IBM

IBM Host Integration in a Secure Network: A Practical Approach

- IBM WebSphere Host On-Demand and Host Publisher
- Integration into a Notes Domino environment
- Other common secure scenarios

George Baker
Jason Beere
Bob Bogardus
Sitisak Jongvattanasiri
Iwan Seeldrayers

Redbooks

ibm.com/redbooks

SG24-5988-00

International Technical Support Organization

IBM Host Integration in a Secure Network: A Practical Approach

July 2001

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix E, “Special notices” on page 263.

First Edition (July 2001)

This edition applies to WebSphere Host On-Demand Version 5.04 and WebSphere Host Publisher Version 2.2.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195

Contents

Preface

By their very nature, e-business applications are more vulnerable to attack than any type of previous I/T system. Security needs to be at the heart of these systems because the costs of getting it wrong can be enormous. The publicity associated with a breach of security reduces customer, supplier and business partner confidence in the enterprise and the share price plummets. There also may be actual monetary theft, or confidential information might be compromised. Whatever the details, the result is a loss to the business, and in the extreme case, the loss of the business itself.

Appropriate security needs to be in place for all systems - whether they are simply company Web pages with no access to the company intranet at all, or full-blown customer-to-business or business-to-business systems with external customers or partners having access to company systems. Of course, the type and extent of the security system will be different in each case.

This redbook is directed at I/T managers and architects planning for, or implementing, an IBM host integration solution. It explores aspects of network security when implementing an IBM Web-to-host integration solution, consisting of IBM WebSphere Host Publisher and IBM WebSphere Host On-Demand. It describes the native security capabilities of each product and how they may be used by themselves and with other network security products to provide 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, Raleigh Center.

George Baker is a Senior I/T Specialist at the International Technical Support Organization, Raleigh Center. He writes and teaches IBM classes worldwide on host integration software. Before joining the ITSO in 2000, George worked for over 30 years in the field as a programmer, technical specialist, sales specialist and manager in the areas of large systems, networking and workstation software systems.

Jason Beere is an I/T Specialist with IBM in the United States. He has five years of experience with IBM in the fields of networking software and Lotus Notes. His areas of expertise include Lotus Notes and Domino, Windows NT, TCP/IP, Host On-Demand, Host Publisher, Communications Server, Personal Communications Manager, and OS/2.

Bob Bogardus is a Software Engineer for the IBM Software Group at RTP. He holds a degree in Computer Science from SUNY Potsdam. He has 17 years of experience in a variety of areas related to AS/400 systems management, logistics/finance/manufacturing solutions, networking and I/T security. In 1999, Bob was part of the team that ported Host On-Demand and Screen Customizer to the AS/400.

Sitisak Jongvattanasiri is an I/T Specialist for IBM Global Services in Thailand. He has four years of experience in a variety of areas related to host integration and Windows NT. He holds a degree in Computer Engineering from Kasetsart University. He also holds an MCSE certificate.

Iwan Seeldrayers is an I/T Specialist for IBM Global Services in Belgium. He has three years' experience in host integration and networking on PC and OS/390 platforms. He holds a Civil Engineering degree from the University of Ghent. His areas of expertise include Host Publisher and Host On-Demand.

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

Byron Braswell
Margaret Ticknor
Thomas Barlen
Juan Rodriguez
International Technical Support Organization Raleigh, NC

Axel Buecker
International Technical Support Organization Austin, TX

Bryan Aupperle
Bobby Barnes
Henry Mok
Matthew Sheard

Byron Williams
IBM Research Triangle Park, NC

Jorge Ferrari
Jim Williams
Tivoli, Research Triangle Park, NC

Juan Manuel Martínez
IBM Spain

Comments welcome

Your comments are important to us!

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

- Fax the evaluation form found in “IBM Redbooks review” on page 281 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Introduction to host integration and security

The Web and its associated technologies offer you virtually unlimited ways to extend the reach of your business information. As a result, providing access to information stored on IBM S/390, IBM AS/400 and other back-end systems, such as UNIX, Microsoft Windows NT and Microsoft Windows 2000 system-based servers, is more important than ever. Your critical business information and applications most likely reside on host systems like these. The quantity and quality of your business information, combined with the reach of the Web, afford you a unique opportunity to transform that information into a powerful competitive advantage.

