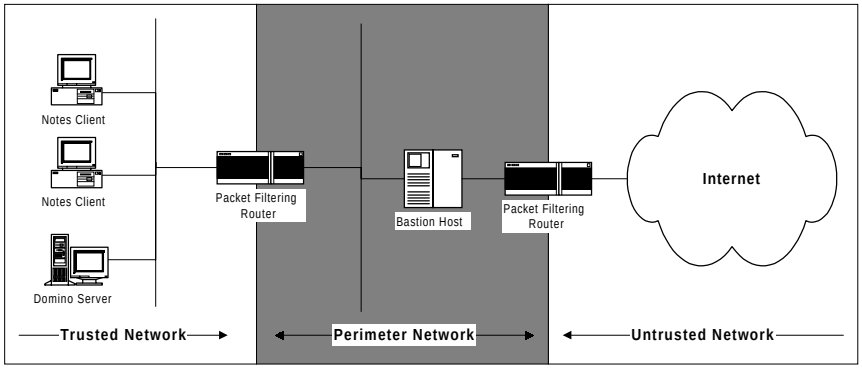
The common requirements are as follows:

- Provide secure access to Internet services for users within the enterprise. This is at least Web (HTTP) access and FTP, and any number of other services (NNTP, Telnet, etc.).
- Provide servers that can be accessed by external Internet users. Usually at least a Web server.
- Provide a gateway for Internet mail.

The recommended solution to these requirements is a demilitarized zone (DMZ) configuration.

Even though we touched briefly on the concept of a demilitarized zone as part of the Screened Subnet firewall architecture (see the figure in the section titled "Screened Subnet Firewall" earlier in this chapter), it nevertheless bears further scrutiny.

From the conceptual diagram in the figure below, we can see that the DMZ is a network that is neither inside the private network nor part of the Internet. IP filters are employed to screen this network. They are designed to allow inbound sessions to reach the services being offered, but still provide some protection.



Between the Internet and the private, trusted network lies a more solid boundary. This employs proxy servers and application gateways to separate the two networks, making the inside invisible from the outside, while nonetheless allowing local users access to the outside world.

For the sake of simplicity, the figure above only shows a configuration that contains a Bastion Host, with no proxies and no gateways. However, the DMZ can be further enhanced by building within it different networks and restricting access to specific services on specific networks within the DMZ.

The advantage of doing this is twofold: 1) it becomes difficult for an attacker to penetrate and take control of your whole DMZ; and, 2) by distributing firewall services within the DMZ, you also eliminate any single point of failure.

The disadvantage is that it adds another layer of complexity and increases the cost of your security architecture. For the sake of simplicity, in this redbook we will limit ourselves to a DMZ built on one network.

Notes and Domino Services

Notes and Domino can communicate using the TCP/IP protocol stack that comes standard with these communication tools.

Before describing how a Domino server can provide the needed services in a firewall configuration or how a Notes client can communicate to a server on the Internet or via the Internet to a secure server, we need to review the classes of application services provided and used, as well as how each of them communicates over TCP/IP. We will also discuss the types of proxies supported by Domino and the manner in which the information flows through them.

Standard Notes and Domino Services

There are three classes of services provided by Notes/Domino: Web application Services, Internet application services and Notes application services. These are broken down in the following manner:

- Web application services provide browser services, such as HTTP and Secure HTTP (HTTPS).
- Internet application services provide key messaging services such as POP, IMAP and SMTP; directory services such as LDAP; and newsgroup services such as NNTP.
- Notes application services use the Notes Remote Procedure Call (NRPC) which provides key Notes services such as Notes Mail, Discussion Databases, Replication, and Calendaring and Scheduling, as well as services to administer a Domino server via the Administrator Client.

The table below shows a breakdown of each service and the TCP port to which it maps, as well as the secure (using SSL) equivalent:

Service	*Port*	*SSL Port*
HTTP	80	443
POP	110	995 (new)
IMAP	143	993 (new)
SMTP	25	465 (new)
LDAP	389	636 (new)
NNTP	119	563 (new)
NRPC	1352	—-
IIOP	53148 (new)	53149 (new)

A common misconception concerns how a Notes client uses ports to access a specific service. A Domino server will use ports consistently. That is, it will handle HTTP requests via port 80 and will handle NRPC requests via port 1352. However, a Notes client will use the first available port in the unprivileged port area, which is port 1024 and above. So, a client might make an HTTP request on port 1035 and make an NRPC request on port 1041. If a client has problems connecting to a server behind a firewall and the problem seems to exist with the client, don't assume that the client is connecting via a specific TCP port.

Proxies Supported by Domino

Domino works with a number of proxies that aid in the implementation and configuration of a secure set of services in a firewall configuration. Domino works with the following:

- HTTP proxy (for both HTTP and HTTPS)
- HTTP Tunnel Proxy, using the HTTP Connect Method (for NRPC and Internet protocols)
- SOCKS v4.2 (for all protocols)
- Domino Passthru Proxy (for NRPC only)

The table below provides an overview of the Notes/Domino services provided by Proxy. Following that is a further explanation of the features of each proxy.

HTTP Proxy Features

The HTTP Proxy is an Application level proxy whose default port is 8080 (note that ports 8000 and 80 are also common). It provides HTTP services such as HTTP, HTTPS, FTP and Gopher.

	SOCKS	*HTTP*	*Passthru*
Notes Mail	☑	☒	☑
Discussion Databases	☑	☒	☑
Replication	☑	☒	☑
Calendar & Scheduling	☑	☒	☑
Directory	☑	☒	☑
Server Admin	☑	☒	☑
Web Navigator	☑	☑	☑
HTTP	☑	☒	☐
SSL	☑	☒	☐
POP	☑	☒	☐
IMAP	☑	☒	☐
LDAP	☑	☒	☐
SMTP	☑	☒	☐
NNTP	☑	☒	☐

☒ Requires the HTTP proxy and the HTTP Connect Method.

The HTTP proxy provides an efficient way to handle the HTTP protocol and provides good logging information (such as URLs) to aid in the administration of the firewall. It should be noted that the HTTP proxy does not support user authentication. It should be used in conjunction with a packet filter to allow access from the trusted network only.

HTTP Tunnel Proxy Features

The HTTP Tunnel Proxy is a Circuit level proxy which uses the HTTP proxy. It extends the HTTP proxy by enabling the HTTP Connect Method. Its default port is 8080 (note that ports 8000 and 80 are also common). It provides Internet protocol services such as POP, IMAP, SMTP, LDAP and NNTP. Finally, it also supports Notes application services, namely, NRPC.

In terms of security services (using the Secure Socket Layer, or SSL), you can configure the HTTP proxy server to allow an SSL client to open a secure tunnel through the proxy. When the secure tunnel is created, the HTTP proxy server does not read or interpret the data being passed between the user and the Internet server. It simply passes the information in a secure

way. SSL uses the Connect Method to extend the HTTP proxy to open a secure tunnel. You can use this method with SSL to communicate using the protocols listed in the previous paragraph.

SOCKS Proxy Features

The SOCKS Proxy is also a Circuit level proxy whose default port is 1080 (note that port 8080 is also common). It provides Internet protocol services such as POP, IMAP, SMTP, LDAP, and NNTP. Domino has built-in support for SOCKS 4.2. Most SOCKS Version 5 servers are backwards-compatible with SOCKS version 4.2 servers. Finally, it also supports Notes application services, namely, NRPC.

Domino Passthru NRPC Proxy Features

The Domino Passthru NRPC Proxy is an application level proxy whose default port is 1352. It provides Notes application services through NRPC. It can also be configured to be a multi-hop proxy, in the sense that it can act as an intermediary proxy and pass requests to another Domino Passthru NRPC proxy. This proxy was initially released as part of the Notes server in release 4.0.

Lotus recommends using a Domino Passthru server as an application proxy for Notes and Domino RPCs. A Passthru server provides all levels of Notes and Domino security while allowing clients who use dissimilar protocols to communicate through a single Domino server. It does not allow other Internet protocols, such as HTTP, IMAP and LDAP for example, to use a Domino Passthru server to communicate.

Real World Examples Using Notes and Domino

At this point, we can now explore the way to securely provide the following services to your users from a Notes/Domino perspective:

- How to securely place a Domino server behind a firewall.
- How to access and update the Domino server databases from within the private network.
- How to use Notes across the Internet (using regular Notes clients and Web browsers).
- How to exploit the non-IP communication capabilities of Notes to eliminate some firewall requirements.
- How to configure a Simple Mail Transfer Protocol (SMTP) relay host.

We now consider how to provide access to Notes databases from the Internet. Note that in these examples, the flow is from a secure environment to an untrusted environment. You can go the other way (from an untrusted environment to a trusted environment) by understanding the examples below and reversing the flow. However, do this with the utmost care and attention and do it within the confines of your security policy.

For a client on the Internet (either a Web browser or a Notes client), a Domino server is just another TCP/IP host. We can reasonably apply standard Internet practices for positioning and protecting it. We can do the same for a Notes client, since both rely on TCP/IP as their protocol of choice.

It is important to reiterate that in addition to adhering to your security policy, you should apply the rules that we discussed earlier in this chapter:

- A server that is accessible from the Internet should never be within a secure network. Instead it should either be directly connected to the Internet, or in a demilitarized zone (DMZ) between the Internet and a private network.
- The server is accessible from any Internet client. It is therefore open to attack and should be treated as an extension of the firewall. Such an exposed server could be thought of as being sacrificial.

That is to say, you will do your best to protect it and prevent an attacker from gaining access to it. However, there is always a risk that it will be compromised. Your firewall design should minimize the damage that can be done if this happens.

No matter what solution you use, you will face one common headache — maintaining the security of the server, while at the same time keeping it up-to-date. This is a problem even for servers that simply publish slow-changing information. It is even worse if the server is publishing data that changes in real time, or is receiving and processing input from client users.

The problem is often surmounted by opening up a number of wormholes, or connections through the internal firewall that support maintenance sessions such as FTP, Telnet, and database connections. However, from an attacker's point of view, each new wormhole represents a potential break-in point.

Domino gives us an excellent solution to this knotty problem. By using the replication capabilities of Notes, we can maintain the data published on and received via the Web on a server located safely in our secure network.

This book does not give a full description of Notes replication. Should you require a refresher, please refer to *Lotus Notes Network Design*, SR23-7378 for a more detailed treatment of this topic.

We still need to have a wormhole to pass data between the two servers, but the connection it provides can be authenticated and encrypted, making it very difficult for an attacker to exploit. It does not even need to be a TCP/IP connection. All administrator access to the Notes databases is also controlled using strong authentication methods, which further improves security.

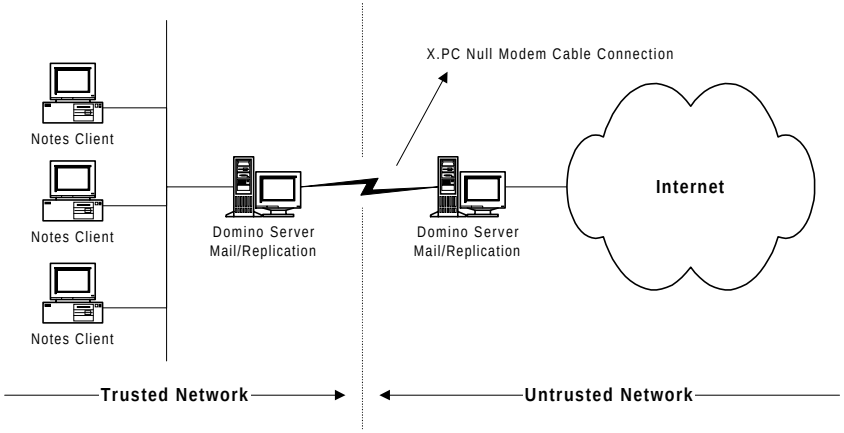
We further recommend setting up any server that is in a non-secure network as a separate Notes domain, so that if it does get compromised it reveals the minimum about the rest of the Notes environment.

With this knowledge and the tools available to us we can now build the necessary firewall infrastructure to fulfill our communications and security needs. This is what we will cover in the rest of this section.

NRPC Services: No Firewall, No Proxy

This configuration is by far the simplest. It does not require any firewall and it is a good step towards having a point of presence on the Internet in a short time frame.

The figure below shows the details of this simple configuration.



What makes this configuration secure is that the communication between the two servers is done via XPC, which is a dial-up serial line transport-optimized protocol. The Notes security model provides the basis for this configuration.

Since we don't communicate using a network card, there is no means for an attacker to take control of the server in the trusted network and use that as a launching point to attack other systems on the network.

The main advantages of this configuration is that it provides a slick, quick, and low-cost method of establishing a point of presence on the Internet.

There are, however, a number of disadvantages:

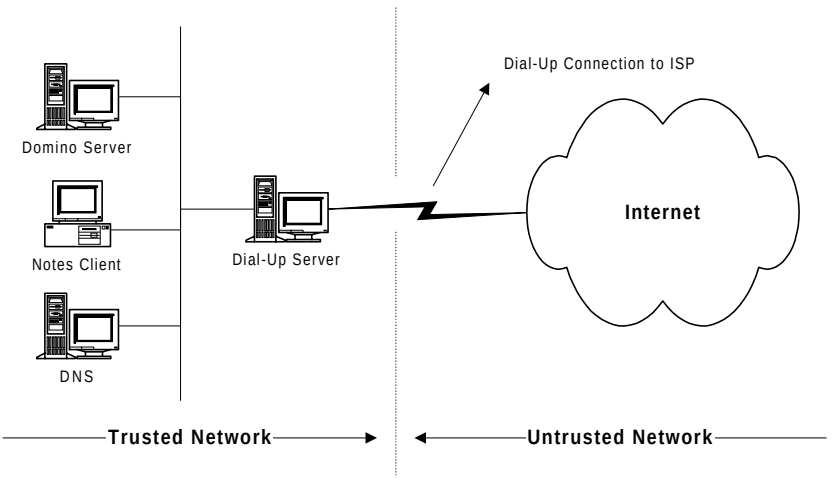
1. The server connected on the outside is basically a sacrificial host that can be prone to attacks. Therefore, it will require constant monitoring to ensure that the data contained on that server is not compromised by such an attack. Basic Notes services, such as the object container services and replication, should help somewhat in ensuring that the data has integrity, but it will not guarantee that the data has not been modified.
2. The connection between the servers is slow. 115,200 bps is the maximum you can hope to attain on a null modem cable between two COM ports. This is a fraction of the speed you can attain between LAN-connected computers (which can transfer data at speeds of 10 or 100 megabits per second).
3. The two machines must be physically close to one another, since the longest length of cable between the two machines is theoretically 64 feet. Therefore, you must secure the room in which both servers are located, in addition to the server room of your trusted network.

This solution should be considered a quick and expedient remedy to publishing information on the Internet. Unless both your means and security needs are modest, you should consider a more substantial solution.

Dial-up Internet Connection

The purpose of showing this configuration is not to suggest that you implement it (contrary to the rest of the configurations shown in this section), but rather that you stay away from it. Many companies are still setting up this kind of configuration, which exposes their whole trusted network to anyone on the Internet.

The figure below shows the details of this flawed configuration.



In this configuration, a Domino or Remote Access Service (RAS) server connects to the Internet Service Provider (ISP) and exchanges information with another server on the Internet. The problem here lies in the fact that the connection, while intended to be outbound to a specific host (or group of hosts) on the Internet, is really a two-way communication link.

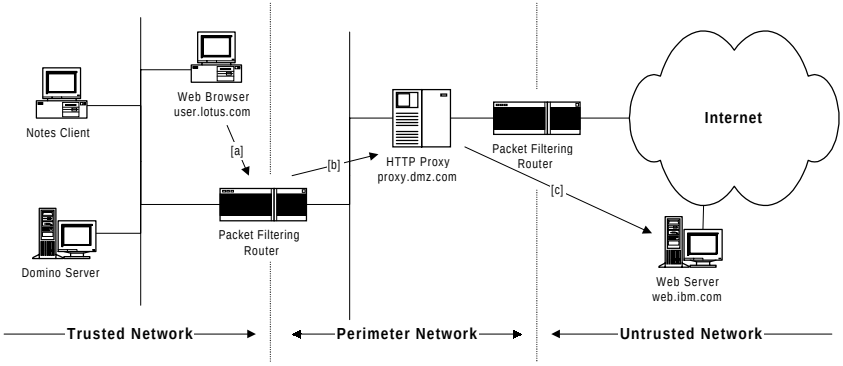
If an attacker can determine the time intervals when communication is established between your dial-up server and your ISP, this person can use the link to access other servers in your trusted network. Namely, the attacker can access your DNS and determine the configuration of your trusted network and also access your production Domino server, causing much damage.

If you have such a configuration in place, it would be wise to temporarily halt any kind of communications with your ISP. In a second phase, change this configuration for the one shown in the section titled "NRPC Services: No Firewall, No Proxy," where the server outside of the trusted network is the one dialing in to your ISP.

Browsing with Proxies and Firewall

This configuration shows how a user can browse to a Web server outside the firewall, going through an HTTP Proxy. This enables you to provide Internet access to your users, while keeping your trusted network completely secure. The browser can be a standard Web browser, as well as a Notes R5.0 client using its internal HTTP services.

The figure below shows the details of this configuration, as well as the details on the connection established between a Notes/Browser client and a Web server on the Internet.



The details of the connection are as follows:

[a] The client, user.lotus.com, wishes to communicate with the Web server web.ibm.com. The details of this transaction are as follows:

```
Src:          user.lotus.com, port: 1024+
Dest:         proxy.dmz.com, port: 8080
Target:       web.ibm.com, port: 80
```

[b] The connection goes through the inner packet filtering router and is passed on to the HTTP proxy, proxy.dmz.com, via port 8080.

[c] The HTTP proxy analyzes the request, changes Src and Dest and makes the HTTP request on behalf of the client:

```
Src:          proxy.dmz.com, port 1024+
Dest:         web.ibm.com, port: 80
```

The Web server web.ibm.com handles the HTTP request and passes the result back to the sender proxy.dmz.com, which then passes it back to the client, user.lotus.com.

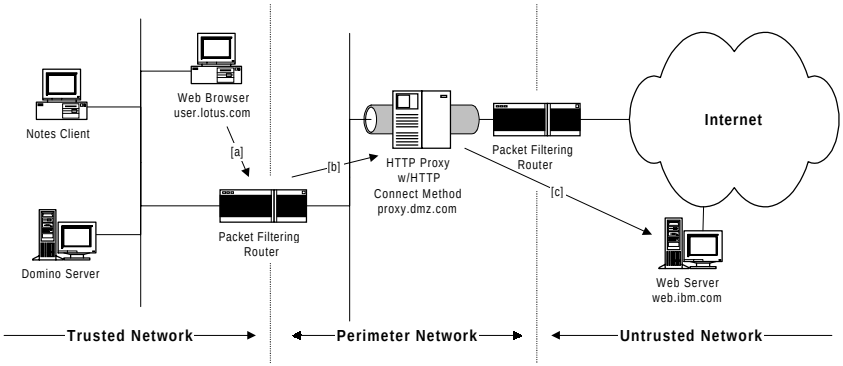
This demonstrates a connection without encryption. In the next example we will add encryption using SSL.

SSL Browsing with Proxies and Firewall

This configuration is not dramatically different from the previous configuration, except in some key areas. It shows how a user can securely browse to a Web Server outside the firewall, going through an HTTP Proxy and using SSL. This enables you to provide secure Internet access to your users, while keeping both your trusted network and the session between the client and the Web Server secure.

As with the previous section, the browser can be a standard Web browser, as well as a Notes R5.0 client using its built-in Web browser.

The figure below shows the details of this configuration, as well as the details on the secure connection established between a Notes/browser client and a Web server on the Internet.



The details of the connection are as follows:

[a] The client, user.lotus.com, wishes to communicate, using SSL, with the Web server, web.ibm.com. The details of this transaction are as follows (note that the port for the target system is no longer 80, but 443 because of SSL):

```
Src:          user.lotus.com, port: 1024+
Dest:         proxy.dmz.com, port: 8080
Target:       web.ibm.com, port: 443
```

[b] The connection goes through the inner packet filtering router and is passed on to the HTTP proxy, proxy.dmz.com, via port 8080.

[c] The HTTP proxy, using the HTTP Connect method, processes the request (but does not analyze the request, since it is encrypted using SSL), changes Src and Dest and makes the HTTP request on behalf of the client (again, note that the port for the destination system is no longer 80, but 443 because of SSL):

```
Src:          proxy.dmz.com, port 1024+
Dest:         web.ibm.com, port: 443
```

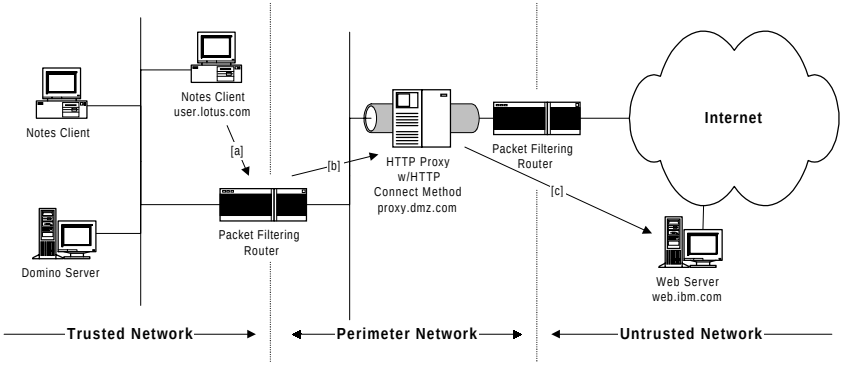
The Web server web.ibm.com handles the HTTP request and passes the result back to proxy.dmz.com, which then passes it back to the client user.lotus.com.

Notes Client Access Using the HTTP Tunnel Proxy

The two previous configurations showed how we could make HTTP requests, using unencrypted sessions and secured via SSL, through proxies and packet filtering routers. We will now see how we can use the same configuration to let the Notes client make both NRPC requests and requests using Internet protocols such as POP, IMAP, SMTP, LDAP, and NTTP.

The figure below shows the details of this configuration and the details on the connection established between a Notes client and a Web server on the

Internet via an HTTP Tunnel proxy.



The details of the connection are as follows:

[a] The client, user.lotus.com, wishes to communicate, using NRPC, with the Web server, web.ibm.com. The details of this transaction are as follows (note that the port for the target system is no longer 80, but 1352 because of NRPC):

```
Src:          user.lotus.com, port: 1024+
Dest:         proxy.dmz.com, port: 8080
Target:       web.ibm.com, port: 1352
```

[b] The connection goes through the inner packet filtering router and is passed on to the HTTP proxy, proxy.dmz.com, via port 8080.

[c] The HTTP proxy, using the HTTP Connect method, processes the request (but does not analyze the request, since it is using the HTTP Connection method), changes Src and Dest and makes the HTTP request on behalf of the client (again, note that the port for the destination system is no longer 80, but 1352 because of NRPC):

```
Src:          proxy.dmz.com, port: 1024+
Dest:         web.ibm.com, port: 1352
```

The Web server web.ibm.com handles the NRPC request and passes the result back to proxy.dmz.com, which then passes it back to the client user.lotus.com.

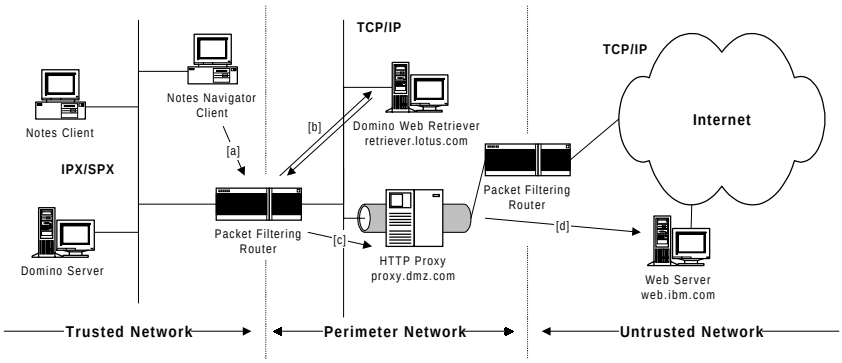
The connection is processed in the same way for Internet protocols, except that it substitutes port 1352 as the target port (and destination port once it is through the HTTP Tunnel Proxy) for the relevant port. These ports are detailed in the table located in the section, “Standard Notes and Domino services.”

Browsing Using the Web Retriever, Proxies, and Firewalls

Depending on your security requirements, permitting users to browse using HTTP might be more of a risk than your security policy allows. A way to still permit users to browse, but which is significantly more secure, is to use Web Retriever.

Web Retriever is a server task (called *Web*) which implements the HTTP protocol to retrieve Web pages and convert them into Notes documents. This task uses the Web Navigator database (WEB.NSF) which resides on the server and stores all pages that users retrieve from Web sites. There is a twofold advantage to using this feature of Notes and Domino; aside from the added security, storing Web pages in a central database reduces connection costs, since the page may already exist in the database for others to browse.

The figure below shows the details of this configuration and the details on the connection established between a Notes client and a Web server on the Internet, via the Web Retriever service provided by a Domino server located in the DMZ.



Note that the trusted network is not using TCP/IP, but IPX/SPX. This means that even if an attacker were able to get access to the DMZ and tried to launch an attack on the trusted network, that person would be thwarted. This is because any tactics or TCP/IP tools would get no further than the internal packet filtering router, since that router does not allow any IP routing between the DMZ and the trusted network. The Notes client and the Domino Web Retriever server communicate via NRPC, which is communication protocol independent (it will run as well on TCP/IP as on IPX/SPX).

The details of the connection are as follows:

[a] The Notes client wishes to communicate with the Web server, web.ibm.com, via the Domino Web Retriever server. An HTTP request (http://web.ibm.com) is generated using NRPC over IPX/SPX to the Domino Web Retriever server, retriever.lotus.com.

[b] The connection goes through the inner packet filtering router and is passed on to the Domino Web Retriever server (in this instance, we assume that the page was not cached in the Web Navigator database of the Domino Web Retriever server, otherwise the server would fetch the cached page in the database and return the information to the Notes Navigator client without any involvement from other firewall components).

[c] The Domino Web Retriever server processes the client request and creates an HTTP request via the HTTP proxy. The details of this transaction are as follows (note that the target port is 80 and not 1352, since this is issued as an HTTP request, converted from an NRPC request):

```
Src:          retriever.lotus.com, port: 1024+
Dest:         proxy.dmz.com, port: 8080
Target:       web.ibm.com, port: 80
```

[d] The HTTP proxy analyzes the request, changes Src and Dest and makes the HTTP request on behalf of the client:

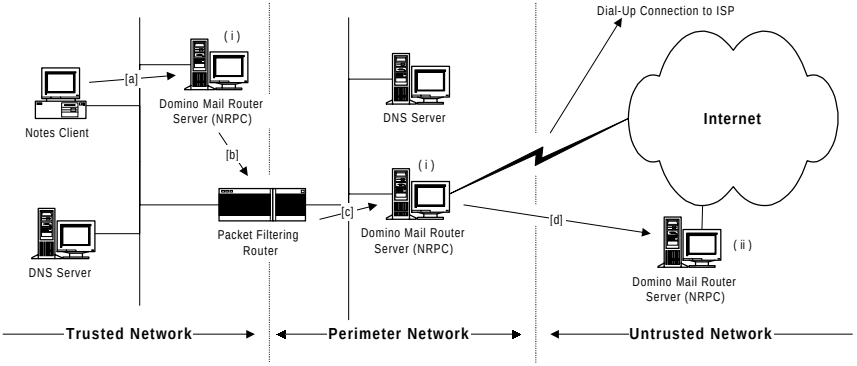
```
Src:          proxy.dmz.com, port 1024+
Dest:         web.ibm.com, port: 80
```

The Web server, web.ibm.com handles the HTTP request and passes the result back to proxy.dmz.com, which then analyzes the returned result set and passes it back to the server, retriever.lotus.com. The returned information is stored in the server copy of the Web Navigator database. The information is then returned via NRPC over SPX/IPX to the Notes Navigator client.