Merging Web technology with your existing information systems defines e-business, and building an e-business means finding a solution that provides a fast and cost-effective way to access, integrate and publish host information to Web-based clients.

The Internet gives you an opportunity to extend the reach of your business to your employees, business partners or new customers around the world. However, you may have questions that make you reluctant to take advantage of the Internet's potential. The most talked-about concern over deploying applications over the Internet is security. This redbook addresses these concerns.

The remainder of this chapter introduces the reader to the basics of security. Part 1 of the book provides a chapter on the overview of some of the more common security techniques available with TCP/IP, followed by chapters that cover the specifics of the security features and functions of IBM WebSphere Host On-Demand and IBM WebSphere Host Publisher. Part 2 of the book illustrates the implementation of these security features and function in the following scenarios:

- Security using a reverse proxy
- Security using a DMZ
- Security using a VPN
- Security using Lotus Domino

1.1 Security policy introduction

Network security is implemented to protect two objects:

- The data that is transmitted on the network
- The computers that are connected to the network

Network security cannot replace physical site security, host security on the connected systems, application security, and user security education. It can only act as a first line of defense.

You should always implement security in layers, assuming that if an error or an attack in one layer opens a hole, there will be a second layer of defense that protects the heart of your assets.

The goals and basic concepts of network security are similar to other aspects of security in computer room systems. The main difference is that network security mainly deals with data that is transmitted, remote parties, and networks that are public (Internets) and are more vulnerable to attacks, or networks that belong to another entity (extranet).

Network security is usually the first line of defense for securing your host systems. The network is replacing the physical doors of entry into your organization. Attackers from outside your organization must break through either your network or your physical security before they can attempt to break into your host system.

The IBM security architecture, described in *Enterprise-Wide Security Architecture and Solutions*, SG24-4579, defines the following security services:

- **Access control**

  This is the first thing that most of us think of when considering security. Access control limits access to the facilities of a system to authorized users. Access control is implemented on the target system and is always implemented in conjunction with the identification and authentication of the user.

- **Identification and authentication**

  Identification and Authentication guarantees the identity of users and components of a system. Signed applet support, described in 3.1, “Signed applet support” on page 47, provides identification of Java applets, while SSL, described in 2.4, “Secure Sockets Layer” on page 38, supports client and server authentication.

- **Confidentiality**

  Confidentiality ensures that data cannot be accessed by unauthorized individuals. SSL is the facility most often used to provide this capability.

- **Data integrity**

  Data integrity protects against the unauthorized modification of data. This is another capability provided by SSL.

- **Non-repudiation**

  This service prevents the partners to a transaction from denying that data was sent or received. This is in many ways an extension to the identification and authentication and data integrity services. The technique of using digital signatures can be used to implement non-repudiation. On the information highway, the digital signature replaces the handwritten signature as a legal proof of authenticity. It may be used to provide proof of submission and proof of transport. A digital signature is used directly to provide proof of whom the data sender was. Most Web servers today support digital certificates, as does Host On-Demand and Host Publisher.

If you are concerned about information technology security in a wider context, you will find additional valuable information at:

http://www.ibm.com/Security

1.2 Defining a security policy

It is important to adopt a systematic approach to security. One of the major concerns that emerges with security solutions is the overall complexity and cost. The following inhibitors to deploying security solutions are frequently mentioned:

- Security is too complex.
- Security policies are becoming impossible to implement.
- The total cost of security is escalating.
- Security topics are stopping e-business initiatives.

Before considering the details of a solution, this section briefly introduces the development of enterprise policy definitions. These policy definitions are the result of comprehensive analysis and risk assessment in the context of the organization's business function, where many technical, business, and legal issues must be considered during this process.

Typically, the process of security policy definition follows these stages: assessment and planning, architecture and design, and, finally, implementation. The following discussion concentrates on generic, Internet-related assessments and planning tasks. All the techniques listed would not necessarily apply in any specific situation, although there should be at least one technique that is useful for any given situation. It is not the intent of this chapter to serve as a guide for performing these activities. If the expertise is not readily available within an organization, the assessment and planning activities should be performed by a security expert. In many cases,

this may be a consultant hired to oversee the design and implementation of a comprehensive enterprise security system.