Mail Routing using Dial-Up NRPC

In this example, we demonstrate how it is possible to communicate to a Domino server located on the Internet for purposes of routing mail using NRPC over TCP/IP. Note that in this example, the connection to the Internet is done over an asynchronous dial-up line to the ISP.

The figure below shows the details of this configuration:



The details of the connection are as follows:

[a] The Notes client dispatches a mail message to the Domino Mail Router server located in the trusted network. The Notes client does this via NRPC (using TCP port 1024+). This server handles the message via TCP port 1352 and stores the mail message (ultimately destined for the Domino Mail Router Server located on the Internet) in its MAIL.BOX database.

[b] Using a Pull-Push Connection Document, when the "Route at Once if [ ] mail messages are pending" threshold has been exceeded or a scheduled routing event occurs based on that Connection Document, the server routes the mail messages through the packet filtering router to the Domino Mail Router server located in the Perimeter Network (DMZ). The Trusted Network Domino Router server does this via NRPC (using TCP port 1024+). If there are any messages waiting for the Trusted Network Domino Mail Router server, this server gets them from the Domino Mail Router server in the DMZ as part of the Pull-Push connection process.

[c] The Perimeter Network Domino Router server handles the message via TCP port 1352 and stores the mail message (ultimately destined for the Domino Mail Router Server located on the Internet) in its MAIL.BOX database.

[d] Using a Pull Push Connection Document, when the "Route at Once if [ ] mail messages are pending" threshold has been exceeded or a scheduled routing event occurs based on that Connection Document, the Domino Mail Router server in the DMZ dials up to the ISP and routes the mail messages to the Domino Mail Router server located on the Internet. The Trusted Network Domino Router server does this via NRPC (using TCP port 1024+). If there are any messages waiting for the DMZ Domino Mail Router server, this server gets them from the Domino Mail Router server in the DMZ as part of the Pull-Push connection process.

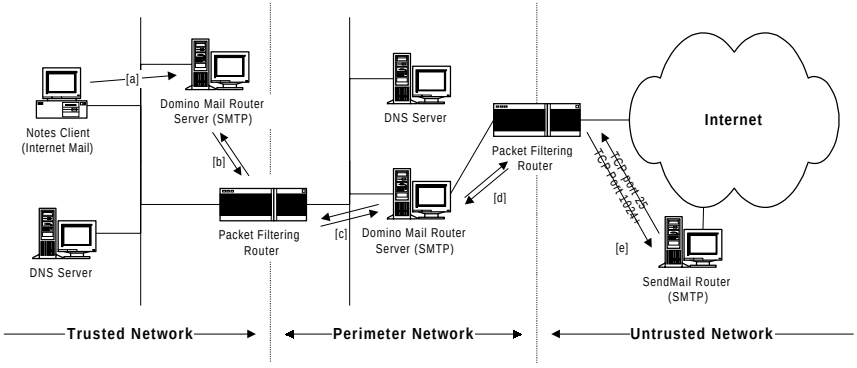
It is important to note that there are two types of Connection Documents involved, which are represented in the above figure:

- The Pull-Push Connection Document used by both the Trusted Network and DMZ Domino Mail Router servers
- The Push-Wait Connection Document used by the Internet Domino Mail Router server. The reason this server does not have a Pull-Push Connection Document is that this server cannot establish a connection with the DMZ Domino Mail Router server and must wait for the other server to contact it. So even if a fixed threshold of mail pending was defined and exceeded, it would not route mail to the DMZ Domino Mail Router server.

SMTP Mail Routing Using a Firewall

In this example, we demonstrate how it is possible to communicate to a Mail server located on the Internet to route SMTP mail. Note that in this example, the end server is not a Domino server, but a SendMail Router server, which interoperates only with the Domino Mail Router server using SMTP and no other messaging protocol. In the example, the connection to the Internet (via an ISP) is permanent and not dial-up, as in the previous example.

The figure below shows the details of this configuration:



The details of the connection are as follows:

[a] The Notes client dispatches an internal mail message to the Domino SMTP Mail Router server located in the trusted network. The Notes client does this via SMTP (using TCP port 1024+). Through the SMTP server task, the server handles the message via TCP port 25 and stores the mail message (ultimately destined for the SendMail Router Server located on the Internet) in its SMTP.BOX database.

[b] SMTP is a point-to-point mail protocol. When the Domino SMTP Mail Router server sends a message over SMTP, it does the following: 1. It checks the recipient's address, which is in the format localpart@domain, and looks up the domain in the Domain Name Service (DNS); 2. DNS returns the IP address of a server in the domain that accepts mail over SMTP; 3. The Trusted Network Domino SMTP Mail Router server connects to the destination server over TCP/IP, establishes an SMTP connection, transfers the message, and closes the connection. In this case, the server routes the mail messages through the packet filtering router to the Domino SMTP Relay server located in the Perimeter Network (DMZ). The Trusted Network Domino SMTP Router server does this via SMTP (using TCP port 1024+).

[c] If there are any messages waiting for the Trusted Network Domino SMTP Mail Router server, this server gets them from the Domino SMTP Relay server in the DMZ (using port 25). Through the SMTP server task, the server handles the message via TCP port 25 and stores the mail message (ultimately destined for the SendMail Router Server located on the Internet) in its SMTP.BOX database.

[d] Like the explanation of the SMTP routing process in [b], the DMZ Network Domino SMTP Relay server connects to the destination server over TCP/IP, establishes an SMTP connection, transfers the message, and closes the connection.

[e] In this case, the server routes the mail messages through the packet filtering router to the SendMail Router server located on the Internet. The DMZ Domino SMTP Relay server does this via SMTP (using TCP port 1024+). If there are any messages waiting for the DMZ Domino SMTP Relay server, this server gets them from the SendMail Router server on the Internet (using port 25).

It is important to note that there are two connections here, one static for SMTP, the other, a dynamic port to enable the client side. The ability to provide only one connection for outbound and inbound SMTP traffic is being considered for a post R5.0 release.

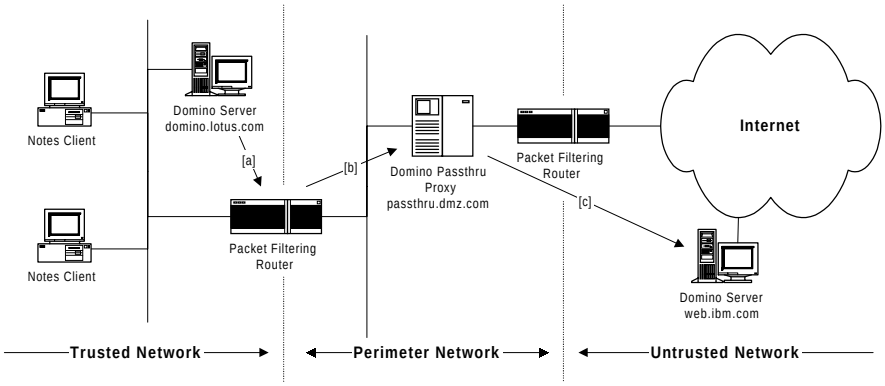
Note On a regular basis, the SMTP Relay server polls the SendMail Router on the Internet, so even without any outgoing mail from the trusted network, mail from the Internet is still coming in.

Domino Replication Using a Proxy and a Firewall

In this example, we demonstrate how it is possible to replicate databases over the Internet between a Domino server located in the Trusted Network and a Domino server on located on the Internet. The main reasons for doing this are the following:

1. The cost of leasing a dedicated line would be too expensive to enable connectivity from one server to the other.
2. The server belongs to another company and dial-up replication is either too expensive or not practical.
3. The company offers a dual mode of receiving the data, either by accessing through Web Browsers or by replicating the data locally.

Although the most common reasons are listed above, the list is not exhaustive. The figure on that follows page shows the details of a configuration that permits secure replication between Domino servers over the Internet.



The details of the connection are as follows:

[a] The Domino server, domino.lotus.com, wishes to replicate, using NRPC, with the Domino server, web.ibm.com. The details of this transaction are as follows (note that the port for the passthru proxy is 1352 and not 8080, since it is after all a Domino passthru server and not an HTTP proxy server):

```
Src:          domino.lotus.com, port: 1024+
Dest:         passthru.dmz.com, port: 1352
Target:       web.ibm.com, port: 1352
```

[b] The connection goes through the inner packet filtering router and is passed on to the Domino Passthru proxy, passthru.dmz.com, via port 1352.

[c] The Domino Passthru proxy analyzes the request, since it understands NRPC, changes Src and Dest, and makes the replication request on behalf of the Domino server located in the Trusted Network:

```
Src:          passthru.dmz.com, port 1024+
Dest:         web.ibm.com, port: 1352
```

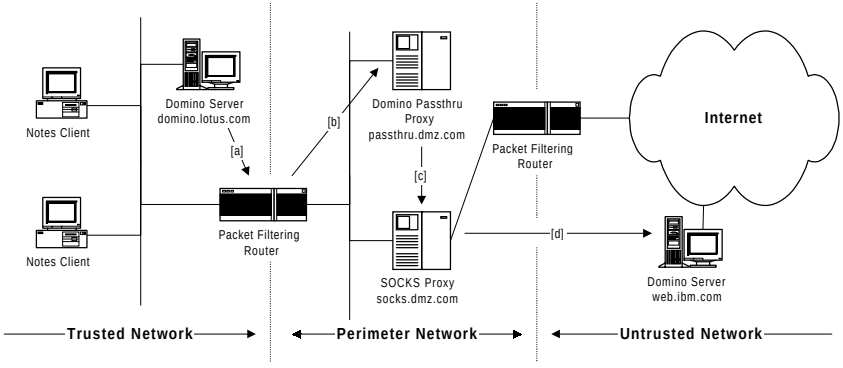
The Domino server, web.ibm.com, handles the NRPC request for replication and if the request is valid (given the provisos of Notes and Domino security, namely that proper authentication has been successfully completed and that the ACL levels in the web.ibm.com server's database are appropriate) the replication process begins. This is done via the Domino passthru server in the DMZ, passthru.dmz.com, and culminates back to domino.lotus.com.

Domino Replication Using Multiple Proxies

In this example, we build upon the previous example and add SOCKS proxy. This might seem like an odd configuration, but there are two things to know about it:

1. It is a more secure configuration, since the SOCKS proxy provides an added layer of security within the DMZ.
2. This is a common configuration for sites that have already standardized on a non-Domino proxy that they don't wish to change or modify.

This configuration is possible by placing a Domino passthru server and funneling the NRPC traffic through the SOCKS proxy. The figure below shows this type of configuration.



The details of the connection are as follows:

[a] The Domino server domino.lotus.com, wishes to replicate, using NRPC, with the Domino server web.ibm.com. The details of this transaction are:

```
Src:          domino.lotus.com, port: 1024+
Dest:         passthru.dmz.com, port: 1352
Target:       web.ibm.com, port: 1352
```

[b] The connection goes through the inner packet filtering router and is passed to the Domino Passthru, proxy passthru.dmz.com, via port 1352.

[c] The Domino Passthru proxy analyzes the request, since it understands NRPC, changes Src and Dest and makes the replication request on behalf of the Domino server located in the Trusted Network. It then funnels the request through the SOCKS proxy, socks.dmz.com located in the DMZ. The details of this transaction are as follows (note that the destination port is 1080 and not 1352 since we are going through a SOCKS proxy):

```
Src:          passthru.lotus.com, port: 1024+
Dest:         socks.dmz.com, port: 1080
Target:       web.ibm.com, port: 1352
```

[d] The SOCKS proxy passes on the request without analyzing the data, since it does not understand NRPC, changes Src and Dest and makes the replication request on behalf of the Domino server located in the Trusted Network:

```
Src:          socks.dmz.com, port 1024+
Dest:         web.ibm.com, port: 1352
```

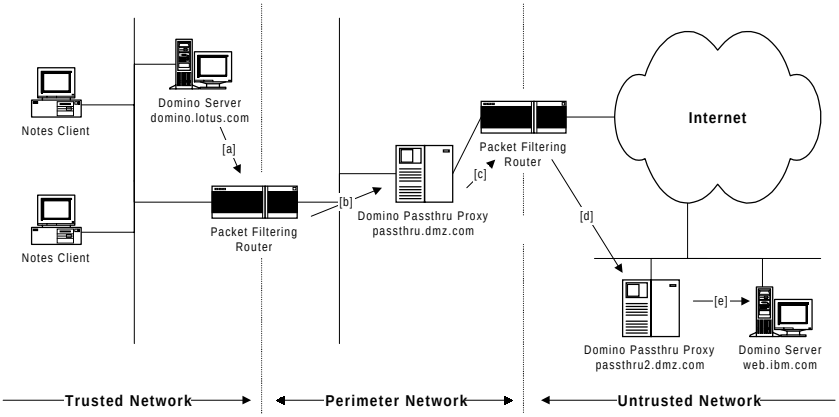
The Domino server web.ibm.com handles the NRPC request for replication and if the request is valid (given the provisos of Notes and Domino security, namely that proper authentication has been successfully completed and that the ACL levels in the web.ibm.com server database are appropriate) the replication process begins. This is done via both the SOCKS proxy, socks.dmz.com, and the Domino passthru server in the DMZ, passthru.dmz.com, and culminates back to domino.lotus.com.

The configuration shown below could have been accomplished without a SOCKS proxy. However, if this already exists, adding a Domino Passthru server only adds a layer of security. Again, the type of configuration present in your DMZ is entirely dictated by your security policy.

Multi-Hop Domino Replication with Proxies and Firewall

In this example, we build upon the example shown in the section "Domino Replication Using a Proxy and a Firewall." This might seem like an odd configuration, but you may be required to replicate through an intermediary Domino server because the IT specialists in the other company may have read this Redbook and have decided that they too wish to protect their Domino server behind a firewall.

The figure below shows this type of configuration.



The details of the connection are as follows:

[a] The Domino server domino.lotus.com wishes to replicate, using NRPC, with the Domino server, web.ibm.com. The details of this transaction are as follows (note that the port for the passthru proxy is 1352 and not 8080, since it is after all a Domino passthru server and not an HTTP proxy server):

```
Src:          domino.lotus.com, port: 1024+
Dest:         passthru.dmz.com, port: 1352
Target:       web.ibm.com, port: 1352
```

[b] The connection goes through the inner packet filtering router and is passed on to the Domino passthru proxy passthru.dmz.com via port 1352.

[c] The Domino passthru proxy passthru.dmz.com analyzes the request, since it understands NRPC, changes Src and Dest, and makes the replication request on behalf of the Domino server located in the Trusted Network:

```
Src:          passthru.dmz.com, port: 1024+
Dest:         passthru2.dmz.com, port: 1352
Target:       web.ibm.com, port: 1352
```

[d] The second Domino passthru proxy, proxy2.dmz.com, analyzes the request since it understands NRPC, changes Src and Dest, and makes the replication request on behalf of the Domino server located in the Trusted Network:

```
Src:          passthru2.dmz.com, port 1024+
Dest:         web.ibm.com, port: 1352
```

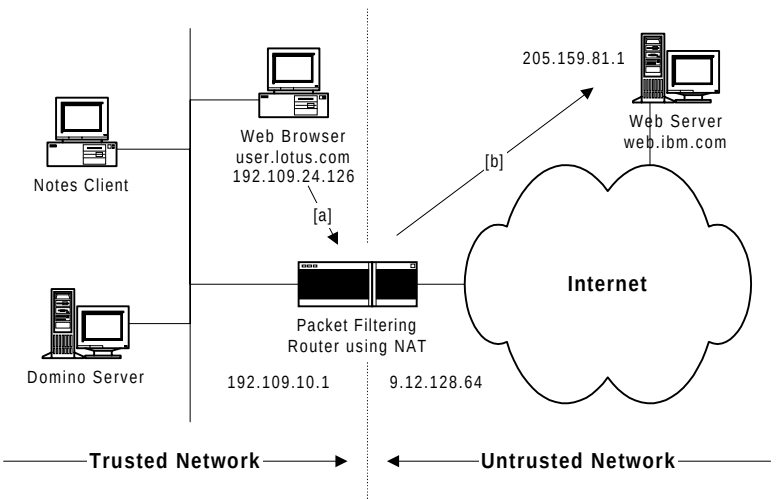
The Domino server web.ibm.com handles the NRPC request for replication and if the request is valid (given the provisos of Notes and Domino security, namely that proper authentication has been successfully completed and that the ACL levels in the web.ibm.com server database are appropriate) the replication process begins, which is done via the second Domino passthru proxy, passthru2.dmz.com to the Domino passthru proxy in the DMZ, passthru.dmz.com, and culminates back to domino.lotus.com.

Firewall Using Network Address Translation

A firewall using Network Address Translation (NAT) is similar to a SOCKS or Proxy configuration, in that the session is broken at the firewall boundary.

In the case of SOCKS, the client needs to be modified so that it can communicate with the SOCKS server and tell it which server to connect to. NAT is more like the proxy server, in that the client does not need to be modified. It sends packets to the real server address and treats that the firewall as just another router in the path.

However, the firewall alters the contents of the packets that arrive, changing the source address to an address that is outside the firewall boundary. The addresses it uses are drawn from a pool of legal IP addresses. The figure below shows this configuration.



In order to focus on the Network Address Translation aspect, the example above is deliberately simple. Since NAT is performed at the router, it is easy to expand this configuration into one involving a DMZ.

Basically, at the packet filtering router you create rules for routing packets from the secure network to the Internet and back. NAT takes care of the translation of the secure addresses. To configure NAT for our replication scenario we need to allocate the pool of addresses for NAT to use and then set up a mapping from the secure network address to this address pool.

The details of the connection are as follows:

[a] The client, user.lotus.com wishes to communicate with the Web server web.ibm.com. The details of this transaction are as follows:

```
Src:    user.lotus.com, ip: 192.109.24.126, port: 1024+
Target: web.ibm.com, ip: 205.159.81.1, port: 80
```

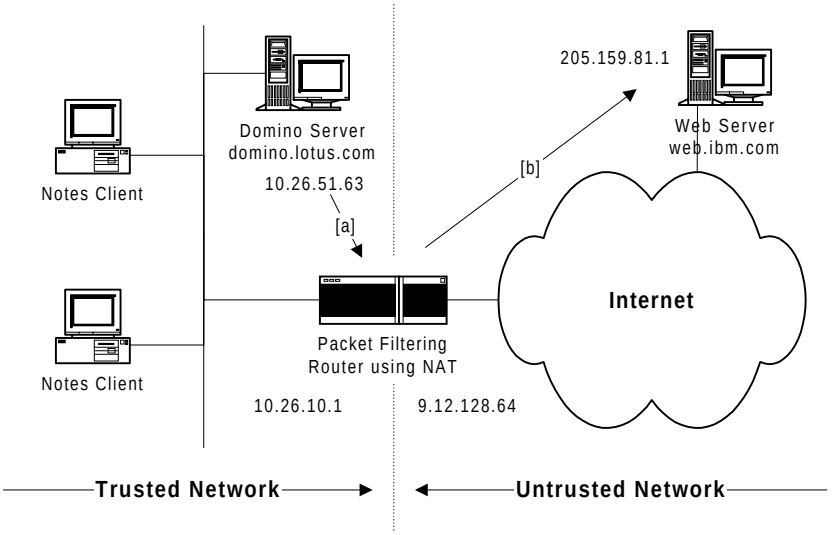
[b] The connection goes through the inner packet filtering router (at IP address 192.109.10.1), which uses NAT to do a simple one-for-one translation of the source address of the client from 192.109.24.126 to 9.12.128.64 (using the same subnet mask), and is sent with this IP address to the Web server web.ibm.com:

```
Src:    user.lotus.com, ip: 9.12.128.64, port: 1024+
Target: web.ibm.com, ip: 205.159.81.1, port: 80
```

The Web server web.ibm.com handles the HTTP request and passes the results of the request back to the client user.lotus.com via the packet filtering router using NAT, with the Web server web.ibm.com communicating using the translated address of the client, user.lotus.com.

Note This configuration shown is for a simple HTTP transaction. Since NAT deals with IP addresses and not ports, this could have been an NRPC transaction using ports 1352, which we will describe in the example configuration below.

Apart from the security advantage gained by breaking the session at the firewall boundary, NAT also allows non-legal IP addresses to be used in the secure network, such as IP address 10.*.*.*. The figure below shows this configuration.



The details of the connection are as follows:

[a] The Domino server domino.lotus.com, which uses a non-routable IP address based on subnet 10.*.*.*, wishes to replicate using NRPC, with the Domino server web.ibm.com. The details of this transaction are:

```
Src:    domino.lotus.com, ip: 10.26.51.63, port: 1024+
Target: web.ibm.com, ip: 205.159.81.1, port: 1352
```

[b] The connection goes through the inner packet filtering router (at IP address 10.26.10.1), which uses NAT to do a simple one-for-one translation of the source address of the client from 192.109.24.126 to 9.12.128.64 (using the same subnet mask), and is sent with this IP address to the Web server, web.ibm.com:

```
Src:    domino.lotus.com, ip: 9.12.128.64, port: 1024+
Target: web.ibm.com, ip: 205.159.81.1,  port: 1352
```

The Domino server web.ibm.com handles the NRPC request for replication and if the request is valid (given the provisos of Notes and Domino security, namely that proper authentication has been successfully completed and that the ACL levels in the web.ibm.com server database are appropriate) the replication process begins, which is done via the packet filtering router using NAT, with the Domino server web.ibm.com communicating using the translated address of the server domino.lotus.com.

It must be pointed out that the ICMP protocol is not supported by NAT, which means that if you attempt to ping from the Internet to a device on the secure network, this will not work, because ping uses the ICMP protocol and NAT will not translate the address. This is a good thing in

matters of security; however, administrators should be aware of this limitation when troubleshooting connections if the firewall uses NAT. Troubleshooting techniques, such as using PING, are explained in the section, "Troubleshooting Tools and Techniques."

Domino and Notes Proxy Configurations

In order for both the Domino server and the Notes clients to use a Proxy, they must be configured to do so, otherwise, the communication attempt through the firewall will fail. This sections shows how to configure both the server and the client.

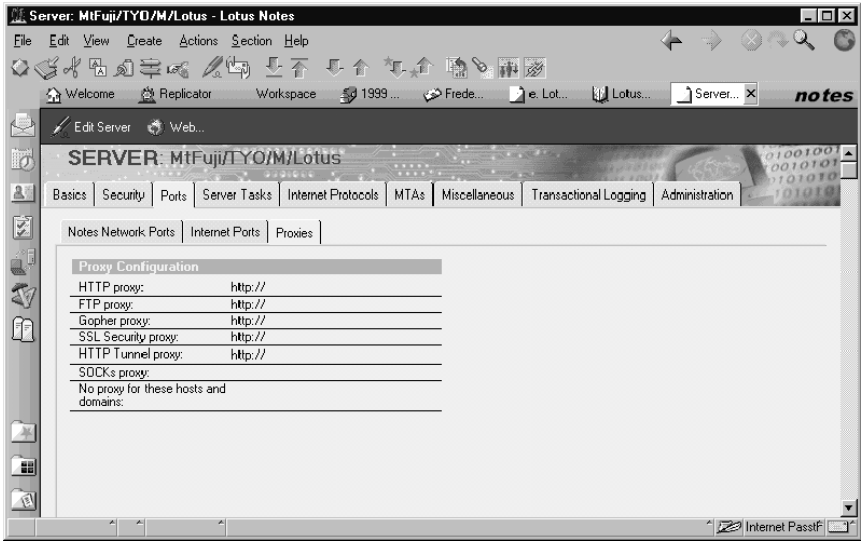
Domino Server Proxy Configuration

If you have a Web proxy server installed, you may want to use the same facility to route other traffic, such as NRPC connections, into the Internet. Since Release 4.5 the Domino server has been able to do this, using a technique called proxy tunneling. It relies on the proxy server supporting the HTTP Connect method, which we have touched upon in the real world configurations above.

The HTTP Connect method is a protocol designed by Netscape to allow non-Web applications to request a relay connection via the proxy server. The HTTP Connect method was originally developed to allow SSL connections to be proxied without losing end-to-end encryption, so you will often see it referred to as SSL tunneling.

To set up a Domino server to use a proxy tunnel for replication and mail delivery, do the following:

1. Open the Lotus Domino Directory (NAMES.NSF) on the server.
2. Select the Server document corresponding to the server you wish to configure.
3. Click on the "Edit Server" action button and a document like the one shown below will appear.
4. Click on the Basics Tab, then the Proxies Tab.
5. Identify type of Proxy running at your site (get this information from your firewall administrator, if you are not that person).
6. Specify the Proxy server TCP/IP host name or address.

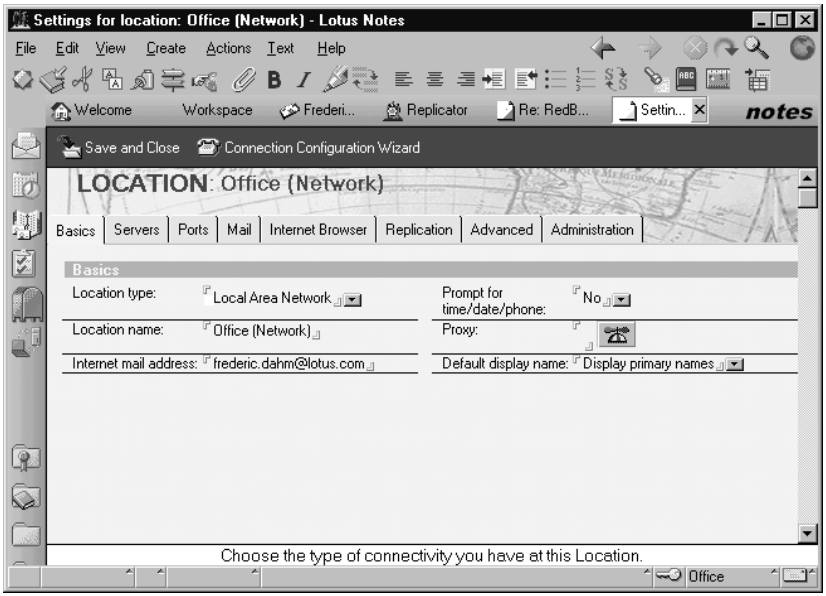


Note It is preferable to enter the TCP/IP host name of the Proxy, in case the IP address of the Proxy needs to be changed. If this happens and you entered an IP address instead of a host name, you will need to re-edit your Proxy Configuration. Also, enter the fully qualified domain name for the Proxy. In the dialog box below, we entered *proxy.dmz.com* as the fully qualified domain name of our HTTP Proxy name, as opposed to simply *proxy*. This will ensure that there will not be any confusion, should the administrator place another proxy (called *proxy*) in another domain to which you have access. An example would be two proxies bearing the same name in different domains: *proxy.lotus.com* and *proxy.ibm.com*.

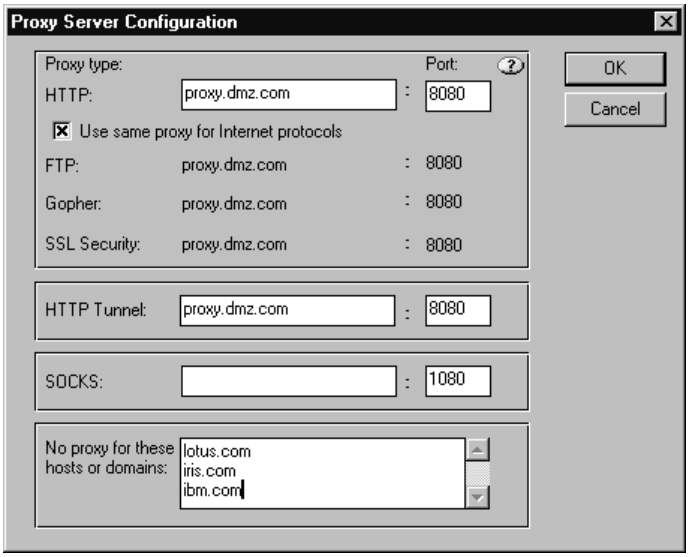
7. If access to specific trusted network hosts is desired, specify the local Domain name in the NoProxy Field. This effectively tells the server to go directly to that host instead of trying to negotiate through the specified Proxy.
8. Once the information is properly entered, save the Server document by pressing on the “Save and Close” action button.

Notes Client Proxy Configuration

To set up a Notes client to use a proxy tunnel for replication and mail delivery, as well as browsing using the native browser in the client, do the following (the procedure goes through the Personal Address book to access the relevant location document; this can also be accomplished by clicking on the location section of the Notes status bar and selecting “Edit Current...”):



1. Open the Personal Address Book.
2. Select from the menu: View - Advanced - Locations.
3. Select the appropriate location by clicking on it.
4. Click on the “Edit Location” action button and a document like the one shown above will appear.
5. Click on the propeller hat button next to the Proxy field to bring up the Proxy dialog box as shown below.
6. Identify the type of Proxy running at your site (get this information from your firewall administrator).
7. Specify the Proxy server TCP/IP host name or address and port number (this enables the proxy).



Note It is preferable to enter the TCP/IP host name of the Proxy, in case the IP address of the Proxy needs to be changed. If this happens and you entered an IP address instead of a host name, you will need to re-edit your Proxy Configuration. Also, enter the fully qualified domain name for the Proxy. In the dialog box below, we entered *proxy.dmz.com* as the fully qualified domain name of our HTTP Proxy name, as opposed to simply *proxy*. This will ensure that there will no be any confusion, should the administrator place another proxy (called *proxy*) in another domain to which you have access. An example would be two proxies bearing the same name in different domains: *proxy.lotus.com* and *proxy.ibm.com*.

8. If access to specific trusted network hosts is desired, specify the local Domain name in the NoProxy Field. This effectively tells your client to go directly to that host instead of trying to negotiate through the specified Proxy.
9. Once the information is properly entered, click on OK.
10. Save the Location document by pressing on the “Save and Close” action button.

Caution In both the Server document and the Proxy dialog box, the entry in the SOCKS field of the dialog box overrides the HTTP/HTTP Tunnel entries.

Troubleshooting Tools and Techniques

Firewalls, especially those that use a combination of proxies and packet filtering routers, can be difficult to monitor, and when things go wrong and communications simply don't work between authorized machines on both sides of the firewall, it can be very difficult to diagnose and resolve.

This section examines the tools and techniques that exist to help you troubleshoot problems and resolve any issues you might have revolving around you firewall.

TCP/IP Tools

TCP/IP offers you a number of tools to help in case of communications problems. While you may be familiar with most of them, we will review them quickly here.

There are three tools worth mentioning in this family:

- Ping

  Ping uses the ICMP Echo and Echo Reply messages to determine whether a host is reachable.

- Traceroute

  Traceroute sends IP datagrams with low TTL values so that they expire en route to a destination. It uses the resulting ICMP Time Exceeded messages to determine where in the Internet the datagrams expired, and pieces together a view of the route to a host.

- Nslookup

  nslookup allows you to locate information about network nodes, examine the contents of a name server database, and establish the accessibility of name servers.

These applications are discussed in the following sections. The description for Ping and Traceroute are from the IBM Redbook, *TCP/IP Tutorial and Technical Overview*, GG24-3376-05 and the description for nslookup is from RFC 1739 - A Primer On Internet and TCP/IP Tools.

Ping

Ping is the simplest of all TCP/IP applications. It sends one or more IP datagrams to a specified destination host requesting a reply and measures the round trip time. The word ping, which is used as a noun and a verb, is taken from the sonar operation to locate an underwater object. It is also an abbreviation for Packet Internet Groper.

Traditionally, if you could ping a host, other applications such as Telnet or FTP could reach that host. With the advent of security measures on the Internet, particularly firewalls, which control access to networks by application protocol and/or port number, this is no longer strictly true. Nonetheless, the first test of reachability for a host is still to attempt to ping it. The syntax that is used in different implementations of ping varies from platform to platform.

Ping uses the ICMP Echo and Echo Reply messages. Since ICMP is required in every TCP/IP implementation, hosts do not require a separate server to respond to pings.

Ping is useful for verifying a TCP/IP installation. Consider the following four forms of the command; each requires the operation of an additional part of the TCP/IP installation:

Command	*Result*
ping loopback	Verifies the operation of the base TCP/IP software.
ping my-IP-address	Verifies whether the physical network device can be addressed.
ping a-remote-IP-address	Verifies whether the network can be accessed.
ping a-remote-host-name	Verifies the operation of the name server (or the flat namespace resolver, depending on the installation).

Traceroute

The Traceroute program can be useful when used for debugging purposes. It enables determination of the route that IP datagrams follow from host to host.

Traceroute is based upon ICMP and UDP. It sends an IP datagram with a TTL of 1 to the destination host. The first router to see the datagram will decrement the TTL to 0 and return an ICMP Time Exceeded message, as well as discarding the datagram. In this way, the first router in the path is identified.

This process can be repeated with successively larger TTL values in order to identify the series of routers in the path to the destination host. Traceroute actually sends UDP datagrams to the destination host. These reference a port number that is outside the normally used range. This enables Traceroute to determine when the destination host has been reached, that is, when an ICMP Port Unreachable message is received.

nslookup

nslookup is an interactive program to query Internet domain name servers. The user can contact servers to request information about a specific host or print a list of hosts in the domain. It comes with many TCP/IP software packages.

A user or administrator can use nslookup to examine entries in the Domain Name System (DNS) database that pertain to a particular host or domain; one common use is to determine the IP address of a host system from its name or the host's name from its IP address. The general form of the command to make a single query is: nslookup [IP_address | host_name].

If the program is started without any parameters, the user will be prompted for input; the user can enter either an IP address or host name at that time, and the program will respond with the name and address of the default name server, the name the server actually used to resolve each request, and the IP address and host name that was queried. "Exit" is used to quit the nslookup application.

Three simple queries are shown in the example below:

```
** SMCVAX$ NSLOOKUP

  Default Server: LOCALHOST
  Address: 127.0.0.1

     ** > PAUL.UQAM.EDU
        Server:  LOCALHOST
        Address:  127.0.0.1

        Name:    paul.info.uqam.edu
        Address:  198.109.12.79
        Aliases:  paul.uqam.edu

     ** > PAUL.INFO.UQAM.EDU
        Server:  LOCALHOST
        Address:  127.0.0.1

  Non-authoritative answer:

        Name:    paul.info.uqam.edu
        Address:  198.109.12.79

     ** > 160.110.21.51
        Server:  LOCALHOST
        Address:  127.0.0.1

        Name:    wizard.OZ.AU
        Address:  160.110.21.51

     ** > set type=MX
     ** > UQAM.EDU
        Server:  LOCALHOST
        Address:  127.0.0.1

  uqam.edu preference = 10, mail exchanger = passerelle.uqam.edu

        passerelle.uqam.edu   internet address = 198.109.12.9

     ** > EXIT
```

1. Requests the address of the host named “paul.uqam.edu”, a system at the Université du Québec à Montréal (UQAM). As it turns out, this is not the true name of the host, but a shortened version of the name that is accepted as an alias by the network. The full name of the host and the IP address are listed by nslookup.
2. Requests the address of host “paul.info.uqam.edu”, which is the same host as in the first query. Note that nslookup provides a “non-authoritative” answer. Since nslookup just queried this same address, the information is still in its cache memory. Rather than send additional messages to the name server, the answer is one that it remembers from before; the server didn’t look up the information again, however, so it is not guaranteed to still be accurate (because the information might have changed within the last few milliseconds!).
3. Requests the name of the host with the given IP address. The result points to the Internet gateway to Australia, “wizard.oz.au”. One additional query is shown in the dialogue below. nslookup examines information that is stored by the DNS. The default nslookup queries examine basic address records (called “A records”) to reconcile the host name and IP address, although other information is also available. In the final query below, for example, the user wants to know where electronic mail addressed to the “uqam.edu” domain actually gets delivered, since “uqam.edu” is not the name of an actual host. This is accomplished by changing the query type to look for mail exchange (MX) records by issuing a “set type” command (which must be in lower case). The query shows that mail addressed to “uqam.edu” is handled though a mail server called “passerelle.uqam.edu”.

Notes and Domino Tools

There are a couple of troubleshooting tools that are available to you to specifically troubleshoot Notes and Domino communication problems: NotesConnect (a command-line executable) and the TraceConnection dialog box in the Notes R5 client.

NotesConnect/NPING

NotesConnect helps you determine whether your TCP/IP connection problems are Domino/Notes-related or the result of a network problem. Iris developer Bob Lomme developed NotesConnect in response to this need to isolate TCP/IP connections from Domino and Notes. NotesConnect is a Notes application that uses Notes API calls to establish TCP/IP connections to any service, without using the Public Address Book or Notes address resolution logic.

NotesConnect is very similar to using a Ping utility to test connections. While Ping uses Internet Control Message Protocol (ICMP) and IP packets to determine if a given host is available, NotesConnect uses TCP/IP to determine if a given host has the desired service available on a specified port. This connection does not involve any exchange of service-specific protocols or application data. Therefore, NotesConnect more closely mimics connections to Domino services than the Ping utility.

NotesCONNECT can help you determine if:

- The target host name is defined in your Host file or within the Domain Name Service (DNS). If so, NotesCONNECT displays the mapping from the host name to the IP address.
- A service is running on a machine (the service might not currently be running, may have never been configured on the machine, or might be running on a different port).
- TCP is configured correctly — on the local machine running NotesCONNECT, the remote machine running a Domino server, or on the network (that is, routers, firewalls, gateways).
- There is a TCP infrastructure problem. For example, you can use NotesCONNECT to test firewalls, both from the intranet to the Internet, and vice versa. NotesCONNECT goes through firewalls exactly as Notes does (even if low-level pings are not allowed). Therefore, you can check if port 1352 is let through the firewall. If you have already opened port 1352 for TCP/IP on the firewall and nping.exe does not allow you to connect, then you need to work on the firewall configuration.

Note NotesConnect is the actual name of the application. The program file, called NPING.EXE can be found on www.notes.net at the following URL:

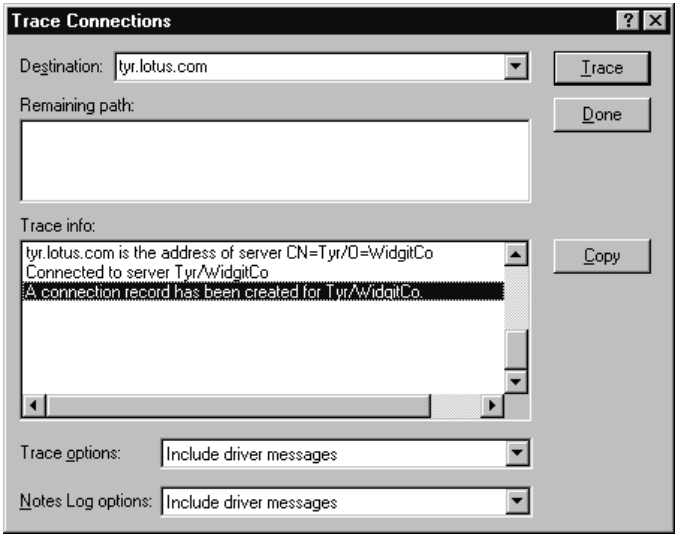
http://www.notes.net/pubdown.nsf/788925f20e7931c7852563e6006a0707/8263ac9a8e83578c852564960054e36d?OpenDocument

TraceConnection dialog box

To test connections from within Domino and Notes — that is, to test connections for any supported protocol using the Public Address Book or Notes address resolution logic — you can use the Trace Connection feature.

To do this, choose File - Preferences - User Preferences, click the Ports icon and click Trace Connection. Select the target server, and click Trace. The Trace InfoBox shows the steps Domino/Notes takes to make the server connection, and full trace information is recorded in the log file.

The figure below shows a screen shot of the Trace Connections dialog box:



Make sure that among the trace options you select "include driver messages". This will provide you with a verbose description of the trace and enable to better diagnose the problem.

Protocol and Network Analyzers

A protocol analyzer is an excellent tool to help analyze protocols used by the network of your computer system and your firewall configuration. They go well beyond the previously mentioned tools, but they can cost quite a lot of money and require up to a week of training to master the intricacies of their operation and to be able to decipher the logs for protocol you are analyzing.

Basic protocol analyzer models passively monitor individual network links, usually testing from the bottom of the protocol stack upward. Many can decode traffic, measure bit error rates, and provide historical data from switches, routers, and other devices.

Recreating and solving protocol problems in a firewall configuration, especially one that uses not only packet filters but proxies and Domino Passthru servers, might require you to acquire and use an advanced protocol analyzer with extensive tracing and analyzing capabilities including even perhaps expert systems built in to help you sort everything out.

Some packet analyzers allow users to test both sides of a network device, emulating traffic going in (including call setup and service negotiation) and monitoring traffic coming out. Sophisticated decoders, triggers, filters, and synchronized time stamps aid understanding of events occurring on different sides of the device being monitored, something you would need testing packet filters, for example.

Combining a protocol analyzer with the server log file provides you with the means (although it still might be time- and resource-intensive on your part) to determine accurately the nature of the problem and the means to resolve it.

There are many manufacturers of protocol analyzers. However, it is not the place of this book to recommend one vendor or another, nor to endorse any specific brand or model of protocol analyzer.

Some Firewall Best Practices

This chapter would not be complete without some best practice recommendations to help you avoid some of the problems experienced by other specialists that have implemented a firewall.

DHCP Issues

As stated previously, DHCP, the Dynamic Host Configuration Protocol, is a great protocol, part of the TCP/IP family which lets you set up a table linking MAC addresses to IP addresses. DHCP simplifies this by allowing the network manager to specify a range of available IP addresses without having to tie each one to a specific MAC address.

However, it is this dynamicity of addresses that can cause problems. These problems can occur in a couple of ways: when assigning dynamic addresses to clients, or worse, by assigning dynamic addresses to servers.

Dynamic Address for Clients

When designing your firewall, you may have set rules in your router to filter packet to allow only specific IP address ranges, or even specific IP addresses for clients.

If this is the case and you have a short lease on the IP addresses provided by your DHCP server, the IP addresses might change for these clients and suddenly, if they fall outside of the permitted address range, they will get their access to the Internet denied.

You should therefore ensure that the lease provided by the DHCP server to clients is long enough or that if these addresses were to change, that they still would fit within the allowed permitted range.

Dynamic Address for Servers

Since servers are few compared to client workstations, the general rule of thumb is to specifically list each server IP address as a permitted IP address in the rule table for the pack filtering router.

However, some administrators don't assign static addresses to these servers, they provide the server IP addresses through DHCP. If the IP address changes, then two things may occur:

- There is a mismatch between the actual IP address and that listed in the DNS database, since DHCP and DNS do not complement each other. Thus, name lookups will fail and the server will not be reachable by clients, even though it is operational and on-line.
- The new address does not match a rule in the packet filtering router log and any connection through the firewall to or from that device will fail, with a corresponding loss in service, which can be minor or near-fatal to your company depending on the client-facing nature of the machine and the information it contains.

You should therefore assign static addresses to servers that will not change or make your packet filtering rules a little bit more lax, if your security policy permits it.

Fully Qualified Domain Names

A lot of problems occur when only the host name is used. For example, if you wish to access the server superman, you might not get to the intended server if you have access to multiple domains and each contains a superman server, such as superman.iris.com, superman.lotus.com and superman.ibm.com.

Caution If you specify only the host name as part of a rule set, the name can be spoofed coming from a different domain and create a hole in your firewall security.

You should therefore specify always the fully qualified domain name (for example, superman.lotus.com), which includes the host name *superman* with the domain name *lotus.com*. This will eliminate any possible naming ambiguity.

IP Addresses in the Domino Directory

Some administrators believe that they should not leave anything to chance and place the IP address of a server in the Domino Directory in documents such as:

- Connection Records
- Server Records
- Location Records
- Account Records
- Proxy dialogs

The problem with this is that, if the IP address of the server changes, you need to revisit every single document to reflect that change. If you place the fully qualified domain name, you save time, since you only need to update the proper record in the DNS Database.

You should therefore avoid the practice of placing IP addresses in these Domino Directory documents.

Encrypt Whenever Possible

Review your security policy and if you feel that your firewall does not afford you enough protection, consider using broad encryption services.

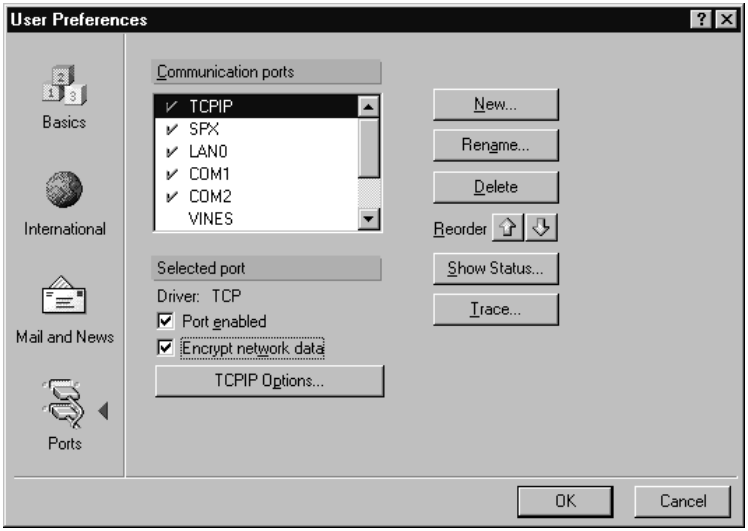
Encrypt Server Databases

The best place to start is to encrypt the databases that are located on servers within the DMZ or connected to the Internet. If the security of either area is compromised by an attacker, this added security will make it difficult for the attacker to reap any rewards from his/her actions.

Enable Encryption for NRPC Services

It is possible to easily encrypt ports for NRPC services by selecting File - Preferences - Notes Preferences... And then by selecting Ports. Choose the TCPIP port and select the Encrypt Network Data check box as shown below. Then click on OK.

The figure below shows the user's port preferences dialog box.



All traffic going through the TCP/IP port will be encrypted.

Use SSL for Internet Protocols

When using Internet protocols such as POP, IMAP, SMTP, LDAP or NNTP, enable SSL to encrypt the information in transit. If you use proxies, make sure that they are configured to use the HTTP Connect Method to be able to pass SSL traffic and not reject it.

Use S/MIME for Mail

Using the S/MIME client in Notes R5.0, encrypt your mail going outside of the Trusted Network so as to prevent it from being intercepted and read and/or modified in transit.

Have a Stringent Security Policy

It is best to be as stringent as possible at the beginning when defining your security policy and then, where appropriate, to be a little bit more lax with some rules. By being stringent, all the access points to your trusted network are known and it is easy to control them. It is far better to proceed in this way than to relax at the onset and then realize that many holes have to be filled and many back doors need to be closed.

Security Is...

Network security as it applies to firewalls is simple: It is a secure *application*, operating on a secure *host*, operating on a secure *network*. The Domino server *and* the network *and* the Administrator(s) must work together; otherwise, true security is impossible.

Summary

In this chapter we have explained the basic functions of a firewall and what protection it can offer. We have covered different types of firewalls and the concept of a demilitarized zone (DMZ) before looking at some real world examples using Notes and Domino through a firewall. Finally, we have discussed troubleshooting tools and techniques as well as firewall best practices.

Chapter 6
Directories and Single Sign On Revealed

The purpose of this chapter is to provide you with a comprehensive overview of Directories and Single Sign On (SSO). We will tell you what our definition is and how you can leverage it using Domino R5.0.

We will give you an introduction to directories, and cover some of the new features and functionalities of the Domino Directory. We will show you how to use the Domino Directory with LDAP (Lightweight Directory Access Protocol) clients.

An extended section will cover the integration of the Microsoft IIS server as the HTTP server for Domino, as well as some operating system specific Single Sign On features for Windows NT and OS/400.

Some of the topics presented in the directory section are general in nature. They cover problems related to multiple directories. So you can get some ideas for your directory strategy, as well as what steps are needed to use multiple-directory-based systems and applications at the same time without the need for authentication to all systems.

Directories and Single Sign On — What's It All About?

In distributed computing environments, users need to log on to many different systems and applications. Each system requires the user to remember and enter a different user or application ID and password.

Single Sign On gives users the ability to authenticate themselves — that is, to prove their identity — to a network one time, and thereafter to have access to all authorized network resources and applications without the need for any additional authentication. It requires a common directory store to get the required information from a single source.

Users and companies who implement Single Sign On can expect the following benefits:

- Simpler administration.

  The primary barrier to adoption for most Single Sign On implementations is that they are bound to the operating system. This adds to the administrator's burden by requiring specific tasks to be performed.

- Better administrative control.

  All network management information should be stored in a single repository, the administration directory. This means that there is a single, authoritative listing of each user's rights and privileges. This allows the administrator to change a user's privileges and be sure that the results will propagate network-wide.

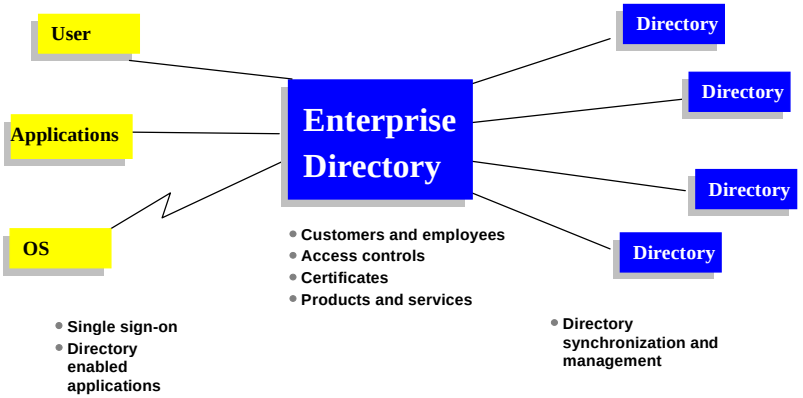
- Improved user productivity.

  Users are no longer required to have multiple logons, nor are they required to remember multiple passwords in order to access resources. This is also a benefit to support personnel, who have fewer requests for forgotten passwords. One approach to get around the issue of forgetting passwords is to use a master password to unlock all the other passwords. Some Single Sign On solutions on the market work that way.

- Better network security.

  All Single Sign On methods provide secure authentication and provide a basis for encrypting the user's session with the network resource. Eliminating multiple passwords also reduces the risk of potential security breaches. Another possibility is the use of Smartcards to securely store user certificates and authentication tokens.

The ongoing challenge is that every operating system and application has its unique system to maintain security information in a proprietary directory. This requires a large company to have multiple servers, applications, platforms, and clients to maintain and administer.



The ultimate goal would be to use an enterprise or meta directory to store all user credentials, together with group information, system specific configuration data, and access control lists.

In the next section, we will explain what the issues are if you opt for a single, common directory, what you need to know, and what the limitations are.

Directories — Technical Background

What Is a Directory?

A directory is a listing of objects, arranged in some logical order with details about each object. Common examples are a city telephone directory and a library catalog. For a telephone directory, the objects listed are people. The names are arranged alphabetically, and the details are addresses and telephone numbers. Books in a library catalog are ordered by author or by title, with further information provided about the book such as the ISBN number.

Object	*Value*
Name	Lotus Administration Manual
Version	4.6a
Language	English
ISBN	123-456-7890

In computer terms, a directory is a specialized database, also called a data repository, that stores ordered information about objects according to type. One directory object could be a list of information about printers (the objects) consisting of typed information such as location (a formatted character string), speed in pages per minute (numeric), print streams supported (for example PostScript or ASCII), and so on.

Directories allow users or applications to find resources that have the characteristics needed for a particular task. For example, a directory of users can be used to look up a person's e-mail address or department. A directory can be searched to find all people working in a specific department. Or a directory of application servers can be searched to find a server that has access to an individual customer's personal information.

The terms white pages and yellow pages are usually used to describe different functional uses of a directory. A "white pages" search takes the name of an object (e.g., person, group) to retrieve other characteristics of that object (e.g., phone number, members). Similarly a "yellow pages" search is made when the name of a particular object is not known, and so the directory is searched by defining attributes other than the a name (e.g., searching for a restaurant category rather than name). However, directories stored on a computer are much more flexible than the yellow pages of a telephone directory because they can usually be searched by a variety and a combination of specific criteria, not just by one predefined set of categories.

Differences Between Directories and Databases

All directories are based on a database, and so a general purpose directory sets it apart from a general purpose relational database. One such characteristic is that directories are accessed (read and searched) much more often than they are updated (written). Hundreds of people might look up an individual's phone number, or thousands of clients might look up the characteristics of a particular group, but the phone number or group characteristics rarely change.

Because directories must have the potential to support high volumes of read requests, they are typically optimized for read access. Write access might be limited to system administrators or the owner of specific pieces of information. A general purpose database, on the other hand, needs to support applications such as airline reservations and banking requiring high update volumes.

Because directories are meant to store relatively static information and are optimized for that purpose, they may not be appropriate for storing information that changes rapidly.

Another important difference between directories and general purpose databases is that directories do not always support transactions. Transactions are all-or-nothing operations that must be completed in total or not at all. For example, when transferring money from one bank account to another, the money must be debited from one account and credited to the other account in a single transaction. If only half of the transaction completes or someone accesses the accounts while the money is in transit, the accounts will not balance. General-purpose databases usually support such transactions, which complicates their implementation. Some vendor's directory implementations also support transactions; this is usually dependent on the underlying database.

Because directories deal mostly with read requests, the complexities of transactions can be avoided. If two people exchange offices, both of their directory entries need to be updated with new phone numbers, office locations, and so on. If one directory entry is updated, and then the other directory is also updated, there is a brief period during which the directory will show that two people have the same phone number. Because updates are relatively rare, such anomalies are most often considered acceptable. For example, it might be acceptable for a company if information such as a telephone number is temporarily out of date.