1.3 Typical threats to security

The following are typical techniques/methods used to gain access to your systems. Additional information can be found in *Hacking Exposed Network Security Secrets & Solutions*, by Stuart McClure and Joel Scambray.

- **Scanning**

  Scanning is similar to the thief who walks through a parking lot looking for cars that have unlocked doors. Several tools exist for attackers to scan for available services. The typical countermeasure is to turn off all unnecessary protocols (a process called hardening) and disallow all anonymous services.

- **Trojan horse**

  If the thief is able to detect your address, one of the typical next moves is to see if he can get someone to install a service that he can control. The typical countermeasure is a firewall on the server side and a personal firewall on the client desktop. It is recommended that a client use some type of firewall between the client and the Internet connection.

- **Sniffing**

  Protocol analyzers are legitimate tools that are used to debug network problems. However, they are also used by hackers. The attacker watches individual packets of information as they are being sent on the network. If information is not encrypted, it is easy to view user ID/password information. The same technique could be used to eavesdrop on a conversation. The source and destination IP addresses may also be visible if you are not using a proxy or redirector. This is a problem that can occur at just about any point in your network. A typical countermeasure is to use end-to-end encryption.

- **Impersonation**

  The attacker tricks your security system by passing as an authorized user. This could happen if someone steals a user ID and password. A typical countermeasure is to send users an activity report to their e-mail addresses when information has been modified.

- **Grinding**

  The attacker can use a password guessing program that tries passwords until a combination works. This process can be slowed down or detected

by disabling a user profile after a certain number of invalid passwords are attempted.

- **Denial of Service**

  A denial-of-service attack is flooding the network or server with very large numbers of requests so as to affect the availability of services on the network or server. The network or server will be so congested with the bogus requests that it will not be able to respond to legitimate requests, or the response time to legitimate requests will be severely impacted.

- **Technology weakness**

  Some TCP/IP protocols and some operating systems that incorporate TCP/IP functionality have inherent security shortcomings. For example, the UNIX sendmail program, SNMP (Simple Network Management Protocol), SNMP (Simple Mail Transfer Protocol), and Syn floods all have known security holes. A typical countermeasure is to turn off or uninstall any unneeded protocols. Other ideas include using a separate server for each service in order to avoid a single point of entry or attack.

1.4 Evaluating the threats

When you have identified the resources you need to protect and the threats to which they are exposed, you must evaluate the following:

- What would the gain be for the attacker?
- What would the damage be for us?
- How much will it impact the users?
- How much will it cost to protect against the threat?

The answers to the above questions should direct you to the level of security you will need to implement. One way to assess risk vs. prevention would be to assess a severity level to each of the above questions, then weigh the results. If, for example, the risk level to either of the first two questions were high, and the answer to the third question is also high, then you would have to seriously consider taking measures to plug that specific security exposure. Finally, the last question is always there: what will it cost? It must be considered in the overall decision.

The following quote from JavaSoft sums it up.

> “Every security policy is derived from balancing the value of your information assets against both the costs of protecting them and the costs of attacking them. Network managers know that no implementation of security is absolute. End users do not.”

Part 1. Host integration security

Chapter 2. TCP/IP security introduction

This chapter discusses the network security techniques available with TCP/IP, and provides an overview of a number of solutions for addressing security issues in networks.

The field of network security in general and of TCP/IP security in particular is very wide, so this chapter concentrates on the most recent and most widely used security techniques. The following topics are covered:

1. Basic concepts of cryptography and digital certificates
2. Firewall concepts
3. Virtual private network (VPN) and IPSec
4. Secure Sockets Layer (SSL)
5. Transport Layer Security (TLS)

For more details on the concepts covered in this chapter, please see *TCP/IP Tutorial and Technical Overview*, GG24-3376.

2.1 Basic concepts of cryptography and digital certificates

If you are sending data in the clear over a network that is not completely under your control from the receiver to the sender, you will be unable to ensure the following security functions:

- **Privacy**

  Anyone who is able to intercept your data might be able to read it.

- **Integrity**

  An intermediary might be able to alter your data.

- **Accountability or non-repudiation**

  It may be impossible to determine the originator of a message with confidence, and thus the person who sent the message could deny being the originator.

Security functions such as identification and authentication are also impacted because if authentication data such as passwords are sent without integrity and privacy, they can be intercepted in transit between sender and receiver, making the authentication compromised and worthless.

To ensure privacy, integrity and accountability in nonsecure networks, cryptographic procedures need to be used. Today, two distinct classes of encryption algorithms are in use: symmetric and asymmetric algorithms. They are fundamentally different in *how* they work, and thus in *where* they are used.

2.1.1 Symmetric encryption algorithms

An encryption algorithm is called symmetric because the same key that is used to encrypt the data is also used to decrypt the data and recover the clear text (see Figure 1). The cipher and decipher processes are usually mathematically complex nonlinear permutations.

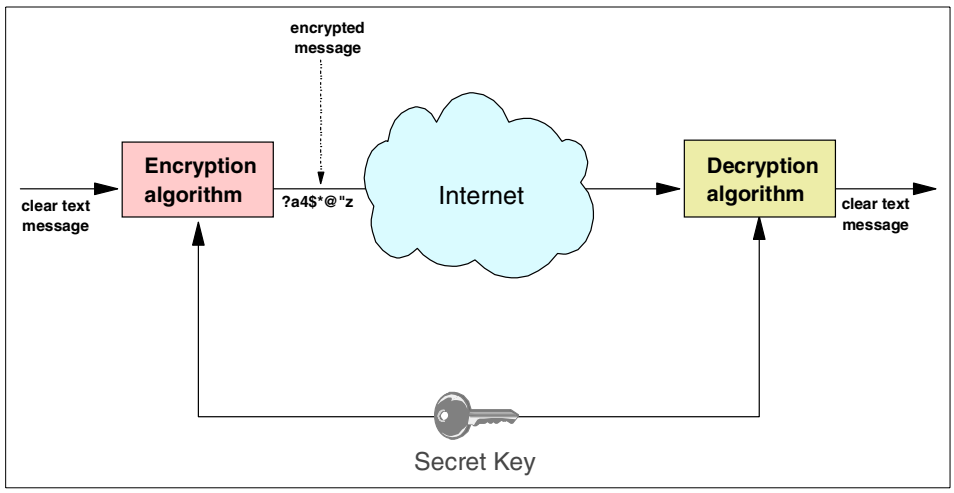

Figure 1. Symmetric encryption and decryption: using the same key

Symmetric algorithms are usually efficient in terms of processing power, so they are ideal for encryption of bulk data. However, they have one major drawback, which is key management. The sender and receiver on any secure connection must share the same key; in a large network where thousands of users may need to communicate securely, it is extremely difficult to manage the distribution of keys so as not to compromise the integrity of any one of them.

Frequently used symmetric algorithms include:

- **Data Encryption Standard (DES)**

  Developed in the 1970s by IBM scientists, DES uses a 56-bit key. Stronger versions called Triple DES have been developed that use three operations in sequence: “2-key Triple DES” encrypts with key 1, decrypts

with key 2, and encrypts again with key 1. The effective key length is 112 bits. “3-key Triple DES” encrypts with key 1, decrypts with key 2, and encrypts again with key 3. The effective key length is 168 bits.

- **Commercial Data Masking Facility (CDMF)**

  This is a version of the DES algorithm approved for use outside the U.S. and Canada (in times when export control was an issue). It uses 56-bit keys, but 16 bits of the key are known, so the effective key length is 40 bits.

- **RC2**

  Developed by Ron Rivest for RSA Data Security, Inc., RC2 is a block cipher with variable key lengths operating on 8-byte blocks. Key lengths of 40, 56, 64, and 128 bits are in use.

- **RC4**

  Developed by Ron Rivest for RSA Data Security, Inc., RC4 is a stream cipher operating on a bit stream. Key lengths of 40 bits, 56 bits, 64 bits, and 128 bits are in use. The RC4 algorithm always uses 128-bit keys; the shorter key lengths are achieved by “salting” the key with a known, non-secret random string.