Most application-specific directories are limited by default to the type of data they are required to store in order to support the application. For example, a directory specialized for system administration might be limited to storing only system specific information such as domain information, servers, people, and groups. However, this limitation is imposed not so

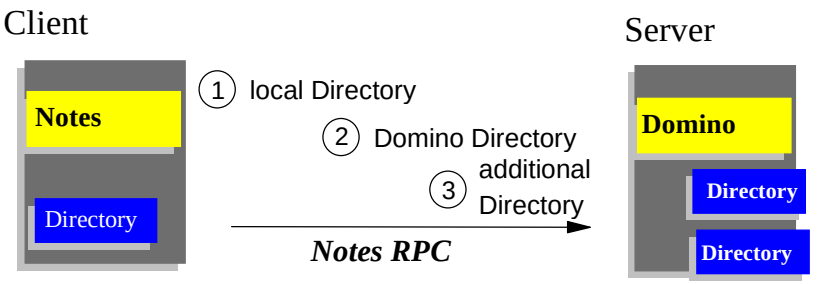
much by the architecture of the directory, but by its design. If the directory is extensible, i.e., its schema can be extended to support new objects and/or new attributes, it can be configured to store new types of information, making it more useful to a wider variety of applications and programs.

Another important difference between a directory and a general-purpose database is the way information is accessed. Relational databases support a standardized, very powerful access method called SQL (Structured Query Language), which allows complex update and query functions but at the cost of program size and application complexity. LDAP-enabled directories, on the other hand, use a simple access protocol optimized for directory operations, that is relatively easy for application developments. LDAP directories and how they work will be explained in a later section of this chapter.

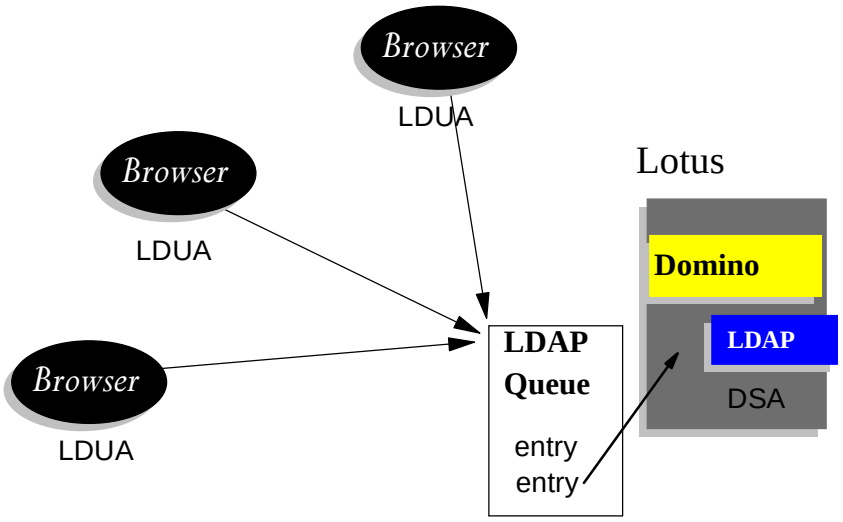
Directories are not intended to provide as many functions as general-purpose databases, and so can be optimized to economically provide applications with rapid access to directory data in well-distributed environments. Because the intended use of directories is restricted to a read-mostly, non-transactional environment, both the directory client and the directory server can be simplified and optimized. A good example is the Domino Directory used by both the Notes Client and the Domino Server.

Directory Clients and Servers

Directories are usually based on the client-server model. An application that wants to read or write information in a directory does this according to the system used and the configuration options specified. Some systems do not allow direct access to directory information, preferring instead an API (Application Programming Interface) call to send a request and wait for the corresponding response. A server process then accesses the information in the directory on behalf of the requesting application. The request is performed by the directory client, and the process that looks up information in the directory is called the directory server. In general, servers provide a specific service to the client. Sometimes a server might become the client of other servers in order to gather the information necessary to process a request. This is also called a directory referral.



A directory service is only one type of service that might be available in a client/server environment. Other common examples are file services and mail services. The client and server process may or may not be on the same machine. A server must be capable of serving many clients. Some servers can process client requests in parallel. Other servers queue incoming client requests for serial processing if they are currently busy processing another client's request.



An API defines the programming interface, a particular programming language uses to access a service. The format and contents of the messages exchanged between clients and server must adhere to an agreed protocol. LDAP defines such a message protocol used by directory clients and directory servers.

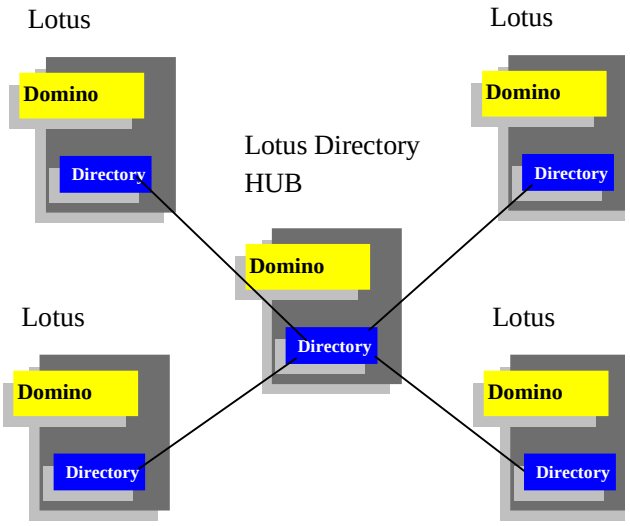
Distributed Directories

The terms local, global, centralized, and distributed are often used to describe a directory or directory service. These terms mean different things to different people in different contexts. In this section, these terms are explained as they apply to directories in different contexts.

The information stored in a directory can be local or global in scope. For example, a directory that stores local information might consist of the names, e-mail addresses, public encryption keys, and so on of members of a department or work group. A directory that stores global information might store information about the entire company.

The clients that access information in the directory can be local or global. Local clients might all be located in the same building or on the same LAN. Global clients might be distributed across the continent or the world.

The directory itself can be centralized or distributed. If a directory is centralized, there is one directory server that provides access to the directory. If the directory is distributed, there is more than one server that provides access to the directory. When people refer to a distributed directory, they are usually referring to distributed directory servers, as in the following chart:



When a directory is distributed, the information stored in the directory can be partitioned or replicated.

- When information is partitioned, each directory server stores a unique and non-overlapping subset of the information. That is, each directory entry is stored by only one server.
- When information is replicated, the same directory entry is stored by more than one server.

In a distributed directory, some information may be partitioned while some information may be replicated.

The scope of information stored in a directory is often set by the requirements of an application or a set of applications. The criteria for the distribution of directory servers and the distribution of data either by partitioning or replicating are often determined by the resultant performance and availability of the directory. For example a distributed and replicated directory might perform better because a read request can be serviced by a

server nearby, whereas a centralized directory may be less available because it is a single point of failure. On, the other hand, a distributed directory might be more difficult to maintain because multiple sites (possibly under the control of multiple administrators) must be kept up-to-date and in running order.

Directory Security

The security of information stored in a directory is a major consideration. Some directories are intended for public access from the Internet, with the caveat that most users should not be able to perform operations against the directory. A company's directory servicing its intranet can be stored behind a firewall to keep the general public from accessing it, but more security control is needed within the intranet itself.

A directory should support the basic capabilities needed to implement a security policy. The directory might not directly provide the underlying security capabilities, but it might be integrated with a trusted network. First, a method is needed to authenticate users. Authentication verifies that users are who they say they are. A user name and password are a basic authentication scheme. Once users are authenticated, it must be determined if they have the authorization or permission to perform the requested operation on the directory object. Authorization is often based on access control lists (ACLs). An ACL is a list of authorizations (such as read, modify, and write) that may be attached to objects and attributes in the directory.

Security Attribute	*Value*
Name	Read, Write, Change
E-mail address	Read, Write, Change
Social Security Number	No Access

The Directory as an Infrastructure Service

A directory that is accessible by all applications is a vital part of the infrastructure supporting a distributed system. A directory service provides a single logical view of users, resources, and other objects that make up a distributed system. This allows users and applications to access network resources transparently. That is, the system is perceived as an integrated whole, not a collection of independent parts. Objects can be accessed by name or function without knowing low-level identifiers such as host addresses, server names, and e-mail addresses.

Directory-Enabled Applications

A directory-enabled application is one that uses a directory service to improve its functionality, ease of use, and administration. Today many applications make use of information that could be stored in a directory. For example, let's look at a group calendar application that is used to schedule meetings of company personnel in various conference rooms.

Consider the scenario that the calendar application does not use a directory service at all. In this case, a user trying to schedule a meeting has to remember the room number of every conference room appropriate for the meeting as well as other criteria such as room size, availability, and so on. The user also has to remember the names and e-mail addresses of every attendee who needs to receive a meeting notice. Such an application would obviously be difficult to use. If, on the other hand, conference room information (size, location, equipment) and personal information (name, e-mail address, phone number) could be accessed from a directory service, the application would be much easier to use, and its functionality improved.

In the past, the developers of directory-enabled applications were faced with the problem that directory services were often limited to certain platforms as well as proprietary APIs and protocols. Therefore application developers often took the approach of developing their own application-specific directory, which in turn compounded the problem.

The Benefits of a Common Directory

An application-specific directory stores only the information needed by a particular application and is not accessible by other applications. An application-specific directory could be as simple as an editable text file, or it could be stored and accessed in an undocumented, proprietary manner.

In such an environment, each application creates and manages its own application-specific directory. This quickly becomes an administrative nightmare. The same e-mail address stored by the calendar application might also be stored by a mail application and by an application that notifies system operators of equipment problems. Keeping multiple copies of information up-to-date and synchronized is difficult, especially when different user interfaces and even different system administrators are involved.

What is needed is an open, standards-based directory. If application developers could be assured of the widespread existence of such a directory service, then application-specific directories might not become necessary. The common directory must be based on an open standard for protocol and API access that is supported by most vendors and most platforms. It must be extensible so that it can hold the types of data needed by arbitrary applications. And it must provide full functionality without requiring excessive resources on smaller systems. Since more users and applications

will access and depend on the common directory, it must be robust, secure, and scalable.

LDAP is the protocol to be used to access this common directory infrastructure. Like HTTP (hypertext transfer protocol) and FTP (file transfer protocol), LDAP will become an indispensable part of any intranet/Internet infrastructure.

Note There will always be the requirement to store system specific information in a system- or application-specific directory. For example, it will not be possible to run a large NT 2000 infrastructure without at least a small Active Directory to host all security and server-specific information. The same will be true for a Domino R5.0 directory, which will for the foreseeable future still be required to host configuration and security settings. For a common directory, you should currently focus on Persons, Passwords, and Groups. System configurations and Security Settings (ACLs) are likely to follow in the future.

The Domino Directory

The Domino Directory, which previous releases referred to as the Public Address Book (PAB) in R4.x and the Name and Address Book (NAB) in R3 and earlier, is a database that is automatically created on every Domino server. The Domino Directory is the primary configuration database in a Domino enviroment and serves two purposes. It is a directory of information about users, servers and groups, as well as other objects that might be added to the directory, for example, printers. It is also a tool that administrators use to manage the Domino system. For example, administrators create documents in the Domino Directory to connect servers for replication or mail routing, to register users and servers, to schedule server tasks, and so on.

Typically, a Domino Directory is associated with a Notes domain. When you register users and servers in the domain, you create Person documents and Server documents in the Domino Directory. These documents contain detailed information about each user and server.

When you set up the first server in a Notes domain, Domino automatically creates the Domino Directory database and gives it the file name NAMES.NSF. When you add a new server to the domain, Domino automatically creates a replica of the Domino Directory on the new server.

Directory Service Features

In addition to the Domino Directory itself, Domino provides three directory service features: the directory catalog, directory assistance, and the LDAP service. These features help users find user names, e-mail addresses, and other information in the Domino Directory.

The directory catalog consolidates key information about users and groups from one or more Domino directories into a small, lightweight database. Notes users who use a local copy of the directory catalog — a mobile directory catalog — can quickly address mail to users throughout the organization, even if the organization uses a large directory and/or multiple directories. In organizations with multiple Domino directories, a directory catalog on a server combines these directories into a single database so that a server can look up names in one database rather than in multiple Domino directories.

Directory assistance is a feature that helps manage name lookups in organizations that use multiple Domino directories and/or third-party LDAP directories. A directory assistance database associates each Domino directory/LDAP directory with specific hierarchical names. So when looking up a hierarchical name, Domino first searches the directory that contains names in that hierarchy.

You can set up a Domino server to run the Lightweight Directory Access Protocol (LDAP) service to enable LDAP clients, like Web browsers and Notes clients, to search for and modify information in the Domino Directory. The Domino LDAP service is LDAP v2 and LDAP v3 compliant.

Documents in the Domino Directory

The Domino Directory contains documents that control directory services, manage server tasks, and define server-to-server communication. Domino automatically creates some documents when you perform certain administrative tasks. For example, Domino creates a new Person document when you register a user. The table below explains the different types of documents in Domino Directory.

Document	*Description*
Certificate	Describes a certifier ID, including public key information
Configuration Settings	Configures mail, LDAP, and the NOTES.INI file
Connection	Provides server and domain information for connecting servers for mail routing, replication, and newsfeeds
Domain	Defines a domain used in mail routing: Foreign, Non-adjacent, Adjacent, Foreign X.400, Foreign SMTP, Foreign cc:Mail, Global
External Domain Network Information	Contains names and addresses of servers in a secondary domain; allows Notes clients to connect to servers in the secondary domain
Group	Defines a list of users and servers for use in mail addressing, ACLs, and server access lists

continued

Document	*Description*
Holiday	Defines Holiday documents that users can download to their calendars
Location	Contains communication and other location-specific settings for use from a client; useful for administrators who also use the Domino Directory as their Personal Address Book
Mail-In Database	Defines the location and properties of a database that can receive mail
Person	Describes a user (Notes or non-Notes) in the directory
Program	Schedules Domino server tasks and other programs to run
Resource	Defines a resource that Notes clients can reserve by using the calendar and scheduling feature
Server	Specifies server configuration settings, including server name, cluster name, security method, port, server tasks, Internet protocol, MTA, transactional logging, and so on
User Setup Profile	Defines a standard set of configuration options for Notes clients including connections, server accounts, replicas, bookmarks, and so on

Multiple Directories

When you set up the first server in an organization, you create a Notes domain and a Domino Directory. Most organizations use only one Notes domain and register all servers and users in one Domino Directory. However, there are reasons to use multiple domains and multiple Domino Directories. For example, in a large organization in which responsibility for system administration is distributed, creating separate domains and using separate Domino directories allows you to more easily adapt to company specific requirements like subsidiaries and independent branches or business units. Another reason to use multiple domains and directories is if your organization merges with another organization that uses Domino. In that case, you may decide to retain separate domains and directories, at least temporarily. To create an additional domain and Domino Directory, you perform a first server setup.

Some organizations create an additional Domino Directory that is not affiliated with a Notes domain. You might do this, for example, to manage non-Notes users, such as Web users. Other reasons might be the need to store mail addresses and additional information about business partners, hotels, suppliers, or others. This information is considered to be private to this organization but should be accessible by everybody inside the organization, mainly through their mail client. To create an additional Domino Directory, you create the directory using the PUBNAMES.NTF template.

In a system with multiple directories, the server's primary Domino Directory is the directory in which that server is registered. A server's primary Domino Directory uses the file name NAMES.NSF. From a given server's standpoint, any Domino Directory within the organization that is not the server's primary Domino Directory is a secondary Domino Directory. In addition, an organization may use a Domino Directory along with a third-party LDAP directory. For example, an organization may have a Domino Directory on a Domino server that runs the Web service, but use a third-party LDAP directory to authenticate Web users that connect to the Domino server. If you run the Domino LDAP service, you can refer LDAP clients that use the service to a third-party LDAP directory.

It is important to plan a strategy for managing multiple directories. You can set up a directory catalog and directory assistance to make it easy to extend name lookups and client authentication beyond the primary Domino Directory.

Directory Catalogs

A directory catalog consolidates entries for users, groups, mail-in databases, and resources from one or more Domino directories into a single, lightweight, quick-access database. You can use a User Setup Profile to create a replica of a directory catalog on Notes clients, known as a mobile directory catalog, so Notes users can quickly address mail to anyone in your organization even when disconnected from the network. Type-ahead addressing searches the mobile directory catalog rather than Domino directories on a server, which reduces network traffic. You can also create a directory catalog for server use so that servers can use one database to search for names in multiple Domino directories. Typically, you create two directory catalogs: a mobile directory catalog and a server directory catalog.

A directory catalog can combine entries from many Domino directories and still be very small. For example, several Domino directories that together contain more than 350,000 users and total 3GB in size, when combined in a directory catalog, are likely to be only about 50MB. In general, each entry in the directory catalog is slightly more than 100 bytes.

By default, entries in the directory catalog include the fields in the following table, which are required to resolve mail addresses. You can include other fields that users in your organization access frequently.

Field configured by default	*Entries that use the field*
FullName	Person, Mail-In Database, Resource
ListName	Group
Type	Person, Mail-In Database, Resource, Group
FirstName	Person
MiddleInitial	Person
LastName	Person
Location	Person
MailAddress	Person
Shortname	Person
MailDomain	Person, Mail-In Database, Resource
InternetAddress	Person, Mail-In Database
MessageStorage	Person, Mail-In Database

To create a directory catalog, you create a source directory catalog database. You create a configuration document in this database in which you specify the file names of directories to combine in the catalog, the fields to include, and other configuration options. The server that stores the source directory catalog typically stores all replicas of the directories combined in the catalog.

After configuring the source directory catalog, you run the Directory Aggregator, the Dircat task, on the server that stores the source directory catalog. The Dircat task summarizes entries for users, groups, mail-in databases, and scheduling resources from all the configured Domino directories and combines the entries into the source directory catalog. If you build a mobile directory catalog, you replicate it to the Notes clients so that Notes users can easily replicate the catalog. If you build a server directory catalog, you replicate it to Domino servers that want to use it. To use a server directory catalog, the catalog file name must be included in the Public Directory Profile.

In general, you should build a source directory catalog on one server in your organization and then replicate it to servers in other Notes domains, rather than building a source directory catalog in each domain.

To keep a directory catalog up-to-date, you run the Dircat task regularly to keep the source directory catalog synchronized with directories combined in it and then replicate the source directory catalog to the clients and servers throughout the organization.

The Server Directory Catalog

You can use a server directory catalog to facilitate the authentication of Web clients registered in secondary Domino directories, to extend LDAP searches from an LDAP server's primary Domino Directory to secondary Domino directories, and to resolve the addresses of users registered in secondary Domino directories for Notes users that don't use a mobile directory catalog. Typically, you use both directory assistance and the server directory catalog, although doing so isn't required. When a search isn't satisfied by the directory catalog, the directory catalog contains document information that allows directory assistance to quickly access the correct entry in the full Domino Directory.

Note It is not necessary to configure the same directories in the server directory catalog and in directory assistance.

Facilitating Web Client Authentication

When Web clients connect to a Domino server running the Web service, you can use Person entries in a secondary Domino directory to authenticate the clients. To do this, you use directory assistance to define which directories are to be trusted and which replicas of the directory are used for this.

The server directory catalog can work along with directory assistance to facilitate Web client authentication. In fact, using both features is recommended if your organization configures multiple secondary directories in the Directory Assistance database. Using the server directory catalog is beneficial because Domino can quickly search for a common name in the directory catalog — the format users most often enter — and derive its hierarchical format without having to sequentially search multiple secondary directories. The directory catalog also contains document information that allows directory assistance to quickly access the correct Person document in the complete secondary Domino Directory.

If you use name and password authentication for Web clients, you can store the passwords in the directory catalog. To do this, you add the HTTPPassword field to the catalog configuration. Then, although you must still use HTTPPassword directory assistance to define the authentication rules, password checking occurs directly in the catalog; therefore, there is no need to access a complete replica of the secondary Domino Directory.

Note In general storing x.509 certificates in the directory catalog is not recommended because it makes the size of the directory catalog too large.

Extending LDAP Searches

Directory assistance works along with the directory catalog on a server to process LDAP searches. When LDAP clients search for fields (attributes) that are not configured in the directory catalog, the LDAP service uses directory assistance to access the complete entries in the actual secondary Domino

directories. You should add the fields that LDAP clients search for most frequently to the configuration, so that, in most cases, you use one database — the server directory catalog — to satisfy the searches. Then, searching and accessing the secondary directories will occur only occasionally.

Creating a Source Directory Catalog

After you initially build a source directory catalog, if you change any of the configuration fields in the Basics section of a directory catalog Configuration document, the next time the Dircat task runs, it rebuilds the entire catalog. In addition, when the catalog next replicates — for example, when a Notes user replicates a mobile directory catalog with the source directory catalog — a full, not incremental, replication occurs. Performing a full replication is a slower process, especially if it is done over dial-up connections.

Create separate source directory catalogs for mobile and for server use.

1. Make sure that you already prepared a server for the source directory catalog.
2. From the Domino Administrator, do the following:

   Choose File - Database - New.

   Select the server that you prepared for the source directory catalog.

   Enter a title for the catalog. You can use any title.

   Enter a file name for the catalog. You can use any file name.

   Select "Create full-text index for searching."

   Click Template Server, select a server that stores the Directory Catalog template (DIRCAT5.NTF), and then click OK.

   Select the Directory Catalog template, and then click OK.

   Keep the - Default - access as Reader so that users can't modify the catalog configuration.
3. Open the database you just created, and choose Create - Configuration.
4. Complete the "Directories to include" field. Enter the file names of the directories to include in the catalog; enter the name of the primary Domino directory (NAMES.NSF) if you want other domains to use the directory catalog and the file names of secondary Domino directories.

Location of secondary Domino Directory	*Enter*
Locally (recommended)	The file name — for example, PINBALL.nsf
Locally in a linked directory	The file name, preceded by the linked directory — for example, DIRECTORIES\PINBALL.nsf

continued

Location of secondary Domino Directory	*Enter*
Over the network on a mapped drive	The file name and path — for example, U:\DIRSERVER\PINBALL.nsf
Over the network through Domino	The file name in this syntax: portname!!!servername!!filename where: • portname is the name you gave to the port • servername is the hierarchical name of the server that stores the directory • filename is the file name for the directory on the server For example — TCPIP!!!dirserv/east/acme!!names.nsf **Note** For one Domino server to access a secondary directory stored on another Domino server, the two Domino servers must have certifiers in common or must be cross-certified. This is an easy way to look up directory entries on other Domino directories without the need for replication.

5. (Optional) Edit the "Additional fields to include" field to customize the fields included in the catalog. It's best not to remove fields from the default configuration, because they are used for optimized quick mail addressing. Read about adding and removing fields before you customize the field configuration.

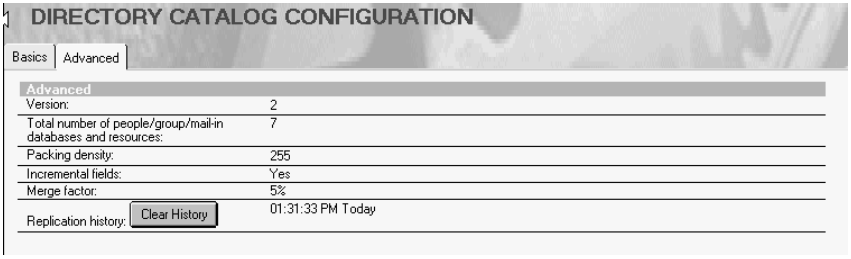
   If the directory catalog will be used by servers, add these fields: AltFullName, AltFullNameLanguage (even if users don't use Alternate Names in their certificates), and members. If the directory catalog will be used to look up Web client passwords, add the HTTPPassword field.

   If the directory catalog will be used as the mobile directory catalog, add AltFullName and AltFullNameLanguage (only if users use alternate names in their certificates). Optionally, add the members field if the majority of users want to do free time lookups when disconnected from the network. Keep in mind that adding the members field increases the size of the mobile directory catalog and involves more replication overhead.

6. (Optional) Change the default settings for these fields. Read about customizing Notes searches of the directory catalog and about customizing group types in the directory catalog before you change these settings.

Field	*Enter*
Sort by	Choose one: • Distinguished name (default) if you expect users to enter a first name followed by a last name • Last name if you expect users to enter a last name followed by first name
Use Soundex	Choose one: • Yes (default) to add Soundex values for names • No to not add Soundex values for names
Remove duplicate users	Choose one: • Yes (default) to remove duplicate entries for identical names and keep the first one encountered by the Directory Cataloger, according to the order in which you list the directories in the "Directories to include" field • No to prompt the user to select a name when duplicates are present
Group types	Choose one: • Mail and Multi-purpose (default) to include only these types of groups in the directory catalog • All to include all groups in the directory catalog

7. Close and save the document, which will then look similar to the one shown:



Building and Updating a Source Directory Catalog

After you configure a source directory catalog, you run the Dircat task to build the source directory catalog. After initially building the directory catalog, you should schedule the Dircat task to run daily to keep entries in the source directory catalog synchronized with entries in the actual directories. The Dircat task summarizes entries from Domino directories and combines the entries into aggregate documents in the source directory catalog. Each aggregate document contains the fields configured in the directory catalog, and each field combines a maximum of 255 entries. Entries in the aggregate documents are sorted alphabetically according to first name, by default.

Note Initially building a source directory catalog takes approximately one hour for every 75,000 entries on a server that is comparable to a Pentium 200 Mhz system. Subsequent updates will take less time.

When the Dircat task runs, the Miscellaneous Events view of the log file (LOG.NSF) records information about the directory catalog configuration selections, the Domino directories that Dircat processed, and the number of updates to the source directory catalog.

Note The hidden view $Users stores the aggregate documents. However, these documents are not meant for viewing and you shouldn't open them, as formatting them for viewing takes a considerable amount of time.

To Schedule the Dircat Task to Run Automatically

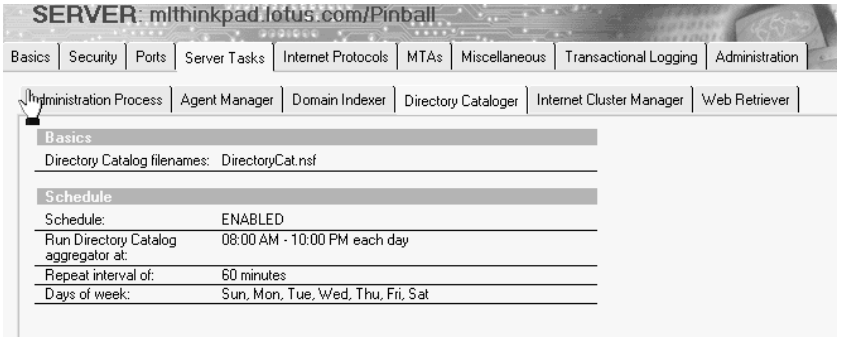
Use these steps to schedule the Dircat task to run according to the default schedule, which is daily, every 6 hours, from 8 AM to 10 PM. This is the recommended way to run the Dircat task.

1. Make sure that you have already created the source directory catalog.
2. From the Domino Administrator, select the server in the server pane on the left that stores the source directory catalog. If you don't see the server pane, click the Favorites folder.

Note Build and update the source directory catalog only on the server on which you created it. If you build and update the source catalog on more than one server, replication conflicts occur.

3. From the Domino Administrator, click the Configuration tab.
4. Choose Server - Current Server Document.
5. Click Edit Server in the Server document.
6. Click the Server Tasks - Directory Cataloger tab.
7. Complete these fields, and then click Save and Close:

Field	*Enter*
Directory catalog file names	The file name of the source directory catalog you created
Schedule	Select Enabled to enable the update schedule



Setting Up the Directory Catalog on a Server

If you've built a source directory catalog for server use, set up the directory catalog on the servers throughout your organization that will use it. In addition to setting up the directory catalog, we recommend that you also set up directory assistance.