- **Advanced Encryption Standard (AES)**

  As a result of a contest for a follow-on standard to DES held by the National Institute for Standards and Technology (NIST), the Rijndael algorithm was selected. This is a block cipher created by Joan Daemen and Vincent Rijmen with variable block length (up to 256 bits) and variable key length (up to 256 bits).

- **The International Data Encryption Algorithm (IDEA)**

  IDEA was developed by James Massey and Xueija Lai at ETH in Zurich. It uses a 128-bit key and is faster than triple DES.

DES is probably the most scrutinized encryption algorithm in the world. Much work has been done to find ways to break DES, notably by Biham and Shamir, but also by others. However, a way to break DES with appreciably less effort than a brute-force attack (breaking the cipher by trying every possible key) has not been found.

Both RC2 and RC4 are proprietary, confidential algorithms that have never been published. They have been examined by a number of scientists under non-disclosure agreements.

With all the ciphers listed above, it can be assumed that a brute-force attack is the only means of breaking the cipher. Therefore, the work factor depends

on the length of the key. If the key length is n bits, the work factor is proportional to 2**(n-1).

Today, a key length of 56 bits is generally only seen as sufficiently secure for applications that do not involve significant amounts of money or critically secret data. If specialized hardware is built (such as the machine built by John Gilmore and Paul Kocher for the Electronic Frontier Foundation), the time needed for a brute-force attack can be reduced to about 100 hours or less (see: *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design,* by Electronic Frontier Foundation, John Gilmore (Editor), 1988). Key lengths of 112 bits and above are seen as unbreakable for many years to come, since the work factor rises exponentially with the size of the key.

2.1.2 Asymmetric encryption algorithms

Asymmetric encryption algorithms are so called because the key that is used to encrypt the data cannot be used to decrypt the data; a different key is needed to recover the clear text (see Figure 2). This key pair is called a public key and a private key. If the public key is used to encrypt the data, the private key must be used to recover the clear text. If data is encrypted with the private key, it can only be decrypted with the public key.

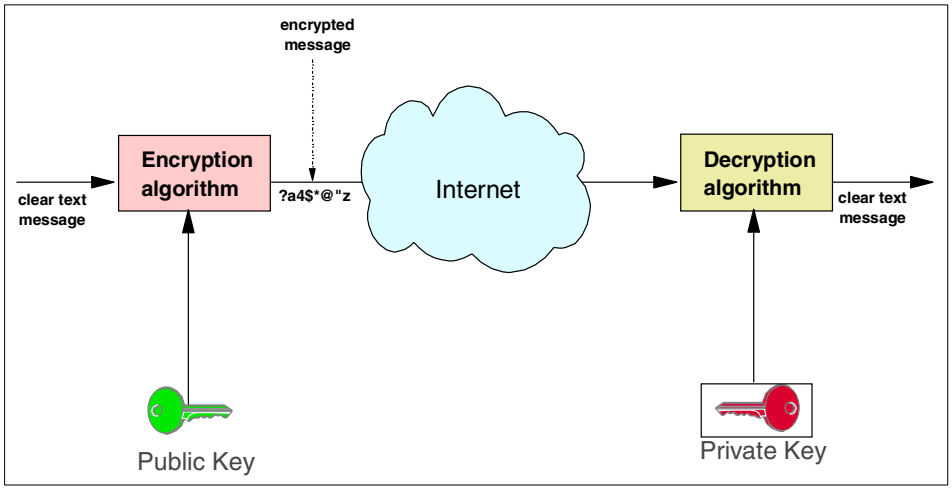

Figure 2. Public-key cryptography: using a key pair

Asymmetric encryption algorithms, commonly called Public Key Cryptography Standards (PKCS), are based on mathematical algorithms. The basic idea is to find a mathematical problem that is very hard to solve. The algorithm in most widespread use today is RSA. However, some companies have begun

to implement public-key cryptosystems based on elliptic curve algorithms. With the growing proliferation of IPSec, the Diffie-Hellman algorithm is gaining popularity. A brief overview of all three methods follows:

- **RSA**

  Invented 1977 by Rivest, Shamir, and Adleman (who formed RSA Data Security Inc.). The idea behind RSA is that integer factorization of very large numbers is extremely hard to do. Key lengths of public and private keys are typically 512 bits, 768 bits, 1024 bits, or 2048 bits. The work factor for RSA with respect to key length is sub-exponential, which means the effort does not rise exponentially with the number of key bits. It is roughly 2**(0.3*n).