1. Make sure that you have already built a source directory catalog for server use.
2. Create a replica of the source directory catalog on other servers in the domain that will use the catalog. Use the same file name for each replica. Domino automatically creates a full-text index for each replica.
3. From the Domino Administrator, in the server pane on the left, select the server that stores the replica of the Domino Directory you want to modify. If you don't see the server pane, click the Favorites folder.
4. Click the Files tab.
5. Select the Domino Directory, and then double-click to open it.
6. Choose Actions - Edit Directory Profile.
7. In the "Directory catalog file name for domain" field, enter the file name that you chose for all replicas of the directory catalog that you created on servers, and then click Save and Close.
8. Make sure there are connection documents that schedule replication between the server storing the source directory catalog and the servers on which you create replicas of the catalog. Scheduling replication ensures that replicas remain synchronized with the source directory catalog.

Setting Up Mobile Directory Catalogs

If you've built a source directory catalog for mobile use, you can create a User Setup Profile to set up mobile directory catalogs on Notes workstations. The User Setup Profile performs two tasks:

- It creates a replica stub (an empty replica) of the directory catalog on each user's workspace.
- It appends the catalog file name to the contents of the "Local address books" field in the user preferences for mail.

Note If you don't use a User Setup Profile, each Notes user must manually perform these two procedures.

1. Make sure that you have already built a source directory catalog for mobile use.
2. (Optional) Create a replica of the source directory catalog on other servers. Then users have more flexibility about which server they use to replicate the source directory catalog. Domino automatically creates a full-text index for the replicas.
3. From the Domino Administrator, click the Files tab, and then open the replica of the directory catalog you want to use.
4. Choose Edit - Copy As Link - Database Link.
5. From the Domino Administrator, in the server pane on the left, select the server that stores the replica of the Domino Directory that you want to use for the setup profile. If you don't see the server pane, click the Favorites folder.
6. Click the People & Groups tab.
7. Choose Domino Directories, and then choose the Domino Directory in which to create/modify the User Setup Profile.
8. Select Setup Profiles.
9. Do one of the following:
   - To modify an existing User Setup Profile, select the profile, and then click Edit Setup Profile.
   - To create a new User Setup Profile, click Add Setup Profile, and enter a name for the profile in the Profile name field.
10. Click the Databases tab, and then click the "Mobile directory catalogs" field.
11. Choose Edit - Paste.
12. Make other entries in the User Setup Profile as desired and then click Save and Close.

13. When users next connect to their mail servers to authenticate after the profile replicates to the servers, they receive a replica stub of the directory catalog. Tell users to schedule replication to occur regularly between the mobile directory catalog and a replica of the directory catalog on a server.

Directory Assistance

Directory assistance is a feature that enables users to locate information in a directory that is not the server's primary Domino Directory. Directory assistance also enables you to authenticate Web clients by using a directory that is not the primary Domino Directory on the server to which the clients connect. You can configure directory assistance for secondary Domino directories and for LDAP directories — for example, for third-party LDAP directories.

To configure directory assistance, you create a Directory Assistance database from the template DA50.NTF. For each directory that you want to use in directory assistance, you create a Directory Assistance document that describes the entries in the directory and its use.

Include secondary Domino directories in directory assistance to:

- Use the directories to authenticate Web clients
- Allow Notes users to easily address mail to users registered in the directories
- Extend LDAP client searches to secondary Domino directories

Include LDAP directories in directory assistance to:

- Use the directories to authenticate Web clients that use the Domino Web service
- Use one directory to verify Web clients' membership in groups in the directory
- Refer LDAP clients that connect to a Domino LDAP service to the directories
- Allow Notes users to verify mail addresses of users in the LDAP directories

Creating a Directory Assistance Document for a Secondary Domino Directory

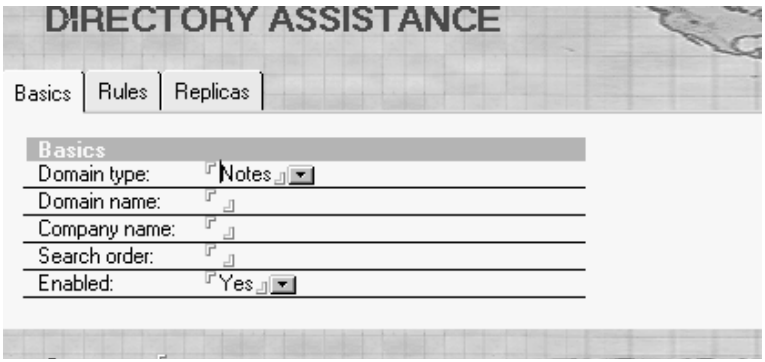
1. Make sure that you have already:
   - Created the Directory Assistance database
   - Identified the Directory Assistance database on each server that will use it
   - Planned locations for replicas of the secondary Domino Directory
   - Enabled access to the locations chosen for replicas of the secondary Domino Directory
2. If you plan to use the secondary Domino Directory to authenticate Web clients, read about and set up security before continuing.
3. If you use a directory catalog, optionally set up the directory catalog to include the secondary Domino Directory. If you use name and password authentication to authenticate Web clients and you want to store the password in the directory catalog, add the HTTPPassword field to the source directory catalog configuration.
4. If you want to extend LDAP searches to this directory, optionally do the following to customize the LDAP configuration for this directory:
   - Customize which fields anonymous LDAP users can access
   - Optimize searches for the replicas of the directory used by directory assistance
   - Allow LDAP users to modify the Domino Directory
   - Enable LDAP searches in alternate languages

Note If you allow LDAP clients to modify a secondary Domino Directory, don't use a directory catalog on the Domino server that runs the LDAP service.

5. From the Domino Administrator, in the server pane on the left, select a server that stores the directory assistance database. If you don't see the server pane, click the Favorites folder.
6. Click the Configuration tab.
7. Choose Directory - Directory Assistance.
8. Click Add directory assistance.

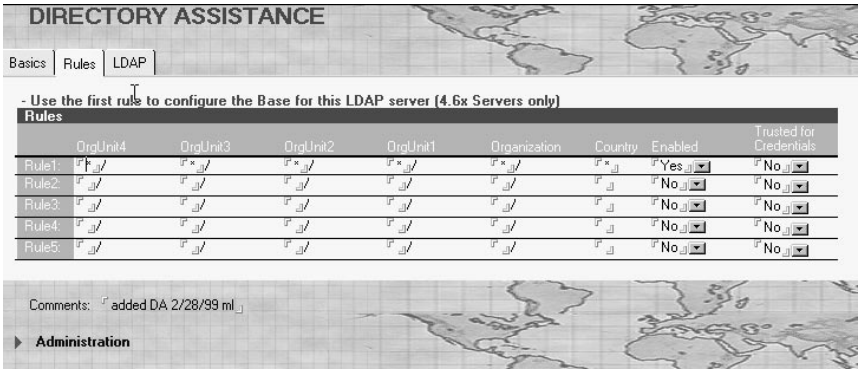
9. On the Basics tab, complete these fields.

Field	*Enter*
Domain Type	Choose Notes
Domain Name	The name of the Notes domain associated with the secondary directory. The domain name must be different from the primary Notes domain and from all other domain names configured in directory assistance. If the secondary directory is not associated with a Notes domain, invent a domain name; do not specify the primary Notes domain.
Company Name	The name of the company associated with this directory. Multiple directory assistance documents can use the same company name.
Search Order	A number representing the order in which this directory is searched, relative to other directories in the Directory Assistance database.
Enabled	Choose Yes to enable directory assistance for this directory.



10. Click the Rules tab, and then complete these fields.

Field	*Enter*
Rule #	One or more rules that describes the names in the directory. By default, the first rule contains all asterisks, indicating all names in the directory.
Enabled	Choose one: • No to disable a specific rule • Yes to enable a specific rule By default, the first rule is enabled.
Trusted for Credentials	Choose one: • Yes to allow Domino to use this directory to authenticate Web clients with names that correspond to the rule • No (default) to prevent Domino from using this directory to authenticate Web clients.



11. Click the Replicas tab, and complete these fields to specify up to five replicas of the secondary directory to use for directory assistance.

Note If you authenticate Web clients registered in the secondary directory and you also use the directory catalog, be sure to include in the Replicas tab the replica of the secondary Domino directory you used to build the source directory catalog.

Field	*Enter*
Database links	Open the database, and then choose Edit - Copy As Link - Database Link. Select the "Database links" field, and then choose Edit - Paste.
Replica #	The server name and file name of each replica of the secondary directory — for example: Server name: thinkpad.lotus.com/Pinball Directory file name: ALLNAMES.NSF

Note Use database links only on Domino R5.0 servers. Using database links may delay server startup because when you restart a server that uses directory assistance, server tasks need to retrieve database information from the remote servers the links refer to. Use database links only if the servers the links refer to are consistently available.

Note If you specify a replica in the Database links field and in the Replica field, the Replica field takes precedence.

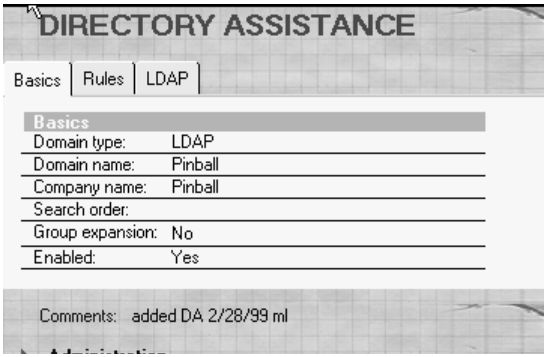
12. Click Save and Close.

Authenticating Web Clients in an LDAP Directory

You can configure directory assistance to use an LDAP directory to authenticate Web clients. In addition, you can verify a Web user's membership in a group stored in the LDAP directory.

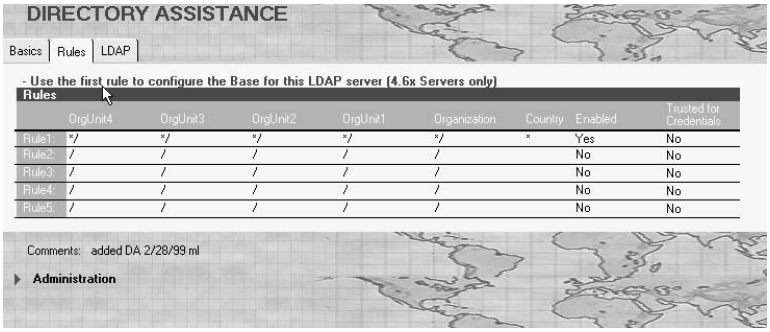
1. Make sure that you:
   - Created the Directory Assistance database.
   - Identified the Directory Assistance database on each server that runs the Web service and that will use it.
   - Set up connections between the LDAP directory server and each server that runs the Domino Web service and that will use the LDAP directory to authenticate Web clients. Use the TCP/IP ping utility to test the connection.
   - Set up security for your site.
2. From the Domino Administrator, choose File - Database - Open, select a server that stores a replica of the Directory Assistance database, select the Directory Assistance database, and then click Open.
3. Do one of the following:
   - If a Directory Assistance document already exists for the LDAP directory, select the document, and then click Edit Directory Assistance.
   - If a Directory Assistance document doesn't exist for the directory, click Add directory assistance.
4. On the Basics tab, complete these fields:

Field	*Enter*
Domain Type	Choose LDAP
Domain Name	A descriptive name that you choose; the name must be different from any other configured in directory assistance
Company Name	The name of the company associated with this directory. Multiple directory assistance documents can use the same company name.
Search Order	A number representing the order in which this directory is searched, relative to other directories in the Directory Assistance database.
Group Expansion	Choose Yes to allow directory assistance to verify the Web users membership in a group in this LDAP directory. You must also specify a naming rule corresponding to the group and select "Trust for Credentials" next to the rule. **Note** You can enable this field for only one LDAP directory. You can choose this option even if you don't use the directory to authenticate the Web users.
Enabled	Choose Yes to enable directory assistance for this directory.



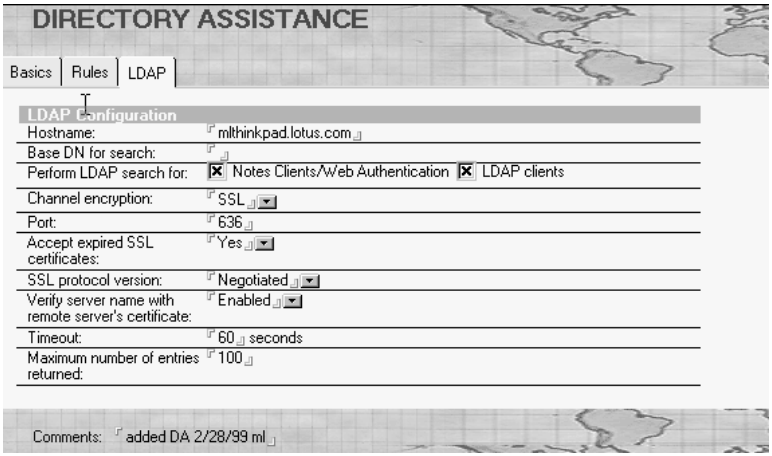
5. Click the Rules tab, and then complete these fields for each rule you want to create.

Field	*Enter*
Rule #	Enter one or more naming rules that indicate: • The names in the directory that can be authenticated if you also choose Yes in the "Trusted for Credentials" field for this rule • The order in which this directory is searched, relative to other directories configured in directory assistance **Note** In Domino Release 4.6x, the first rule specified a search base required by an LDAP directory server, rather than the "Base DN for search field below." If you used the first rule to configure a search base in Release 4.6x, when you replace the design of the R4.6x Master Address Book, the first rule is copied to the "BaseDN for search field" below.
Enabled	Choose one: • No (default) to disable a specific rule • Yes to enable a specific rule
Trusted for Credentials	Choose one: • Yes to allow Domino to use names in this directory that correspond to the rule to authenticate Web users • No (default) to prevent Domino from using names in this directory that correspond to the rule to authenticate Web users



6. Click the LDAP tab, complete these fields.

Field	*Enter*
Hostname	The host name for the LDAP directory server — for example, ldap.acme.com.
Base DN for search	A search base, if the LDAP directory server requires one. Examples: o=Ace Industry o=Ace Industry,c=US
Perform LDAP search for	Choose Notes clients.
Channel encryption	Choose one: • SSL (default and strongly recommended) to use SSL when the Domino server connects to the LDAP directory server. • None to not use channel encryption. If you choose SSL, these additional options appear: • Accept expired SSL certificates. Choose Yes (default) to accept an expired certificate from the directory server. • SSL protocol version. Select a specific protocol version to use, or select Negotiated (default). • Verify server name with remote server's certificate. Choose Enabled (default) to require that the subject line of the server's certificate includes the LDAP directory server's host name; choose Disabled not to require this.
Port	The port number to use to connect to the LDAP directory server; your selection in the "Channel encryption" field determines the default: • If you chose SSL, the default port is 636. • If you chose None, the default port is 389. If the LDAP directory server doesn't use one of the default ports, enter the port number manually.
Timeout	The maximum number of seconds allowed for a search of the LDAP directory; default is 60 seconds. If the LDAP directory server also has a timeout setting, the lower setting takes precedence.
Maximum number of entries returned	The maximum number of names that the LDAP directory server will return for the name searched. If the LDAP directory server also has a maximum setting, the lower setting takes precedence. If the server's maximum timeout is exceeded, it only returns the number of names found to that point. Default is 100.



7. Click Save and Close.

Note In Domino R5.0, you can use LDAP directories in addition to Domino directories to authenticate Web clients accessing a Domino server. Domino authenticates Web clients when you set up the client with name and password client authentication.

Note To set up authentication using LDAP directories, create the Directory Assistance database using the Directory Assistance template (DA50.NTF) or replace the design on the Master Address Book using the Directory Assistance template. In the directory assistance document for the LDAP directory, select Yes in the Trusted for Credentials field for the corresponding rule.

Note When you mark the domain as trusted, Domino searches the primary address book for the user name and password. If Domino cannot find the user name and password, it searches the secondary address books and LDAP directories for the trusted domains. You specify the order in which to search the trusted domains when you set up directory assistance.

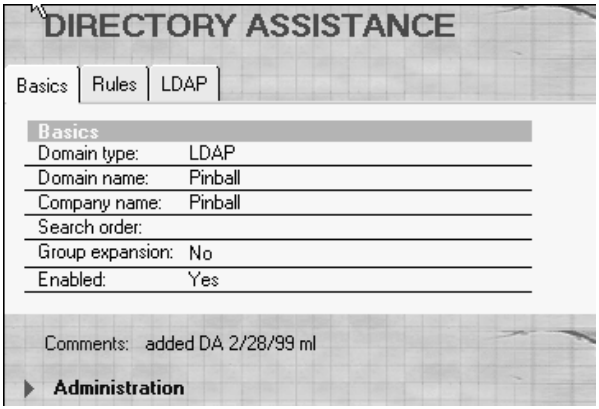
Note The hierarchical name returned by the secondary address book or LDAP directory is checked against the trusted rule in the Directory Assistance database to make sure the organization and organizational units match the rule specified. For example, if the user name returned is Dave Lawson/Lotus, then the Directory Assistance document must include */Lotus.

Referring LDAP Clients to an LDAP Directory

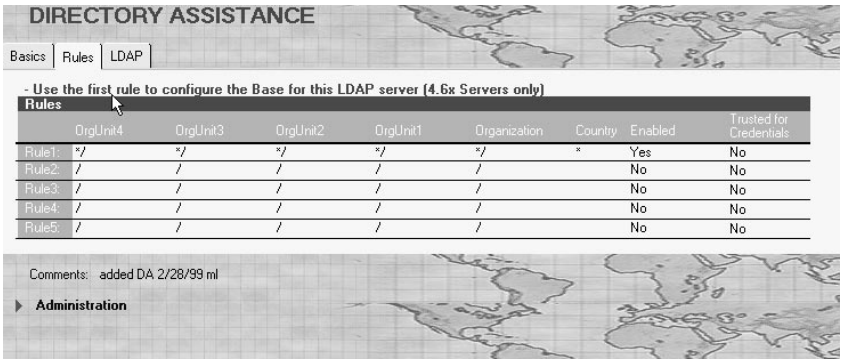
Set up the LDAP service on one or more servers and make sure you have identified the Directory Assistance database on each server that will use it.

1. From the Domino Administrator, choose File - Database - Open, select a server that stores a replica of the Directory Assistance database, select the Directory Assistance database, and then click Open.
2. Do one of the following:
   - If a Directory Assistance document already exists for the LDAP directory, select the document, and then click Edit Directory Assistance.
   - If a Directory Assistance document doesn't exist for the directory, click Add directory assistance.
3. On the Basics tab, complete these fields:

Field	*Enter*
Domain Type	Choose LDAP.
Domain Name	A descriptive name that you choose; the name must be different from any other configured in directory assistance.
Company Name	The name of the company associated with this directory. Multiple directory assistance documents can use the same company name.
Search Order	A number representing the order in which the Domino LDAP service refers clients to this LDAP directory, relative to other LDAP directories that are enabled for referrals and that are configured in directory assistance. If this is the only LDAP directory to which you refer LDAP clients, you do not need to specify a search order, unless you also use this LDAP directory for another purpose. The search order is used when LDAP clients don't specify a search base or when the specified search base corresponds to a naming rule specified for more than one directory. **Note** Within directory assistance, secondary Domino directories are always searched before LDAP directories configured for referrals, regardless of the search order specified for the secondary Domino directories. If an LDAP search is successful in a secondary Domino Directory, no referrals are returned.
Enabled	Choose one: • Yes to enable directory assistance for this directory. • No to disable directory assistance.



4. Click the Rules tab.

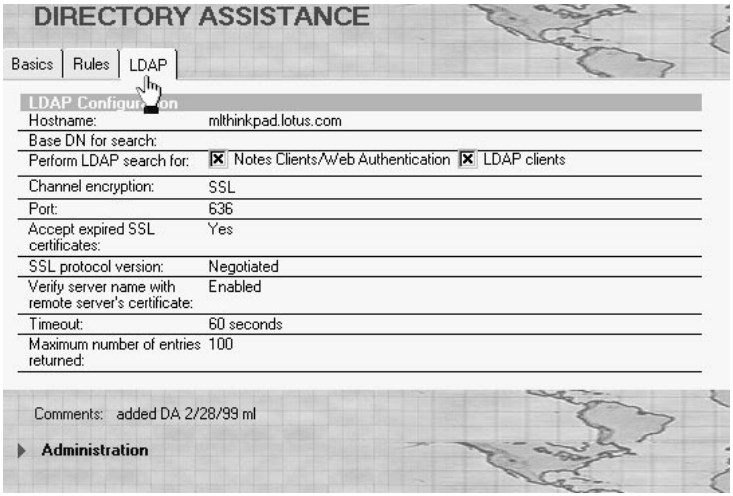


5. Complete these fields for each rule that you want to create.

Field	*Enter*
Rule #	Specify one or more rules to represent the names of entries in the directory. When an LDAP client specifies a search base that correspond to one of the rules, the Domino LDAP service refers the clients to this directory server.
Enabled	Choose Yes to enable directory assistance for this directory.

Note In Domino R4.6x, the first rule specified a search base required by an LDAP directory server, rather than the "Base DN for search field below." If you used the first rule to configure a search base in Release 4.6x, when you replace the design of the R4.6x Master Address Book, the first rule is copied to the "BaseDN for search field" below.

6. Click the LDAP tab



7. Complete these fields.

Field	*Enter*
Hostname	The host name for the LDAP server — for example, ldap.acme.com. The Domino LDAP service includes this information in the referrals.
Base DN for search	A search base, if the LDAP directory server requires one. The Domino LDAP service includes this information in the referrals. Examples: o=Ace Industry o=Ace Industry,c=US
Perform LDAP search for	Choose LDAP clients.
Port	Enter the port that LDAP clients should use to connect to the LDAP directory server. If the server uses SSL channel encryption, typically the port is 636; if the server does not use channel encryption, typically the port is 389. The Domino LDAP service includes this information in the referrals. The Domino LDAP service doesn't connect to the LDAP directory server itself; instead it includes the port information in referrals so that LDAP clients can connect.

8. Click Save and Close.

The Domino LDAP Service

LDAP uses TCP/IP to allow clients to access directory information. LDAP defines a standard way to search for and manage entries in a directory, where an entry is one or more groups of attributes that are associated with a distinguished name. A distinguished name — for example, cn=Phyllis

Spera,ou=Sales,ou=East,o=Acme — is a name that uniquely identifies an entry within the directory tree. A directory can contain many types of entries — for example, entries for users, groups, devices, and application data.

To enable the LDAP service on a server, you start the LDAP task on it (by typing "load ldap" on the server console or adding it to the line "ServerTasks" in the notes.ini). Clients that run the LDAP protocol and are set up to connect to the server and LDAP-enabled applications — for example, Notes Release 5.0 clients that have accounts for the server, Microsoft Internet Explorer (IE) clients, Netscape Communicator clients — can then query the Domino server to retrieve information about entries in the Domino Directory that meet specified criteria. For example, an LDAP client could retrieve e-mail addresses and phone numbers for all Person entries that have the last name Browning. Authenticated LDAP clients that have at least Editor access to the Domino Directory can also add, delete, and modify entries, provided that you configure the Domino Directory to allow this.

LDAP Service Features

The Domino LDAP service supports these features:

- LDAP v2 and v3.
- Anonymous access to fields that you specify; name and password authentication, SSL and x.509 certificate authentication, Simple Authentication and Security Layer (SASL) protocol.
- LDAP searches extended to secondary Domino directories; LDAP client referrals to other LDAP directories.
- Use of a third-party, LDAP-compliant server (such as the Netscape Enterprise Web server) to authenticate users that have name and passwords or x.509 certificates stored in the Domino Directory on a Domino server running the LDAP service. For information on setting up a third-party server to do this, see the documentation for the server.
- Use of LDAP clients to add, delete, and modify directory entries.
- Schema publishing and customization.
- Searches based on alternate languages.

Domino also supports these features that don't require the LDAP service:

- Command-line utility for searching LDAP directories.
- Migration tool that lets you import entries from another LDAP directory and register the entries in Domino.

Setting Up a Domino LDAP Service

TCP/IP port 389 and TCP/IP port 636 are the industry standard ports for LDAP connections over TCP/IP and SSL, respectively. You should use these default port numbers in most cases.

Set up the Domino server, and set up security for the server.

1. To allow clients to connect to the LDAP service over the Internet, connect the server that runs the LDAP service to an Internet service provider (ISP) and register the server's DNS name and IP address with the ISP.
2. Create a full-text index for the replica of the Domino Directory on the server that runs the LDAP service.
3. Start the Domino server, and then start the LDAP task.
4. If your organization uses more than one Global Domain document, you must specify the one that the LDAP service uses to return users' Internet addresses to LDAP clients. Open the Global Domain document. In the "Use as default Global Domain" field, choose Yes. A global Domain document is used to specify the settings for all used LDAP directories.
5. Set up LDAP clients to connect to the LDAP service.
6. (Optional) Customize the default LDAP service configuration and/or modify the default ports. In most cases, the LDAP service functions correctly when using the default settings.
7. To check whether you set up the LDAP service correctly, use an LDAP client or the lsearch utility to issue a query to the LDAP service.

Extending the LDAP Schema

The Domino Directory uses forms to define the LDAP object class structure of entries in the directory and uses fields to define the LDAP attributes for the object classes.

A schema defines the set of all object classes and attributes that can be stored in the directory. For each object class, the schema defines where it can be created in a directory by specifying the parents of the object class. X.500 schemas are defined in RFC 1804. The standard Domino Directory schema allows for searches of documents created from all forms except these: CrossCertificate, Location, Server\Configuration Settings, Server\Connection, Server\Domain, Server\External Domain Network Information, Server\Holiday, Server\User Setup Profile, and all hidden forms — that is forms with parentheses around their names ( ).

In Domino R5.0, you can extend the standard schema. Following specific procedures, you create subforms to add auxiliary object classes and you create forms to add new object classes.

You use the Domino Designer or a Notes API program to extend the schema. The table in the Appendix "LDAP Schema Used by Domino" will explain the current schema used by Domino R5.0.

Directory Synchronization — What Can Be Accomplished Today?

In the previous sections we have talked a lot about directories, what kinds of different directories exist, and how you can leverage the different types of Domino directories that are available in your environment.

This section will give you an introduction to the common ways to synchronize existing directories and will explain some tools and solutions available to assist you in that work.

There are two common methods available to synchronize existing directories.

- Synchronization based on having the same user name and password in all directories (point-to-point synchronization or n-way synchronization)
- Integration of existing directories into one meta directory (one way and two way synchronization)

The first method, based on the use of the same user name and password, requires that a synchronization mechanism exists between the different directory resources and types.

To do this, a common identifier (unique key) is required on all systems. This unique key can be as simple as a personal number, a social security number, or any other custom-made string, as long as it is unique and is supported by all systems. Based on the systems used, there will be cases where multiple instances of user credentials, as well as aliases, are required.