- **Elliptic Curve**

  Public-key cryptosystems based on elliptic curves use a variation of the mathematical problem of finding discrete logarithms. It has been stated that an elliptic curve cryptosystem implemented over a 160-bit field has roughly the same resistance to attack as RSA with a 1024-bit key length. Properly chosen elliptic curve cryptosystems have an exponential work factor (which explains why the key length is so much smaller). Elliptic curve cryptosystems are now standardized by FIPS PUB 186-2, the digital signature standard (January 2000).

- **Diffie-Hellman**

  W. Diffie and M.E. Hellman, the inventors of public key cryptography, published this algorithm in 1976. The mathematical problem behind Diffie-Hellman is computing a discrete logarithm. Both parties have a public-private key pair each; they are collectively generating a key only known to them. Each party uses its own private key and the public key of the other party in the key generation process. Diffie-Hellman public keys are often called *shares*.

The beauty of asymmetric algorithms is that they are not subject to the key management issues that beset symmetric algorithms. Your public key is freely available to anyone, and if someone wants to send you a message he or she encrypts it using that key. Only you can understand the message, because only you have the private key. Asymmetric algorithms are also very useful for authentication. Anything that can be decrypted using your public key must have been encrypted using your private key, in other words, by you.

2.1.3 Performance issues of cryptosystems

Elliptic curve cryptosystems are said to have performance advantages over RSA in decryption and signing. While the possible differences in performance

between the asymmetric algorithms are somewhere in the range of a factor of 10, the performance differential between symmetric and asymmetric cryptosystems is far more dramatic.

For instance, it takes about 1000 times as long to encrypt the same data with RSA (an asymmetric algorithm) than with DES (a symmetric algorithm), and implementing both algorithms in hardware does not change the odds in favor of RSA.

As a consequence of these performance issues, the encryption of bulk data is usually performed using a symmetric cryptosystem, while asymmetric cryptosystems are used for electronic signatures and in the exchange of key material for secret-key cryptosystems. With these applications, only relatively small amounts of data need to be encrypted and decrypted, and the performance issues are less important.

2.1.4 Cryptosystems for data integrity

Data integrity is the ability to assert that the data received over a communication link is identical to the data sent. Data integrity in an insecure network requires the use of cryptographic procedures. However, it does not imply that only the receiver is able to read the data, as with data privacy. Data could be compromised not only by an attacker, but also by transmission errors (although those are normally handled by transmission protocols such as TCP).

2.1.4.1 Message digest algorithms

A message digesting algorithm (often also called a “digital hash”) is an algorithm that “digests” (condenses) a block of data into a shorter string (usually 128 or 160 bits), which is called a Message Digest, Secure Hash, or Message Integrity Code (MIC). See Figure 3 on page 15 for a graphical representation. The principle behind message digest algorithms is as follows:

- The message cannot be recovered from the message digest.

It is very hard to construct a block of data that has the same message digest as another given block.

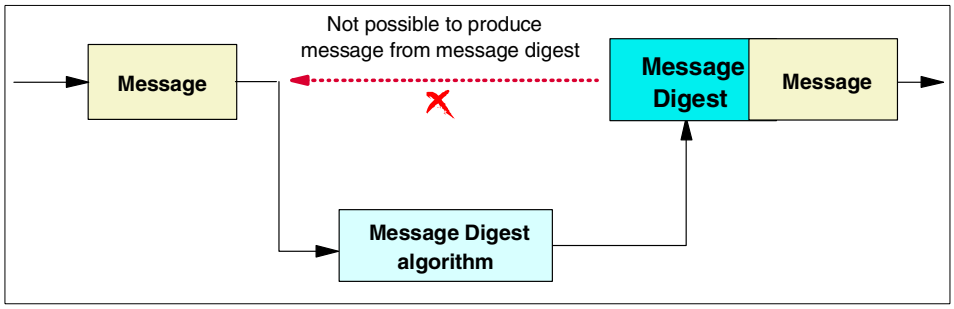

Figure 3. Message digest

Common message digest algorithms are:

- **MD2**

  Developed by Ron Rivest of RSA Data Security, Inc. The algorithm is mostly used for Privacy Enhanced Mail (PEM) certificates. MD2 is fully described in RFC 1319. Since weaknesses have been discovered in MD2, its use is discouraged.

- **MD5**

  Developed in 1991 by Ron Rivest. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest of the input. The MD5 message digest algorithm is specified in RFC 1321, *The MD5 Message-Digest Algorithm*. Collisions have been found in MD5; see *Cryptanalysis of MD5 Compress*, by Hans Dobbertin, available at `http://www.cs.ucsd.edu/users/bsy/dobbertin.ps`.

- **SHA-1**

  Developed by the National Security Agency (NSA) of the U.S. Government. The algorithm takes as input a message of arbitrary length and produces as output a 160-bit “hash” of the input. SHA-1 is fully described in standard FIPS PUB 180-1, also called the Secure Hash Standard (SHS). SHA-1 is generally recognized as the strongest and most secure message digesting algorithm.

- **SHA-256, SHA-512**

  Developed by the National Security Agency (NSA) of the U.S. Government. The security of a hash algorithm against collision attacks is half the hash size and this value should correspond with the key size of encryption algorithms used in applications together with the message digest. Since SHA-1 only provides 80 bits of security against collision attacks, this is deemed inappropriate for the key lengths of up to 256 bits

planned to be used with AES. Therefore, extensions to the Secure Hash Standard (SHS) have been developed. SHA-256 provides a hash size of 256 bits while SHA-512 provides a hash size of 512 bits.

2.1.4.2 Message digests for data integrity

The sender of a message (block of data) uses an algorithm, for example, SHA-1, to create a message digest from the message (see Figure 4). The message digest can be sent together with the message to provide data integrity. The receiver runs the same algorithm over the message and compares the resulting message digest to the one sent with the message. If both match, the message is unchanged.

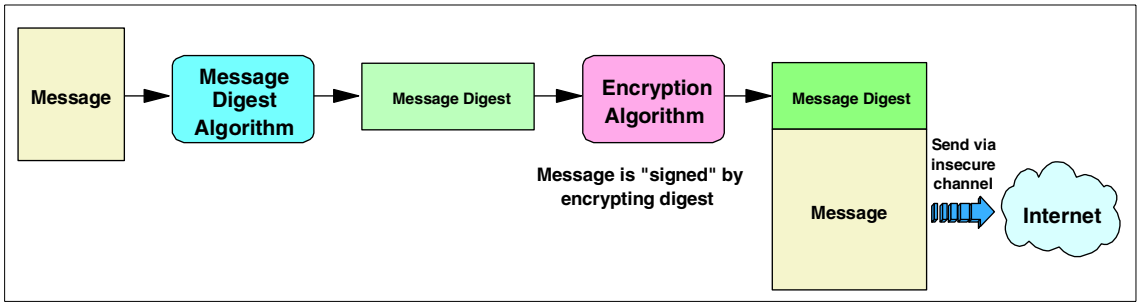

Figure 4. Message digest for data integrity

The message digest should not be sent in the clear: Since the digest algorithms are well-known and no key is involved, a man-in-the-middle could not only forge the message but also replace the message digest with that of the forged message. This would make it impossible for the receiver to detect the forgery. The solution for this is to encrypt the message digest, that is, to use a Message Authentication Code (MAC).

2.1.4.3 Message authentication codes

Secret-key cryptographic algorithms, such as DES, can be used for encryption with message digests. A disadvantage is that, as in secret-key cryptosystems, the keys must be shared by sender and receiver. Furthermore, since the receiver has the key that is used in MAC creation, this system does not offer a guarantee of non-repudiation. That is, it is theoretically possible for the receiver to forge a message and claim it was sent by the sender. Therefore, message authentication codes are usually based on public/private key encryption in order to provide for non-repudiation. This is discussed further in 2.1.5, “Digital signatures” on page 17.