A typical set of user credentials would then look similar to the following attributes list, in the Domino Directory and the Windows NT Directory:

Domino Directory	*Windows NT Directory*
User Name Short Name Network Account Name	User Name
User Name First Name Last Name	Full Name
Password	Password
Unique Identifier	Unique Identifier

Common attributes or attribute mapping tables will ensure that during the synchronization process between the systems all unique user information is synchronized. In a few cases, directory mapping tables will be used to make it possible for the end users to match certain attributes to custom defined fields.

Note In a large installation, one additional requirement is the ability to change the user password on all involved systems. Alternatively, you can define a password change policy so that password changes are only supported on one system. Depending on the systems in place, this can be more difficult to achieve. Your security policies need to determine this requirement.

Some third-party providers that can assist you with directory out-of-the-box synchronization solutions are:

- Zoomits VIA Classic and Meta Manager for Domino
- Netvison's Synchronicity (NDS <=> Domino Directory)
- Isocor's Metaconnect

The second strategy is based on a central directory, a common directory to store all relevant information like user names, passwords, and group names as well as user certificates. This can be any directory store capable of storing all the required information and with access to the connecting systems and applications.

The challenge is to integrate all the existing systems and applications to authenticate through this central directory. Based on the available tools and APIs, this can result in multiple prompts to the end user for authentication. There are currently some limitations related to the implemenation of a common directory with a Notes/Domino environment.

The typical tools used for this will focus on application integration, such as:

- C/C++
- Visual Basic
- LDAP APIs
- Java
- ADSI (MS Active Directory Services Interface)
- JNDI (Java Naming and Directory Interface)
- LEI 3.0 (Lotus Enterprise Integrator)

Existing Single Sign On Solutions

In addition to the synchronization tools mentioned and the application toolkits listed, there is a growing list of Single Sign On (SSO) focused solutions, such as:

- IBM Global Sign On
- Passgo Single Sign On
- Netegrity Site Minder
- Neturia SFLOCK
- Cybersafe The Trustbroker Security Suite and DSSO (Defensor Single Sign On)
- Memco Proxima.

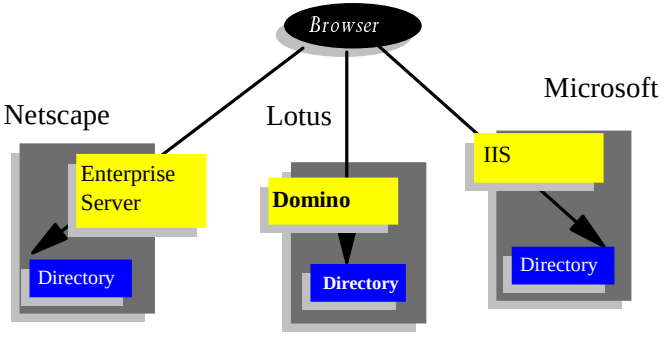
Using the Domino Directory as a Central Directory for Authentication

With Domino R5.0, it is possible to use the Domino directory as a central directory to provide authentication for Notes users and Web Browser users. This would allow Web browser users to authenticate throughout the Domino Directory and then be connected to the destination system.

But let us explain this approach step by step. We will be starting with a typical scenario, where company Pinball, Inc. has three different systems to support:

- Domino servers
- Microsoft IIS servers
- Netscape servers

The user interface is in this case a Web client. It does not matter for the moment which version or brand the company is using, as long as it supports HTTP authentication and LDAPv2 or LDAPv3.

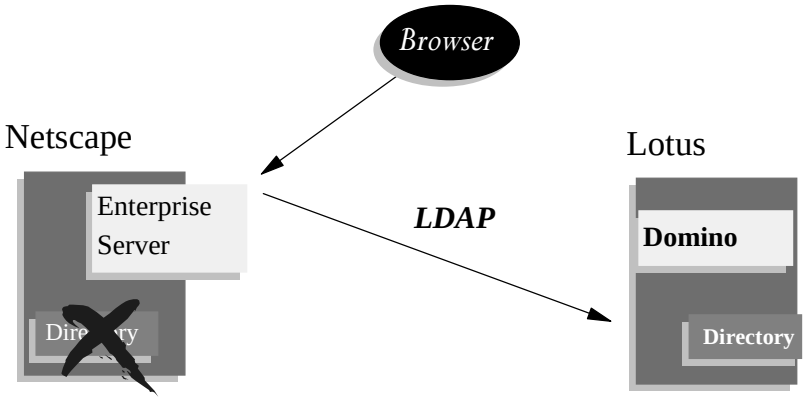


Prior to Domino R5.0, the environment of Pinball, Inc. could have looked like the figure above. With Domino R5.0, we can reduce administration considerably by removing the requirement for all three directory systems.

The users of Pinball, Inc. who are using Web browsers can access the Netscape, Domino and IIS servers but will be required to enter their user IDs and passwords each time they connect to one of the above mentioned servers. To make it more convenient to the end user, the administrator of Pinball, Inc. should opt to elect one of the existing directories as the master directory.

In our example we will select the Domino directory as the master directory. This will allow the Netscape and Microsoft HTTP servers to authenticate their Web clients through the Domino directory.

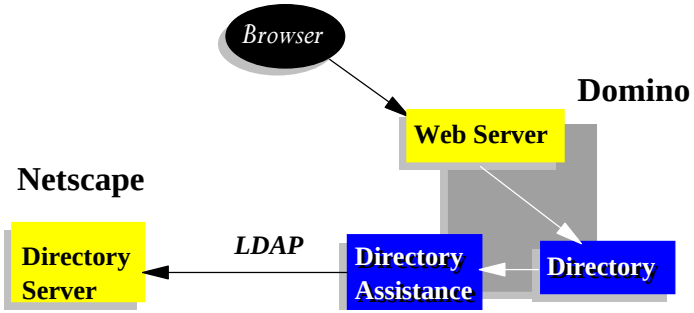
The Domino Certification Authority (CA) issues the required browser certificates to connect to all corresponding systems. This will work as long as the client is supporting the functionality of LDAPv2 or LDAPv3 access to an HTTP server. The architecture should look similar to this picture:



A Web browser, connecting to the Netscape Enterprise server, will then be redirected for authentication to the Domino Directory. There is no more need to host a directory server for the Netscape world.

Another possibility to replace one of the existing directories would be to use the LDAP functionality of the Domino Server and Directory Assistance to redirect authentication requests using LDAP to an external directory source.

This would then create a scenario similar to the following, where Domino Directory Assistance builds a trusted relationship between the Domino Directory and the Netscape Directory.



Note There are a few issues related to this scenario. Every security system hashes passwords differently, e.g., Domino cannot read the password hash of a Netscape Directory and vice versa. A possible workaround might be to use multiple password fields in the same directory. More information related to this topic will be covered in the section, Domino Services for Microsoft Internet Information Server (IIS), later in this chapter.

Domino R5.0 Application Strategies

Domino R5.0 includes some new security related features to extend the Web features of Domino. In this section, we explain how they can best be used to ease the authentication and administration process for users and administrators.

Web Realms

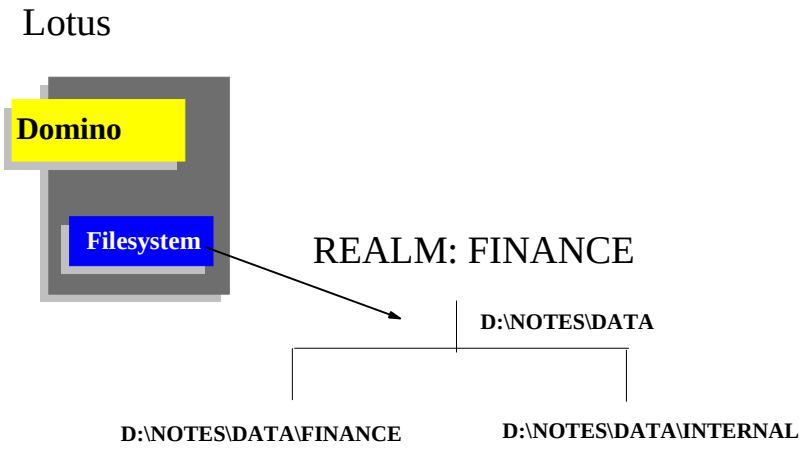
Domino R5.0 provides supports for Web realms. A Web realm is a protected area in a page tree, similar to an access restricted directory on an operating system. The goal is to avoid multiple authentication prompts for the end user. Other Web servers might implement Web realms in a different way. There is no common standard to implement Web realms across different platforms, mostly because of user application or server specific features.

With a Domino Web realm, you can specify a string of text that is displayed when users try to access a certain drive, directory, or file on a Web server. When the browser prompts the user for a name and password, the browser authentication dialog displays the realm text string. The browser uses the realm to determine which credentials (user name and password) to send to which location (the drive, directory, or file specified) with subsequent requests. The browser caches credentials for different realms to avoid prompting the user again for the same credentials. They are stored in the cookie files of the browser.

The realm string also applies to requests mapped to paths that have the specified path as their root, provided that the child paths of the root do

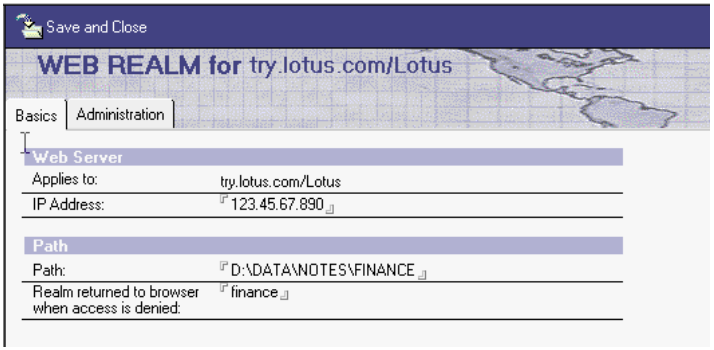
not already have a specified realm. For example, the realm string specified for D:\NOTES\DATA would also apply to a request mapped to D:\NOTES\DATA\FINANCE, if the latter did not have a specific realm specification.

If there is a realm specification for a given path, Domino uses this specification as the realm string, in our example "Finance":



Setting Up a Web Realm

1. Set up the Domino server for the Web, and set up security for the server. More information related to this topic can be found in the Domino adminstration guide.
2. From the Domino Administrator, click the Configuration tab.
3. Click Server - All Server documents.
4. Highlight the Server document to which you want to add the Web realm.
5. Click the Web button, and select Create Realm.
6. A form similar to the following will be displayed:



See the following table for more information about the values to enter:

Field	*Enter*
IP Address	The IP address of the Virtual server. This field only needs to be filled in if you are creating a Web realm for a Virtual server.
Path	The path mapped to the realm. This is the path to which the user requests access. You can define the path as a drive, directory, or file. The path is relative to the Domino server data directory.
Realm returned to browser when access is denied	A text string that describes the location (defined in the Path field) on the server. This string is passed back to the browser whenever there is an authentication or authorization failure at the location. Domino displays this text in the browser authentication dialog.

Cookies

To use session-based authentication, users must have a browser that supports the use of cookies. Domino can use cookies to track user sessions.

Cookies are commonly used to track persistent data of a server or a specific Web application. Since cookies reside in the browser, they are persistent and accessible across different Web servers and applications. The kind of information it provides depends on the server, the application, and the developer.

A shopping cart cookie might need to store different information than a cookie which is used for authentication to a free e-mail system in place of a user password.

Based on the security settings of the Web browser user, session-based authentication will not be available if the support for cookies has been turned off. This may result in the user being prompted multiple times for user name and passwords, while navigating through a Web site, depending on the access control settings for the site.

Enabling Session-Based Authentication for Domino

To enable session-based authentication for a Domino server, the server document has to be modified as follows:

Go to the server document of the server for which you want to enable session-based authentication:

1. Select the Internet Protocols tab.
2. Select the Domino Web Engine tab.
3. Change the setting "Session Authentication:" to "Enabled."

To specify the idle timeout period
Edit the Server document. On the Internet Protocols - Domino Web Engine tab, enter a value in the Idle Session Timeout field.

To specify the number of concurrent, active user sessions
Edit the Server document. On the Internet Protocols - Domino Web Engine tab, enter a value in the Maximum Active Sessions field.

To log out
To log out from a Web client, the ?logout command must be specifed in the URL. For example:

http://server-name/sessions1.nsf?logout

DSAPI

The Domino Web Server Application Programming Interface (DSAPI) is a C API for writing your own extensions to the Domino Web Server. A DSAPI extension, or "filter," is a program you create that is notified when certain events occur on the Web server, such as when a URL request is received or when a user is about to be authenticated.

For example, you might choose to write a program that performs custom authentication, which is often one part of implementing "single sign on" within a corporation. In this scenario, the DSAPI program could be notified when Domino is about to authenticate a user. The DSAPI program could then parse the user name, check the user name and password against a legacy mainframe system, and if successful, notify the Domino Web server that it has handled the event (authentication), and pass Domino the distinguished name of the user.

A DSAPI filter is built as a shared library, for example, as a DLL on the Windows platform. DSAPI is supported on all Domino server platforms.

Since the filter is written in C, you can use the Notes C API to access Domino data or other C interfaces to access other systems.

Note Because a DSAPI filter is a server extension, the filter has the privileges of the server ID when accessing Domino databases through the C API.

Note A DSAPI filter must specify two entry points: initialization and event notification. Domino calls the initialization function when the filter is loaded, and the termination function when the filter is unloaded. The filter is loaded when the Domino HTTP server task is started, and unloaded when the task quits.

The event-notification function does the actual work of the filter. Domino calls the event-notification function whenever a particular event occurs during the processing of a request. Events occur in this order:

- Domino receives the request
- Domino parses the request headers and read the request content
- Domino maps the request URL to a physical location
- Domino authenticates the user
- Domino constructs a group list for the user
- Domino completes processing the request and is about to return a response
- Domino sends the response

When Domino calls the filter's event-notification function, it passes information about the request and the event being processed. On each call the filter can decide to handle the event, with or without an error return, or decline to handle the event.

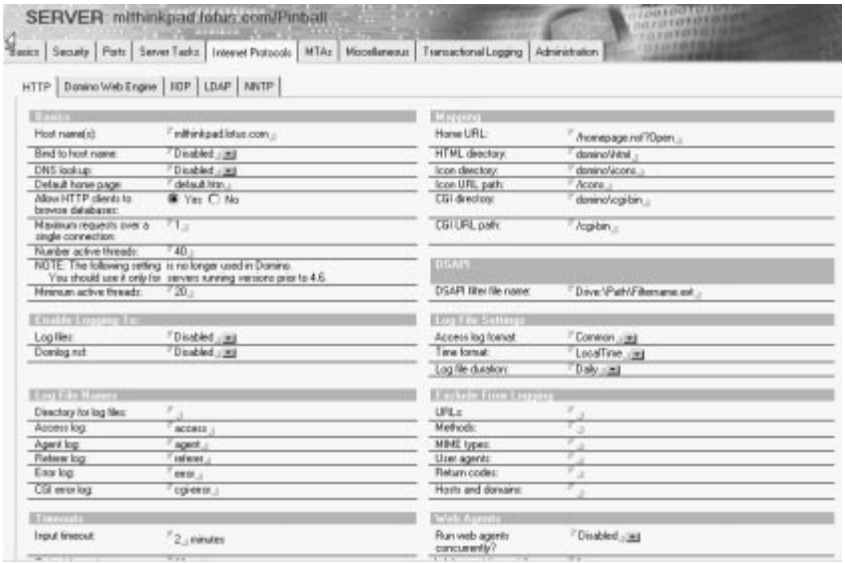
The filter may also define a termination entry point. Domino will call this function whenever the filter is about to be unloaded to be reinitialized. The filter can use this function to clean up resources used by the filter.

DSAPI also provides server call-back functions that the filter can call to get additional information or to have a service performed:

- Allocate and free memory
- Add a name to a group list
- Get information about a request
- Write a response directly to the client
- Add or change HTTP headers on the request or response

Enabling a Custom DSAPI Filter

To enable a DSAPI filter for Domino, the server document has to be modified to include the filter name:



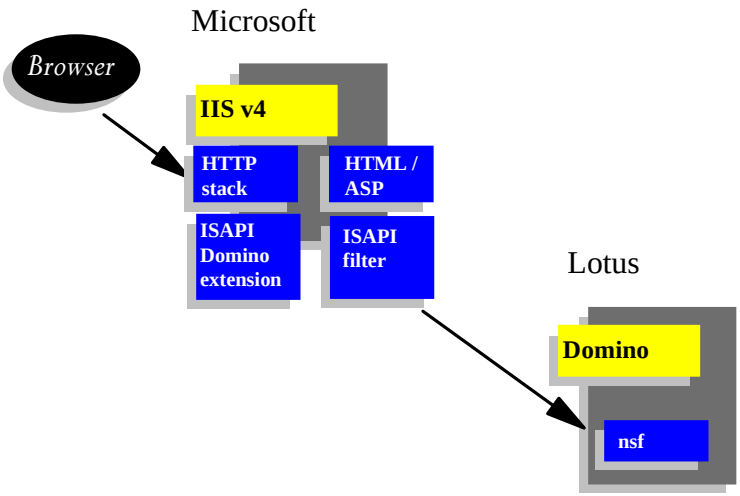
For more information about the DSAPI filter, see the Release Notes for Notes/Domino R5.0.

Domino Services for Microsoft Internet Information Server (IIS)

With Domino R5.0, it is possible to use a Microsoft IIS Web Server as the HTTP engine for Domino. If you currently use Microsoft IIS, this allows you to leverage your existing NT/IIS Web environment while providing all of the benefits of Domino as a Web application server: replication, security, indexing, messaging, collaboration, and more.

New Features of Using IIS with Domino

Any reference to Domino database URLs (e.g. http://host/database.nsf/...) will cause IIS to refer processing of that resource to the Domino Server. Any other type of pages like HTML or Active Server Pages (ASP), for example http://host/default.asp or http://host/default.html, are processed by IIS. This also means that all flat HTML files are served by IIS as well.



It is possible for IIS to take advantage of this capability because Domino is built upon a modular architecture. Therefore, it is quite simple to allow IIS to provide the HTTP services through the Lotus ISAPI interface.

IIS has several methods of security authentication. This chapter covers how that fits in with the Domino security model. In brief, if IIS authenticates the Web user, those security credentials are passed over to Domino. If the Web user comes into IIS anonymously, then normal Domino security authentication applies when accessing a Domino resource.

Why Use This Feature?

If you are currently running the Domino Server, you might be wondering why you would want to run the IIS HTTP stack, when you have been running the native HTTP stack that comes with Domino. Chances are there is probably no good reason to change.

However, if you currently use IIS, or have developers that are developing Active Server Pages, or if you have Web browser users running with Microsoft Internet Explorer 4 (IE4) and currently authenticating with NT and IIS, then you might want to consider using IIS to take advantage of the rich Web application services that can be provided by Domino.

Other advantages include using the same SSL certificate for both Web servers as well as using the NT Log as a common log file to monitor both servers.

By using the IIS HTTP server instead of using the Domino native stack, you have the opportunity to take advantage of the extended authentication mechanisms in IIS. For example, if you are using IE4 and you have already logged onto an NT domain controller, your security credentials are automatically passed through to IIS. This is an advantage for users who don't like entering multiple user names and passwords.

Background

Before going into more detail about IIS and how it integrates with Domino from a security perspective, let's take a step back and briefly look at how NT security works. This will then lead into how IIS uses NT security, and consequently how IIS security works with Domino.

NT Security Model

Windows NT uses a domain security model. An NT Domain can contain many NT Servers that will share a common security database. This security database is known as the Security Account Manager (SAM), and it maintains all of the accounts for all resources within the domain (for example, local user accounts, group accounts, service accounts, and global user/group).

It is necessary for the SAM to be located in a central location. A particular server designated for this purpose is known as the Primary Domain Controller (PDC).

Optionally, a domain can contain one or more Backup Domain Controllers (BDC). A BDC receives a copy of the SAM from the PDC, which is updated periodically. A BDC can also validate user logons.

Method of Authentication

Windows NT employs an authentication method known as Windows NT Challenge/Response (NTLM).

Challenge/Response (NTLM) Authentication

NTLM is an NT authentication protocol that is supported for backwards compatibility with Microsoft LAN Manager (LM), and is used by Windows 3.x and Windows 95/98 clients.

The following describes how the Challenge/Response authentication occurs within NT:

1. A user enters their user name and password for access to a Domain.
2. The Server challenges by sending a request to the PDC.
3. The PDC generates a random challenge string and sends it back to the Server.

4. The client generates a response string, based on the received challenge string hashed with the user's password, which is sent back to the Server with the original challenge string and user name.
5. The Server passes the user's response string back to the PDC, which then compares the received response string with the hash based on the original challenge string with the stored password hash in SAM. If all is well, the server will generate a session key.

Remember, the password is never sent over the network. In fact, the password is used as the key for the hash, like a private key. NT Challenge Response authentication uses MD4 for encryption, which is based on a 128-bit hash. With Windows 2000 (formerly known as NT Version 5), there will be a change, due to the support of a Kerberos-based ticketing system for authentication.

IIS Security

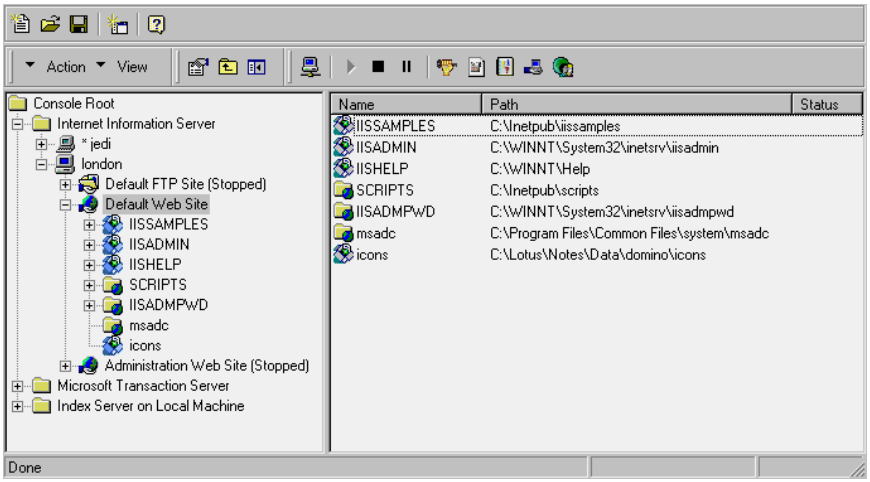
Leveraging the NT Security Model

The security model of IIS is integrated with the NT user accounts and file access permissions. When running the NT file system (NTFS), an administrator can set read, write, and execute permission on all files and directories on the Server. Since IIS acts as another NT service, it must also provide security credentials to the NT domain controller, when accessing a resource, for example HTML files on the file system. So whenever a user requests an HTML file on the NT Server, it must 'impersonate' or 'proxy' that user when actually carrying out the request on the NT system.

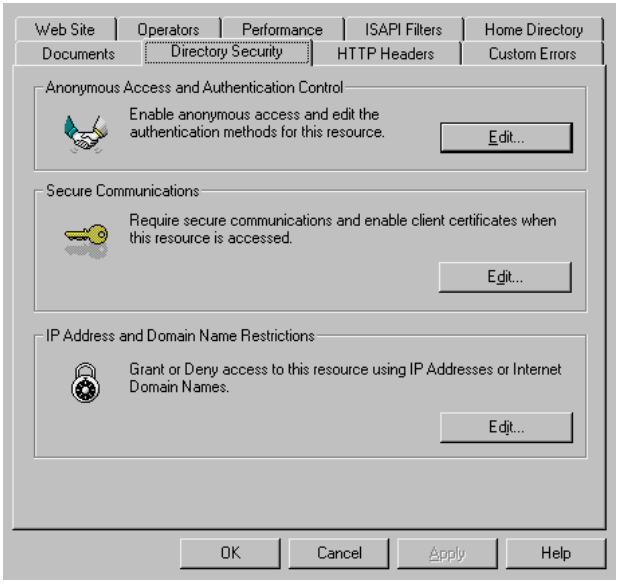
This is achieved through one of several authentication methods:

- Anonymous Access
- Basic Authentication
- Windows NT Challenge/Response
- SSL

You administer IIS through the Microsoft Management Console (MMC), a systems management application that provides a consistent interface for managing several Microsoft products.

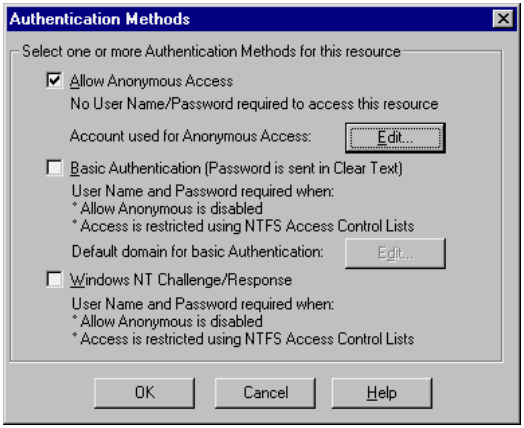


By selecting the IIS Server, you can access the properties for that Web Server, and, in particular, the Directory properties.



Note In this panel, SSL communication has been placed under "Secure Communications" rather than as a part of the "Anonymous Access and Authentication Control." This can be a bit confusing, since you can use Client Certificates for authentication within IIS.

Selecting the "Anonymous Access and Authentication Control" option, you get another panel, which presents you with the various authentication schemes.



Anonymous Access

With Anonymous access, no user name or password is required by the browser user. You may wonder how it is possible for NT to allow browser clients to access NT resources without a password. NT achieves this by 'impersonating' a special anonymous user account that is created when IIS is installed. The special user account takes the format of IUSR_machinename, where machinename is the hostname of the IIS Server. Whenever IIS is configured for anonymous access, it uses this special account for accessing NT resources.

Basic Authentication

Basic authentication works in the same way as mentioned earlier in this chapter, the user name and password is sent over the network using base64 UUEncoding. The user name and passwords map onto NT user name and hashed passwords held by Windows NT. IIS checks the user name and password locally or against the NT Domain SAM for authentication, to allow access to HTML files on the NTFS file system.

Windows NT Challenge/Response

IIS is able to take advantage of the NT Challenge/Response (NTLM) authentication process, which is only supported by users using Internet Explorer. What this means is that if a user on a Windows machine logs onto an NT machine, IE will automatically and invisibly send their security credentials (user name, domain name, and hash) on to IIS for authentication.

Note The NT Challenge/Response protocol maintains a TCP state during this handshake, which means you will not be able to use this if connecting over a proxy server, effectively limiting the use of this feature to secure intranet environments.

Which Authentication System to Use?

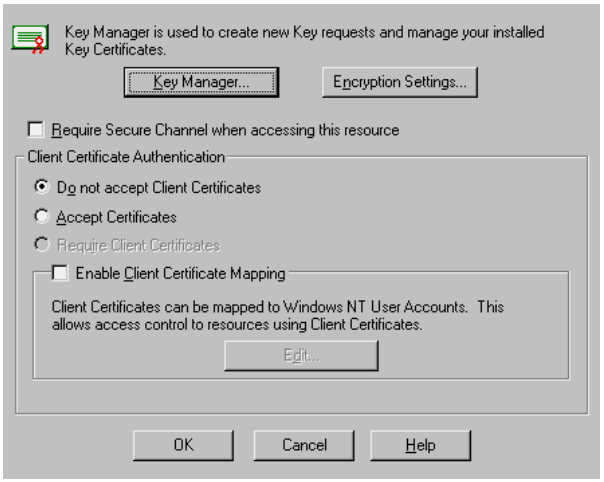
For most sites, anonymous access is acceptable, since there may only be small sections of the Web site which need to be secured. With IIS you can use any, all, or a combination of the authentication methods mentioned. A consequence of using NT Challenge/Response for authentication is that IIS does not know the password or the password hash of the user. If IIS needs to access an NT resource on another NT machine (for instance, a back-end database), the remote NT Server will challenge IIS to prove who is accessing the resource. This causes IIS a problem since it is impersonating the user and does not have the user password hash or password to respond to the remote challenge. There is no way to solve this, except to use Basic Authentication or X.509 Client Certificates.

Leveraging the Internet Security Model

X.509 Client Certificates

IIS can use X.509 digital certificates for authentication, and can also encrypt the session using SSL. If you want to use client certificates for authentication, map each certificate with the NT user that is stored in the SAM. The administrator will need physical access to the certificate. You could also write some ASP code to pull the certificate information automatically into the NT SAM.

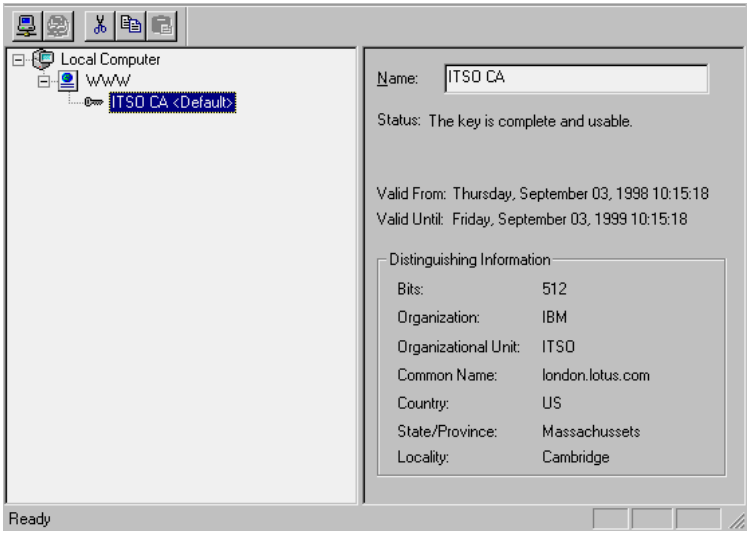
On selecting the "Secure Communications" option from the "Directory Security" tab, you are presented with the following.



From this tab, you have several options available to you:

- Enable SSL for server authentication.
- Request and install a Server certificate, through the Key Manager application.
- Accept X.509 Client certificates for authentication, but not necessarily through SSL encryption (map X.509 Client certificates to NT User accounts held in the SAM).

To enable SSL for server authentication, select "Require Secure Channel when accessing this resource." As an administrator, you can set up Server authentication by generating a certificate request through the Key Manager utility.



The Key Manager utility can generate a Server certificate request either from a recognized CA or an internal CA, for example the Domino CA, or Microsoft Certificate Server. Once the Server Certificate has been signed by the CA, Key Manager will install it into IIS.

How Domino Integrates With IIS

Now that you have a basic understanding of the NT security model, and how IIS uses NT for security, and manages X.509 certificates, we will explain how Domino fits into all of this.

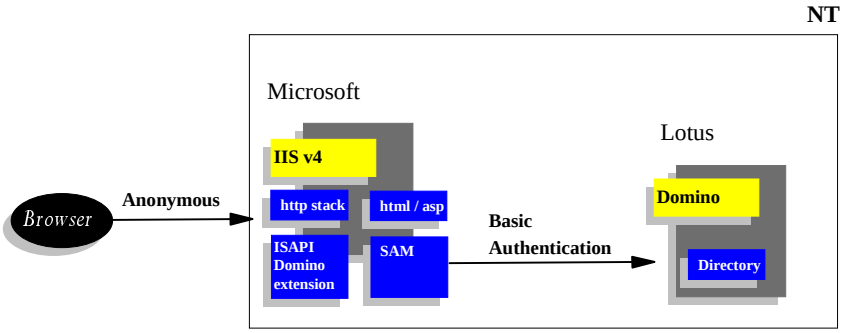
To enable Domino to use the IIS HTTP stack, you need to configure the Domino server to run as an ISAPI extension to IIS. ISAPI is the Internet Server Application Programming Interface supported by IIS. Developers use this interface to create programs, called extensions, that can extend the capabilities of IIS. They are basically DLLs that let you modify server request processing.

How Domino Leverages the IIS Security Model

IIS requires user authentication in order to control access to resources owned by IIS such as the file system and Active Server Pages. If a user requests access to a Domino resource, IIS passes the authentication information to Domino. The information passed depends on the combination of authentication methods enabled on the IIS server. Domino therefore trusts the IIS authentication.

Anonymous Authentication

If a browser user authenticates with IIS anonymously (the user does not supply a user name or password), then accesses a Domino resource, normal Domino Server authentication rules apply.

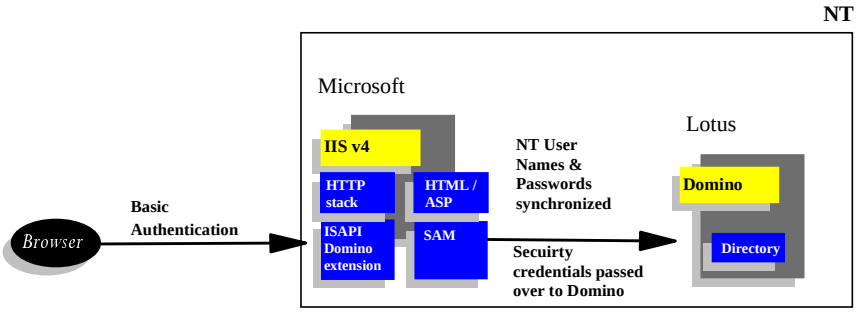


Notice that when a browser connects to IIS anonymously, and then accesses a secure Domino resource, the normal Dcomino Basic Authentication process occurs.

Basic Authentication

If a browser user authenticates with IIS through Basic Authentication (the user supplies a user name and password in the clear to IIS) and then accesses a Domino resource, those security credentials are automatically passed over to Domino. The user name entered at the browser is automatically compared against a hidden view called $Users (this provides support for more name variations and therefore, less security) or the $LDAPDN view, which provides support for fewer name variations and, therefore, more security. The view used by Domino depends on the option set in the Domino Directory.

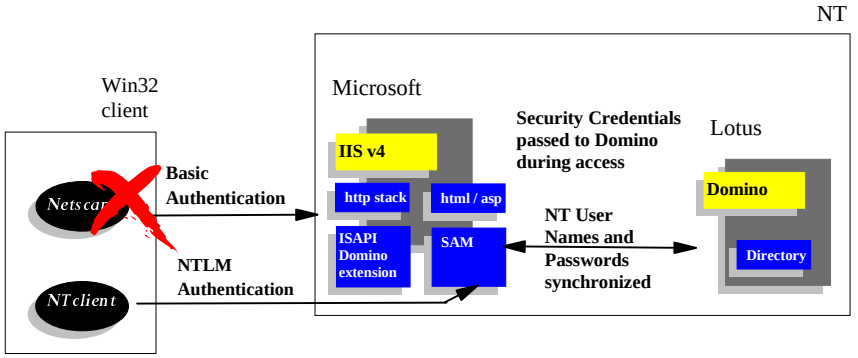
Note The NT password and the password stored in the Person document in the Domino directory are not required to be the same.

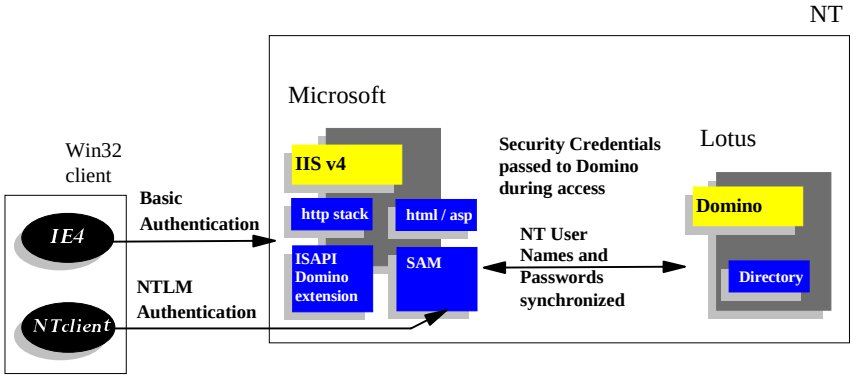


If a browser user accesses a secure file through IIS, IIS then authenticates with the NT, and, if correct, passes these security credentials to Domino.

NT Challenge/Response (NTLM)
If the browser user authenticates with IIS through NTLM (the user logs onto NT) and IE has automatically passed its security credentials to IIS, then if the user accesses a Domino resource, those security credentials are automatically passed over to Domino. The NT account name recognized in the SAM is compared against the User name field of the Person document in the Domino Directory. The User name field must have an alias that refers to the NT account name that follows the syntax "domain\username" or "machinename\username."

Note The NT password does not have to be the same password held in the Domino directory, because IIS does not know what the password of the user is, since it only receives the password hash. As a result, Domino trusts that NT has correctly authenticated the browser user name and password with IIS, if you are using Microsoft Internet Explorer. This does not work if you are using Netscape Communicator.

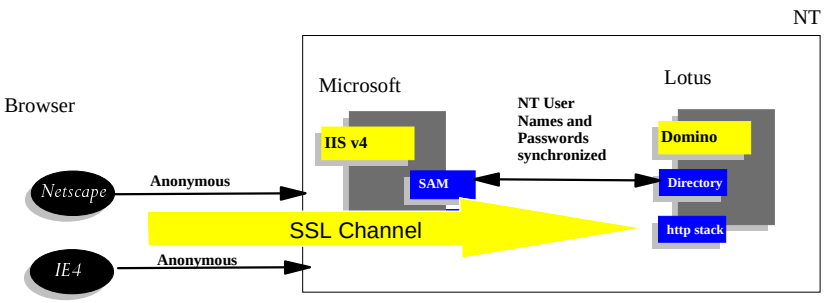




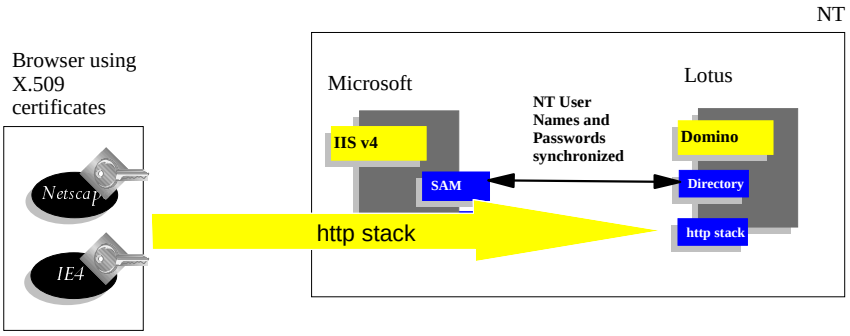
X.509 Client Certificates

It is possible for the browser user to authenticate with IIS using X.509 Client certificates running SSL. This means is that the administrator would previously have had to manually accept the client certificate and map it to an NT User account. IIS will trust the client certificate if it has been signed by a CA that is trusted, also called a "Self Signed Certificate." If it does, it will check to see if it maps with an NT User account. Assuming it does, NT will then authenticate the user and Domino will trust the authentication the same way it does in NTLM.

Note Domino supports using either client certificates alone or in combination with any of the other authentication options (Anonymous, and so on). In all cases, Domino relies on IIS to verify the certificate signer. The certificates do not have to be stored in the Domino directory since it uses the common name of the certificate sent from IIS. No additional configuration of the Domino server is necessary other than enabling client certificates on the SSL port. Please refer to the Domino R5.0 Release Notes for more details about SSL and Domino Services for IIS.



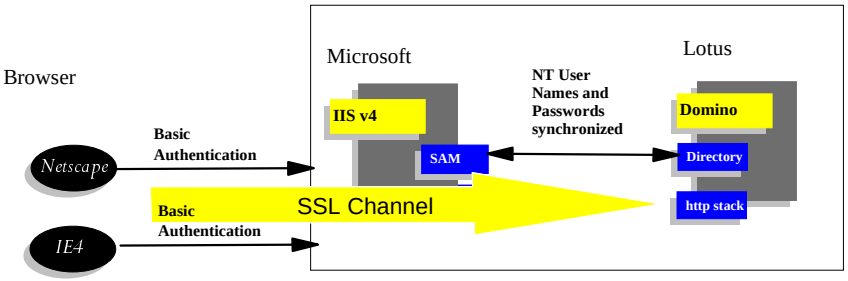
Here you can see how the X.509 certificate sent by the browser is compared against the mapped list of NT User names. Assuming IIS trusts the browser certificate and there is a corresponding mapping to an NT User name, NT authenticates the user and the credentials are passed over to Domino. Again Domino trusts the information provided by IIS.

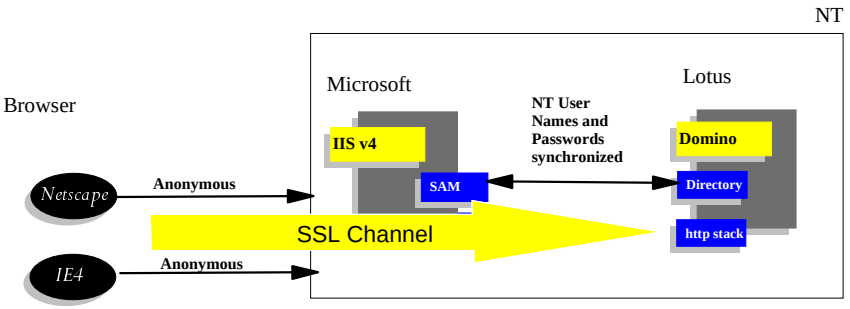


IIS also allows you to enable X.509 Client Authentication, without enabling SSL for encrypting the channel.

SSL — Server Authentication

If you set up IIS to use SSL without Client Certificates, then only Server authentication will occur between the browser and IIS; normal IIS authentication rules apply, with the channel now encrypted. If the browser user then tries to access a Domino resource, all network traffic to the Domino server is also encrypted using the SSL channel initiated by IIS.





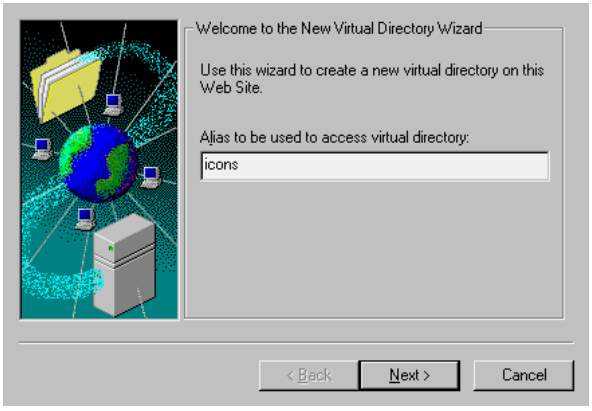
In both of these examples, standard IIS authentication occurs, 'Basic Authentication' and 'Anonymous,' with the added security of channel encryption using SSL, assuming the browser trusts that the IIS Server has a trusted root certificate. If you want to use Domino certificates, the IIS Server requires that the Domino server certificate to be added as a trusted root using the Microsoft Management Console.

Main Steps for Configuring IIS

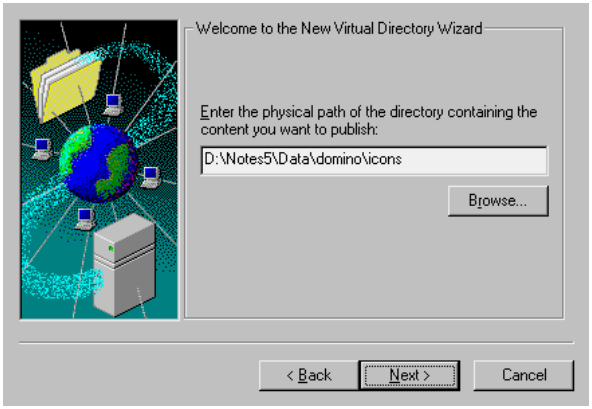
We will not attempt to give you a step-by-step guide for installing and configuring IIS, since this is covered by the standard documentation. Here is a short checklist to make sure the important steps are done:

1. Set up a virtual directory for Domino Web icons.

   In the MMC, right-click on the IIS Web site icon and choose New - Virtual Directory.



   For the Virtual Directory Alias, enter "icons." Click Next.

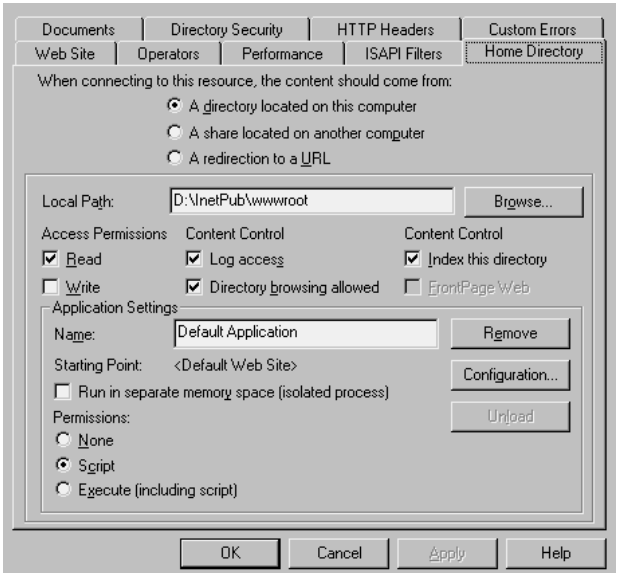


Enter the physical directory path of the Web icons for the virtual directory. For example, D:\NOTES5\DATA\DOMINO\ICONS. Click Next.

Enter the permission settings (the default settings are sufficient) and click Finish.

2. Set up the Domino ISAPI extension.

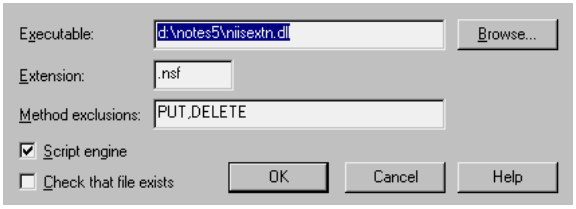
In the MMC, right-click on the IIS Web site icon and select Properties.



Click the Home Directory tab on the Properties dialog box.

Click the Configuration button.

Click Add.



In the Executable field, enter the full path name of the NIISEXTN.DLL file for example, D:\NOTES5\NIISEXTN.DLL. This file comes together with the Domino server files.

In the Extension field, enter ".NSF".

In the Method exclusions field, enter "PUT,DELETE".

Leave the "Script engine box selected and the "Check that file exists" box deselected.

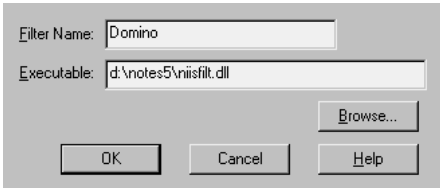
Click OK twice.

3. Set up the Domino ISAPI filter.

   The filter is used to ensure that user credentials are correctly passed through to Domino. It is also used to enable the "/?Open Server" command.

   In the MMC Web site Properties dialog box, click the ISAPI Filters tab.

   Click Add.



   In the Filter Name field, enter any text that you want. For example, Domino.

   In the Executable field, enter the path name of the NIISFILT.DLL filter. For example: D:\NOTES5\NIISFILT.DLL. If you are installing on a Windows NT Alpha machine, the DDL name is AIISFILT.DLL. This file comes together with the Domino server files.

   Click OK twice to exit the Properties dialog box.

4. Check the NT file permissions.

   Any Domino operation that uses the file system, such as opening a database or writing a temp file, is subject to NT directory and file permissions. In order to determine file-system access rights, IIS always authenticates a World Wide Web user as a specific NT account name.

For Domino to operate reliably under IIS, the following must all have NT permissions set to "change" access or higher for all Web users:

- The domino data director and all its subdirectories
- Any database files reached by database pointer files
- The Notes ID file if it is in another directory

Be sure to verify that your permissions are correct, and that default access hasn't been modified.

5. Install the standard Domino Java applets.

   Domino R5.0 includes standard Java applets such as the view, rich text, and outline applets. The Java class files that support these applets must be located in the root directory of the Web site. The Domino install program places the class files in the Notes program directory, which acts as the root directory for the Domino HTTP server task. However, when you run Domino for IIS, the IIS Web site "home directory" acts as the root directory. Therefore, in order to use the Domino applets under IIS, you must copy all of the files that have a .JAR extension from the Notes program directory to the IIS home directory.

Security Considerations When Using Domino with IIS

Server Document Settings

The Domino ISAPI extension uses a subset of the fields in the Server document that apply to the HTTP task for the full Domino server. The following table lists all of the Server document fields that are used by the HTTP task, and indicates which of them apply to the Domino ISAPI extension.

Server document section/fields	*Applies to Domino for IIS?*
Security - Agent Restrictions	
Run restricted agents	Yes
Run unrestricted agents	Yes
Ports - Internet Ports - Web	
TCP/IP port number	No
TCP/IP port status	Yes
Authentication options	
Name & password	Yes
Anonymous	Yes
<all SSL settings>	No

continued

Server document section/fields	*Applies to Domino for IIS?*
Internet Protocols - HTTP - Basics	
Host name	No
Bind to host name	No
DNS lookup	No
Default home page	No
Allow HTTP clients to browse databases	Yes
Maximum requests over a connection	No
Number active threads	No
Internet Protocols - HTTP - Mapping	
<all settings>	No
Internet Protocols - HTTP - Enable Logging To:	
<all settings>	No
Internet Protocols - HTTP - Log File Settings	
<all settings>	No
Internet Protocols - HTTP - Log File Names	
<all settings>	No
Internet Protocols - HTTP - Exclude from Logging	
<all settings>	No
Internet Protocols - HTTP - Timeouts	
<all settings>	No
Internet Protocols - Domino Web Engine - HTTP Sessions	
<all settings>	Yes
Internet Protocols - Domino Web Engine - Java Servlets	
<all settings>	No
Internet Protocols - Domino Web Engine - Memory Caches	
<all settings>	Yes
Internet Protocols - Domino Web Engine - Character Set Mapping	
<all settings>	Yes
Internet Protocols - Domino Web Engine - Conversion/Display	
Image conversion format	No
Interlaced rendering	No
Default lines per view page	Yes
Maximum lines per view page	Yes
Default search results limit	Yes
Maximum search results limit	Yes
Make this site accessible to crawlers	Yes

The Domino ISAPI extension does not use these configuration options:

- The HTTPD.CNF file
- URL redirections and mappings in DOMCFG.NSF

However, custom error pages are still supported.

Limitations and Features Supported by Domino for Microsoft IIS

Domino for IIS uses most of the features that the native Domino HTTP service uses. The following sections describe which native Domino HTTP service features are supported by Domino for IIS.

- Web application features

  Domino for IIS supports all Web application features available in Domino Designer. Java applets are also supported, including all of the designer applets installed with Domino.

  When you run a Web application using Domino for IIS, Domino for IIS processes only requests that specify a Domino database — that is, a file with an .NSF file extension. All other requests (for example, requests for HTML files and Java applets) are handled by IIS. Applications that access Domino databases generally do not need to be modified when using Domino for IIS. However, applications that access other files — such as HTML, CGI scripts, and Java servlets — may need to be modified to work with Domino for IIS.

- Database security

  All Domino database security features — for example, database ACLs and Readers fields — are supported by Domino for IIS.

- Connections and logging

  IIS handles network connection and server request logging; therefore, these features on the Domino server are not supported.

- Web server API filters

  Domino Web server API (DSAPI) filters are not supported.

- Internet Cluster Manager

  The Domino Internet Cluster Manager (ICM) is not supported. An IIS cluster manager is available from Microsoft.

- Web Configuration documents

  The Web Configuration documents defined in the Domino Directory — that is, virtual servers, URL mappings/redirections, realms, and file protection — are not supported. You can protect system files using NT file permissions.

- Servlet Manager

  The Domino Servlet Manager is not supported. You can use a third-party servlet manager for IIS, such as the IBM WebSphere Application Server.

- Server document options

  Domino for IIS uses only a subset of the native Domino HTTP service settings in the Server document. The table earlier in this chapter in the section "Security Considerations When Using Domino with IIS," lists the Server document settings that the native Domino HTTP service uses and indicates which of settings Domino for IIS uses.

Integrating Domino and Microsoft NT to Provide Single Sign On for Domino Users

On the NT platform, Lotus Domino introduced the functionality to synchronize directories by passing over the NT credentials to Lotus Notes. This allows a user who is already logged into a NT domain to access Domino resources without the need to enter an additional password.

This is accomplished by an NT service, which can be installed during Domino server setup. This feature has been supported since Domino Version 4.5 and has been updated in later releases.

Domino NT Integration

As you create a new user account in the Windows NT User Manager for Domains, you can register the new user in Notes at the same time. You can also register existing Windows NT users in Notes. Registration typically includes creating a person document, Notes ID, mail file, and a password.

Users can also be registered without mail and Notes ID files to gain authenticated access to a Domino Web server without using the Notes client.

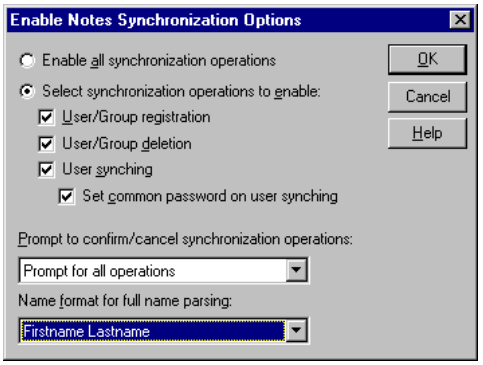
You can register NT users in Notes by using the registration defaults or by using registration options that you define. If you are using defaults, the computer on which you are making changes to Windows NT user accounts must also be a Domino server.

This server functions as the registration server (the server on which the Domino Directory entry is created) and the mail server (the server storing the user mail file).

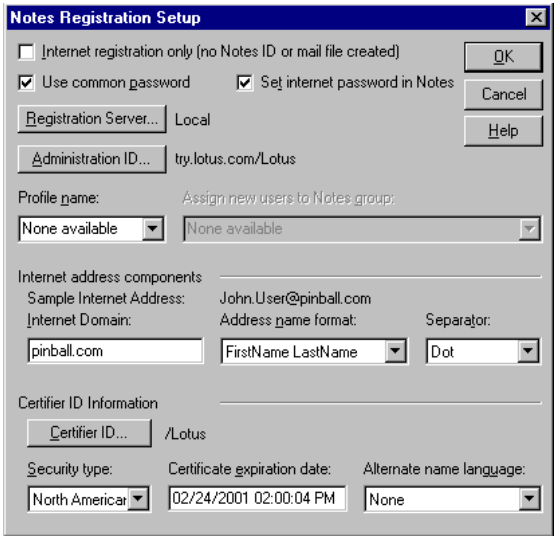
Creating New Windows NT User Accounts and Registering Notes Users Simultaneously

Before creating Windows NT user accounts and registering Notes users, do the following:

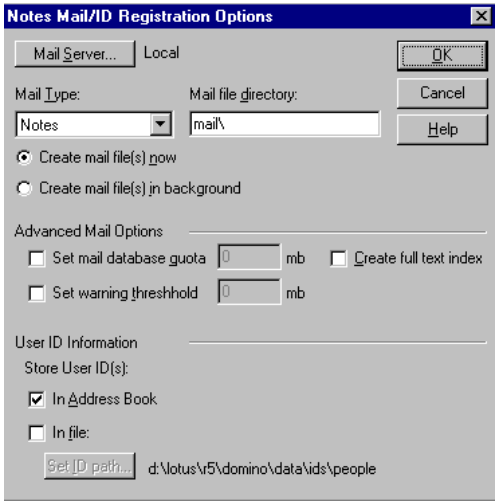
1. Make sure that Notes User registration is enabled in the Windows NT User Manager:



Change the default user registration options as necessary:



Change the default Mail/ID registration options as necessary:



Note Make sure you are a member of the local Administrator Group or local Account Operator Group in Windows NT.

2. To create new Windows NT user accounts, from the User Manager select User and proceed as instructed in your Windows NT user documentation.
3. After you finish creating the Windows NT user accounts, complete these fields, and then click OK:

Field	*Enter*
First name, middle initial and last name	Accept the default names derived from the user's full name in Windows NT
Org unit	The name of the organizational unit the user is included in. For example, if user John Smith is part of engineering, the organization unit could be Eng. The user name would be John Smith/Eng. Organizational units are useful for differentiating between users of the same name. For example, John Smith/Eng/Acme and John Smith/Doc/Acme, where one employee is a member of Engineering and the other is a member of Documentation. Each is assigned a different organization unit name.
Use common password	Assigns the same password to the user for Notes, Windows NT, and Notes Internet. Activates the Notes password for user name and the Confirm password fields. To preserve the existing Windows NT password, enter that password as the common password. If "Use common password" is not selected, activates the Notes password for user name and the Confirm password fields.

continued

Field	*Enter*
Notes password for username	The password you are assigning to this user when using Notes
Confirm password	Reenter the new Notes password for this user
Set Internet password in Notes	Enters the Internet address in the user's Person document in the Domino Directory. This field applies only if the user is registered for Notes mail. Activates the following fields: • Internet address Internet password for user name • Confirm Internet password
Internet address	Accept the default Internet address as derived from the Windows NT user name and the current host domain — for example, KCarter@domain.com This field displays if POP, IMAP, or Notes mail type is selected. The Internet address is required for Notes mail routing in Domino 5.0.
Internet password	Enter an Internet password for this user
Confirm Internet password	Reenter the Internet password for this user
Forwarding address	A forwarding address to allow the user to receive mail routed through Domino. This field displays if "Other" Mail Type is selected on the Notes Mail/ID Registration Options dialog box.
Internet forwarding address	A forwarding address to allow the user to receive mail routed through Domino. This field displays if "Other Internet Mail" is selected on the Notes Mail/ID Registration Options dialog box.

4. If you are creating more than one user account, click Close.
5. When prompted, do one of the following:

   Click Begin Registration to register new users immediately. After registration has begun, click Stop Registration at any time to stop registration after the current user registration is complete. Any users not registered remain pending.

   Click Cancel to register new users later. User information that you entered is stored until you exit User Manager.
6. To complete the process, click OK.

Note You can also register pending accounts in Notes at any time by choosing Notes - Register Notes Users Now.

Domino errors have no effect on User Manager. If a Domino or Notes error prevents a user from being registered in Notes, the user is still added to User Manager.

Registering Existing Windows NT User Accounts in Notes

Perform the following steps and change the default options if required:

1. In the User Manager User name window, select the user accounts that you want to register in Notes.
2. Choose Notes - Add selected NT Users/Groups to Notes.
3. If you are registering multiple users, choose one of the following options, and click OK:
   - "Prompt for the name and password of each user" to enter information manually for each user.
   - "Register users at once without additional prompts" to use Windows NT full names as Notes user names and to generate random Notes passwords in a database titled New User Passwords (NTSYNC45.NSF). If you choose this option, skip to Step 6.
4. If you are registering only one user or if you chose to enter user information manually, complete these fields:

Field	*Enter*
First Name, MI, Last Name	The default name as derived from the Windows NT full name. You can accept this name or change it.
Use common password	Assigns to the user the same password for Notes, Windows NT, and Notes Internet. If you are registering this user as an Internet Only user, this password field supplies the Internet or common NT/Internet password. To preserve the existing Windows NT password, enter that password as the common password.
Notes Password for username	The password you want to use, or leave blank to use a blank password. This field displays if you selected "Use common password."
Confirm password	Re-enter the Notes password for this user.
Set Internet password in Notes	Enters the Internet address in the user Person document in the Domino Directory. This field applies only if the user is registered for Notes mail. Activates the following fields: • Internet address • Internet password for user name • Confirm Internet password
Internet address	Accept the default Internet address as derived from the Windows NT user name and the current host domain — for example, KCarter@domain.com. This field displays if POP, IMAP, or Notes mail type is selected. The Internet address is required for Notes mail routing in Domino R5.0.
Internet password	Enter an Internet password for this user.

continued

Field	*Enter*
Confirm Internet password	Re-enter the Internet password for this user.
Forwarding address	A forwarding address to allow the user to receive mail routed through Domino. This field displays if "Other" Mail Type is selected in the Notes Mail/ID Registration Options dialog box.
Internet forwarding address	A forwarding address to allow the user to receive mail routed through Domino. This field displays if "Other Internet Mail" is selected in the Notes Mail/ID Registration Options dialog box.

5. When User Manager asks if you want to register the new Windows NT users in Notes, do one of the following:

   Click Begin Registration to register new users immediately.

   Click Cancel to register new users later.

6. If you chose "Register users at once without additional prompts" in Step 3, distribute the passwords to users so they can install their Notes workstations. After installation, users can create new passwords.

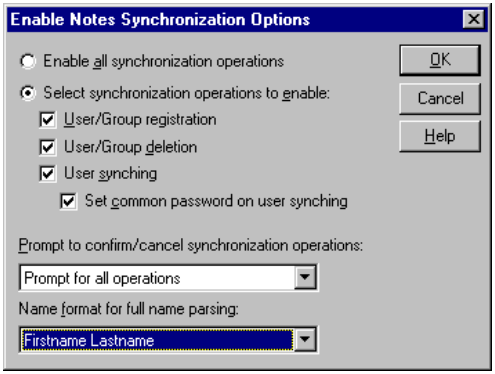
Note Automatically generated passwords apply only to Notes user IDs and not to Windows NT or Notes Internet passwords.

Enabling Notes Synchronization Operations in Windows NT User Manager

You must enable Notes synchronization features to make Notes commands available to you on the Notes menu in Windows NT User Manager.

Note By default, no synchronization operations are enabled.

1. From the User Manager, choose Notes - Notes Synchronization Options.
2. Complete these fields and then click OK:



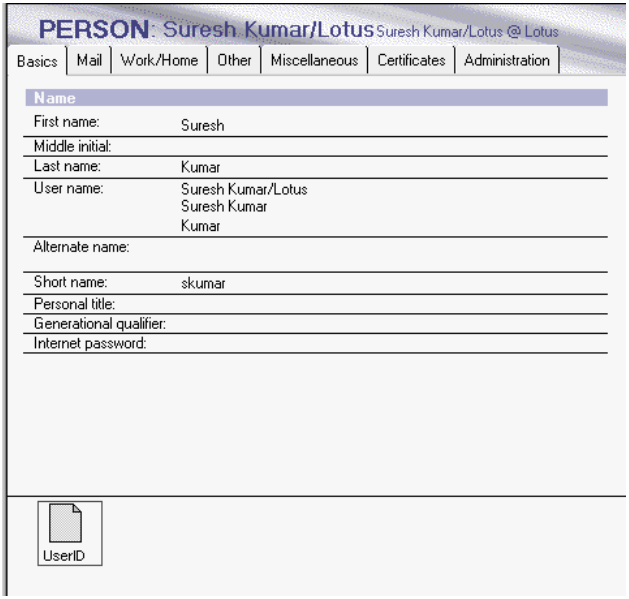
Field	*Enter*
Enable all synchronization operations	To enable all Notes synchronization operations listed under the "Select synchronization operations to enable" field. Whenever you perform one of the synchronization operations in User Manager for Domains, you are prompted to decide whether or not to perform the same operation in Notes.
Select synchronization operations to enable	Choose one of these to enable and disable selected Notes synchronization operations: • User/Group registration to register new or existing Windows NT users and groups in Notes. This option enables the Add Selected NT User/Group to Notes, Registration Setup, and Mail/ID Registration Options on the Notes menu. • User/Group deletion to delete a user or group from Windows NT and have that user or group deleted from the Domino Directory. Enables the "Delete/User Synch Options" command on the Notes menu. • User synching to change a user account name in User Manager and duplicate that name change in the Network account name field of the Person document in the Domino Directory, allow changes to the user's full name and copy the new name to the "User name" field in the Person document, enable the Notes menu command "Synch Selected NT Users with Notes," and activate the "Set common password on user synching" field.
Set common password on user synching	To synchronize the Windows NT password and the Notes Internet password when you synchronize users (available only if you selected user synching).
Prompt to confirm/cancel synchronization operations	Choose one: • Prompt for all operations (default) • Prompt only for user/group deletions • Do NOT prompt for any operations
Name format for full name parsing	Choose the parsing format that is most compatible with the name format of the Windows NT domain list. Full-name parsing is used to parse Windows NT full names into Notes name components. The default is "first name last name."

3. To save and reapply the settings in the next User Manager session, choose Options - Save Settings when you exit this session.

Resulting Person Record

The output of the NT registration process will be a person record created for the Domino directory. The information will be taken from the NT directory; Notes-specific information will be added according to the options specified.

A sample Domino person record looks like this:



Integrating Domino and OS/400 to Provide Single Sign On for Domino Users

You can achieve password synchronization between the client operating system, Notes, and the OS/400 system.

This allows password changes at the client operating system and/or Notes to be reflected in the client operating system, Lotus Notes, and OS/400 system.

The requirements for this to work are the following:

- Client operating system: Windows 95, Windows 98, or Windows NT
- Notes Client: Version 4.62 or Version 5
- Domino Server Version: Version 5
- OS/400: Client Access/400 V3R2M2 or later

More details and a step-by-step guide will be provided with the Domino for AS/400 Installation Instructions.

Summary

In this chapter you have learned about single sign on and how it is dependent on a common directory for authentication. We have explained the directory as an infrastructure service, and discussed how to set up the Domino LDAP service. Domino security integration with IIS, NT and AS/400 has also been described.

Appendix A
The Future: PKIX

Customers have expressed their need for reliable open, secure, interoperable, and maintainable security.

They would like to have a single point of administration to achieve easy and secure administration. They would like to invest only once in a turnkey solution, which would support all their existing and upcoming computing resources.

PKIX is the IETF Standard for Public Key Infrastructures for X.509. It currently meets the base requirements for a PKI infrastructure and received very strong support from such software development companies as:

- Netscape
- Verisign
- Microsoft
- Entrust

In 1998, IBM and Lotus developed a reference implementation of PKIX. This was donated as a software developer toolkit to the industry and is currently being reused in a broad range of new and upcoming products.

The PKIX software library will be used in future versions of Lotus Notes and Domino. IBM is currently implementing parts of the library to extend existing products such as IBM keywords and Global Sign On to conform to this standard.

Future versions of the AS/400 and OS/390 operating systems will benefit from the reuse of PKIX source code elements to become more open platforms.

The PKIX reference code is currently hosted at MIT, the URL is http://www.mit.edu/pfl but the download is currently restricted to US citizens only.

For more information on PKIX you may subscribe to a mailing list hosted by the Internet Mail Consortium (imc_pfl[-request]@imc.org).

Appendix B
LDAP Schema Used by Domino

Schema derived from	*Attribute returned by Domino*	*Field in Person document*	*Label displayed in Person document*	*Section of Person document label appears in*
Domino	AltFullName	AltFullName	Alternate Name	Name
Domino	AltFullNameLanguage	AltFullNameLanguage	-	-
x.500	secretary	Assistant	Assistant	Work
Domino	AvailableForDirSync	AvailableForDirSync	-	-
Domino	CalendarDomain	CalendarDomain	-	-
Domino	ccMailDomain	ccMailDomain	-	-
Domino	ccMailUserName	ccMailUserName	-	-
Domino	ccMailLocation	ccMailLocation	-	-
LDAPv3	mobileTelephoneNumber	CellPhoneNumber	Cell phone	Work
Domino	Certificate	Certificate	Certified public key	Public Keys
Domino	CheckPassword	CheckPassword	-	-
Domino	Children	Children	Children	Home
x.500	City	City	City	Home
Domino	ClientType	ClientType	-	-
LIPS	Description	Comment	Comment	Misc
Domino	CompanyName	CompanyName	Company Name	Work
Domino	Country	Country	Country	Home
Domino	Department	Department	Department	Work
Domino	DocumentAccess	DocumentAccess	-	-
Domino	EncryptIncoming Mail	EncryptIncoming Mail	Encrypt Incoming Mail	Misc
LDAPv3/LIPS/x.500	givenName	FirstName	First name	Name
x.500	ObjectClass	Form	-	-
Domino	FullName	FullName	User name	Name
x.500	homefax	HomeFAX PhoneNumber	FAX phone	Home

continued

Schema derived from	*Attribute returned by Domino*	*Field in Person document*	*Label displayed in Person document*	*Section of Person document label appears in*
Domino	HTTPPassword	HTTPPassword	Internet Password	Name
Domino	InternetAddress	InternetAddress	Internet Address	Mail
LIPS	Title	JobTitle	Title	Work
LIPS	SN	LastName	Last name	Name
Domino	LocalAdmin	LocalAdmin	Administrator	Administration
LDAPv3	LocalityName	Location	Location	Work
LIPS	mail	MailAddress	Shortname & domain.com or Forwarding address/Internet address	Mail
Domino	MailDomain	MailDomain	Domain	Mail
Domino	MailFile	MailFile	Mail file	Mail
Domino	MailServer	Mailserver	Mail server	Mail
Domino	MailSystem	MailSystem	Mail System	Mail
x.500	manager	Manager	Manager	Work
Domino 4.6x	MessageStore	MessageStore	-	-
Domino R5	MessageStorage	MessageStorage	-	-
LIPS	MiddleInitial	MiddleInitial	Middle initial	Name
Domino	NetUserName	NetUserName	-	-
LIPS	L	OfficeCity	City	Company Information
Domino	C	OfficeCountry	Country	Company Information
x.500	facsimileTelephone Number	OfficeFAXPhone Number	FAX phone	Work
LIPS	PhysicalDeliveryOffice Name	OfficeNumber	Office number	Company Information
x.500	telephoneNumber	OfficePhoneNumber	Office phone	Work
x.500	State	OfficeState	State/Province	Company Information
x.500	street	OfficeStreetAddress	Street address	Company Information

continued

Schema derived from	*Attribute returned by Domino*	*Field in Person document*	*Label displayed in Person document*	*Section of Person document label appears in*
LIPS	postalcode	OfficeZIP	Zip/postal code	Company Information
IWPS	creatorName	Owner	Owners	Administration
Domino	PasswordChangeDate	PasswordChangeDate	-	-
Domino	PasswordChange Interval	PasswordChange Interval	-	-
Domino	PasswordDigest	PasswordDigest	-	-
Domino	PasswordGracePeriod	PasswordGracePeriod	-	-
LDAPv2	homephone	PhoneNumber	Home phone	Home
LDAPv3	PagerTelephoneNumber	PhoneNumber_6	Pager number	Work
Domino	Profiles	Profiles	-	-
Domino	PublicKey	PublicKey	Public key	Public Keys
Domino	ShortName	ShortName	ShortName and/or Internet address for R4.x SMTP MTA	Name
Domino	ShortName	ShortName	Short name	
Domino	spouse	Spouse	Spouse	Home
x.500	State	State	State/province	Home
x.500	streetaddress	StreetAddress	Street address	Home
LIPS	GenerationQualifier	Suffix	Generational qualifier	Name
x.500	PersonalTitle	Title	Personal title	Name
Domino	Type	Type	-	-
x.500	UserCertificate	userCertificate	X.509 certificate	Public Keys
LDAPv3	Url	WebSiteWebSite	Web page	Misc
LIPS	textEncodedORaddress	x400Address	Other X.400 address	Misc
LDAPv2	HomeZip	Zip	Zip/postal code	Home
IWPS	OfficePager	PhoneNumber_6	Pager number	Work
LDAPv2	CommonName	derived from FullName field	User name	Name
LDAPv2	officefax	OfficeFAXPhone Number	FAX phone	Work

continued

Schema derived from	*Attribute returned by Domino*	*Field in Person document*	*Label displayed in Person document*	*Section of Person document label appears in*
LDAPv2	HomePostalAddress	returned by concatenating the following five fields, each separated by a $: StreetAddress City State country Zip	Street address City State/province Zip/postal code Country	Home
LDAPv2	DN	derived from FullName field	User name	Name
LDAPv2	ufn	derived from FullName field	User name	Name
LDAPv3	(Not Hierarchical)	-	-	-
LDAPv3	MHSORaddress	x400Address	Other X.400 address	Misc
LIPS	CN	derived from FullName field	User name	Name
LIPS	Initials	MiddleInitial	Middle initial	Name
LIPS	middleName	-	-	-
LIPS	O	derived from FullName field	User name	Name
LIPS	PostalAddress	returned by concatenating the following five fields, each separated by a $: OfficeStreetAddress OfficeCity OfficeState OfficeCountry OfficeZIP	Street address City State/province Zip/postal code Country	Company Information
LIPS	OU	derived from FullName field	User name	Name
x.500	mobile	CellPhoneNumber	Cell phone	Work
x.500	Info	Comment	Comment	Misc
x.500	S	LastName	Last name	Name
x.500	Surname	LastName	Last name	Name

continued

Schema derived from	*Attribute returned by Domino*	*Field in Person document*	*Label displayed in Person document*	*Section of Person document label appears in*
x.500	rfc822MailBox	MailAddress	Forwarding address/Internet address	Mail
x.500	RoomNumber	OfficeNumber	Office number	Company Information
X.500	homePhone	PhoneNumber	Home phone	Home

Schema derived from	*Attribute returned by Domino*	*Field in Group document*	*Label displayed in Group document*	*Section of Group document label appears in*
Domino	ListDescription	ListDescription	Description	Basics
Domino	ListName	ListName	Group name	Basics
x.500	member	Members	Members	Basics

Special Notices

This publication is intended to help you use Lotus Domino Release 5.0.

The information in this publication is not intended as the specification of any programming interfaces that are provided by Lotus Domino. See the publications section of the announcement for Lotus Domino and related products 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 products, programs, or services may be used. Any functionally equivalent program that does not infringe on any IBM 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, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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 information about non-IBM ("vendor") products in this manual has been supplied by the vendors, and IBM assumes no responsibility for its accuracy or completeness. 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.

Any performance data contained in this document was determined in a controlled environment, and therefore the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF, when it becomes available to each customer according to the normal IBM PTF distribution process.

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

AIX	NetView
AS/400	Network Station
BookManager	OS/2
BookMaster	OS/400
DB2	OS/390
DB2 Universal Database	VisualAge
IBM®	Visual Beans
MQSeries	VisualGen

The following are trademarks of Lotus Development Corporation in the United States and/or other countries:

1-2-3®	LotusScript®
Approach®	Lotus SmartSuite®
cc:Mail	Notes HiTest
DataLens®	Notes ViP®
Freelance®	Notes Mail®
InterNotes	NotesPump
InterNotes Web Publisher	NotesSQL

Lotus®	Notes/FX
Lotus Domino	Phone Notes®
Lotus Notes Reporter	Phone Notes Mobile Mail
Lotus Notes®	SmartIcons®
Lotus QuickPlace	Video Notes
	Word Pro

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

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 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. (For a complete list of Intel trademarks see www.intel.com/dradmarx.htm)

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

International Technical Support Organization Publications

For information on ordering these ITSO publications see, "How To Get ITSO Redbooks."

- *Lotus Notes 5.0: A Developers Handbook*, IBM form number SG24-5331, Lotus part number CT6HPIE
- *Enterprise-Wide Security Architecture and Solutions,* IBM form number SG24-4579
- *Protect and Survive, Using IBM Firewall 3.1 for AIX*, IBM form number SG24-2577
- *TCP/IP Tutorial and Technical Overview*, IBM form number GG24-3376
- *Understanding LDAP*, IBM form number SG24-4986

Other Lotus-Related ITSO Publications

The publications listed in this section may also be of interest:

- *Lotus Solutions for the Enterprise, Volume 1. Lotus Notes: An Enterprise Application Platform*, IBM form number SG24-4837, Lotus part number 12968
- *Lotus Solutions for the Enterprise, Volume 2. Using DB2 in a Domino Environment*, IBM form number SG24-4918, Lotus part number CT69BNA
- *Lotus Solutions for the Enterprise, Volume 3. Using the IBM CICS Gateway for Lotus Notes*, IBM form number SG24-4512
- *Lotus Solutions for the Enterprise, Volume 4. Lotus Notes and the MQSeries Enterprise Integrator*, IBM form number SG24-2217, Lotus part number 12992
- *Lotus Solutions for the Enterprise, Volume 5. NotesPump, the Enterprise Data Mover*, IBM form number SG24-5255, Lotus part number CT69DNA

- *Eight Steps to a Successful Messaging Migration:A Planning Guide for Migrating to Lotus Notes and Domino*, IBM form number SG24-5335, Lotus part number CT6HINA
- *Lotus Notes 4.5: A Developers Handbook*, IBM form number SG24-4876, Lotus part number AA0425
- *LotusScript for Visual Basic Programmers*, IBM form number SG24-4856, Lotus part number 12498
- *Secrets to Running Lotus Notes: The Decisions No One Tells You How to Make,* IBM form number SG24-4875, Lotus part number AA0424
- *Deploying Domino in an S/390 Environment*, IBM form number SG24-2182, Lotus part number 12957
- *Developing Web Applications Using Lotus Notes Designer for Domino 4.6*, IBM form number SG24-2183, Lotus part number 12974
- *The Next Step in Messaging: Case Studies on Lotus cc:Mail to Lotus Domino and Lotus Notes*, IBM form number SG24-5100, Lotus part number 12992
- *Lotus Notes and Domino: The Next Generation in Messaging. Moving from Microsoft Exchange to Lotus Notes and Domino*, IBM form number SG24-5167, Lotus part number CT7NLNA
- *Lotus Notes and Domino: The Next Generation in Messaging. Moving from Microsoft Mail to Lotus Notes and Domino*, IBM form number SG24-5152, Lotus part number CT7NJNA
- *Lotus Notes and Domino: The Next Generation in Messaging. Moving from Novell GroupWise to Lotus Notes and Domino*, IBM form number SG24-5321, Lotus part number CT7NNNA
- *High Availability and Scalability with Domino Clustering and Partitioning on Windows NT*, IBM form number SG24-5141, Lotus part number CT6XMIE
- *From Client/Server to Network Computing, A Migration to Domino*, IBM form number SG24-5087, Lotus part number CT699NA
- *Lotus Domino Integration Guide for IBM Netfinity and IBM PC Servers*, IBM form number SG24-2102
- *Lotus Domino Release 4.6 on IBM RS/6000: Installation, Customization and Administration*, IBM form number SG24-4694, Lotus part number 12969
- *High Availability and Scalability with Domino Clustering and Partitioning on AIX*, IBM form number SG24-5163, Lotus part number CT7J0NA
- *AS/400 Electronic-Mail Capabilities,* IBM form number SG24-4703

- *Mail Integration for Lotus Notes 4.5 on the IBM Integrated PC Server for AS/400*, IBM form number SG24-4977
- *Using Lotus Notes on the IBM Integrated PC Server for AS/400*, IBM form number SG24-4779
- *Lotus Domino for AS/400: Installation, Customization and Administration*, IBM form number SG24-5181, Lotus part number AA0964
- *Lotus Domino for S/390 Release 4.5: Installation, Customization & Administration*, IBM form number SG24-2083, Lotus part number AA0963
- *Lotus Domino for S/390 Performance Tuning and Capacity Planning*, IBM form number SG24-5149, Lotus part number CT6XNIE
- *Porting C Applications to Lotus Domino on S/390*, IBM form number SG24-2092, Lotus part number AB1720
- *Enterprise Integration with Domino for S/390,* IBM form number SG24-5150
- *Managing Domino/Notes with Tivoli Manager for Domino, Enterprise Edition, Version 1.5*, IBM form number SG24-2104
- *Measuring Lotus Notes Response Times with Tivoli's ARM Agents*, IBM form number SG24-4787, Lotus part number CT6UKIE
- *Image and Workflow Library: Integrating IBM FlowMark with Lotus Notes*, IBM form number SG24-4851
- *Implementing LAN Server for MVS in a Lotus Notes Environment*, IBM form number SG24-4741
- *Using ADSM to Back Up Lotus Notes*, IBM form number SG24-4534
- *NetFinity V5.0 Database Support*, IBM form number SG24-4808
- *An Approach to ODBC: Lotus Approach to DB2*, IBM form number SG24-4685

Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs:

CD-ROM Title	*Collection Kit Number*
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
Application Development Redbooks Collection	SK2T-8037
RS/6000 Redbooks Collection (PostScript)	SK2T-8041
RS/6000 Redbooks Collection (HTML, BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF)	SK2T-8043
AS/400 Redbooks Collection	SK2T-2849
Transaction Processing and Data Management Redbook Collection	SK2T-8038
Networking and Systems Management Redbooks Collection	SK2T-6022
System/390 Redbooks Collection	SK2T-2177

Other Publications

These publications and Web sites are also relevant as further information sources:

- *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage* by Clifford Stoll, Mass Market Paperback Reprint edition; July 1995; Pocket Books; ISBN: 0671726889.
- *An Introduction to Computer Security: The NIST Handbook*, Special publication 800-12, available online at http://www-08.nist.gov/nistpubs/800-12/
- *The Computer Security Act of 1987*, Public Law 100-235 (H.R. 145) January 8, 1988. Available online at http://www.epic.org/crypto/csa/default.html
- *Fourth Edition of the RSA FAQ*. Available online at http://www.rsa.com/rsalabs/faq/
- *Lotus Notes Network Design*, by John P. Lamb, Peter W. Lew, McGraw-Hill, September 1997, ISBN 0-07-036160-6; IBM order number SR23-7378

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site http://www.redbooks.ibm.com**
  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 via e-mail including information from the redbooks fax order form to:

	e-mail address
In United States:	usib6fpl@ibmmail.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 (toll free)	1-800-267-4455
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 for customers may be found at http://www.redbooks.ibm.com/ and for IBM employees at http://w3.itso.ibm.com/.

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 also view redbook, residency, and workshop announcements at http://inews.ibm.com/.

IBM Redbook 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, MasterCard, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Symbols

Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

ITSO Redbook Evaluation

Lotus Notes and Domino R5.0 Security Infrastructure Revealed SG24-5341-00

Your feedback is very important to help us maintain the quality of ITSO redbooks.

Please complete this questionnaire and return it using one of the following methods:

- Use the online evaluation form at `http://www.redbooks.ibm.com`
- Fax it to: USA International Access Code +1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

___Customer ___Business Partner ___Solution Developer ___IBM employee

___None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes _____ No _____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

Printed in the U.S.A.

SG24-5341-00

Part No. CT6TPNA