2.1.4.4 Keyed hashing for message authentication (HMAC)

H. Krawczyk and R. Canetti of IBM Research and M. Bellare of UCSD invented a method to create a message authentication code called HMAC, which is defined in RFC 2104 as a proposed Internet standard. A simplified description of how to create the HMAC is as follows: The key and the data are concatenated and a message digest is created. The key and this message digest are again concatenated for better security, and another message digest is created, which is the HMAC.

HMAC can be used with any cryptographic hash function. Typically, either MD5 or SHA-1 are used. In the case of MD5, a key length of 128 bits is used (the block length of the hash algorithm). With SHA-1, 160-bit keys are used. Using HMAC actually improves the security of the underlying hash algorithm. For instance, some collisions (different texts that result in the same message digest) have been found in MD5. However, they cannot be exploited with HMAC. Therefore the weakness in MD5 does not affect the security of HMAC-MD5.

HMAC is now a PKCS#1 V.2 standard for RSA encryption (proposed by RSA Inc. after weaknesses were found in PKCS#1 applications). For further details, see `http://www.ietf.org/rfc.html`. HMAC is also used in the Transport Layer Security (TLS) Protocol, the successor to SSL.

2.1.4.5 Message authentication used with SSL

In the Secure Sockets Layer Protocol (SSL), a slightly different MAC algorithm has been implemented. The MAC write-secret and the sequence number of the message are concatenated with the data, and a message digest is created. The MAC write-secret and this message digest are again concatenated for better security, and another message digest is created, which is the MAC. Again, for the hash function, either MD5 or SHA-1 can be used. If compression is used, the text is compressed before the MAC is calculated. For further details, see:

`http://home.netscape.com/eng/ssl3/draft302.txt`

2.1.5 Digital signatures

Digital signatures are an extension to data integrity. While data integrity only ensures that the data received is identical to the data sent, digital signatures go a step further: they provide non-repudiation. This means that the sender of a message (or the signer of a document) cannot deny authorship, similar to signatures on paper. As illustrated in Figure 5 on page 18, the creator of a message or electronic document that is to be signed uses a message digesting algorithm such as MD5 or SHA-1 to create a message digest from

the data. The message digest and some information that identifies the sender are then encrypted with an asymmetric algorithm using the sender's private key. This encrypted information is sent together with the data.

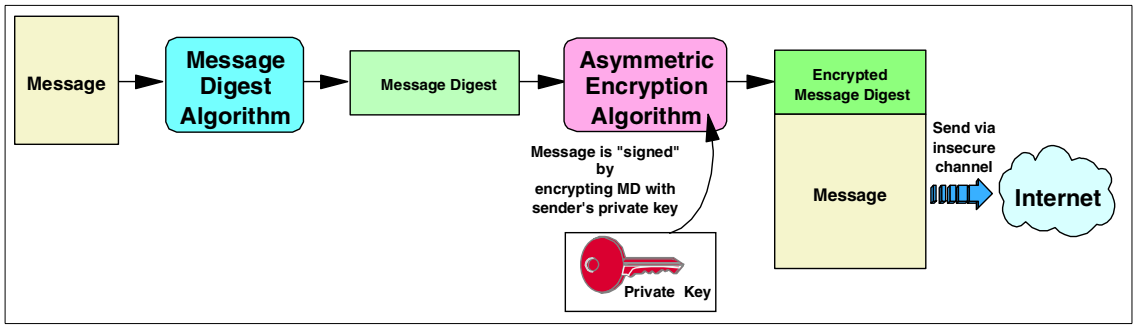

Figure 5. Digital signature creation

The receiver, as shown in Figure 6, uses the sender's public key to decrypt the message digest and identification of the sender. He or she will then use the message digesting algorithm to compute the message digest from the data. If this message digest is identical to the one recovered after decrypting the digital signature, the signature is recognized as valid proof of the authenticity of the message.

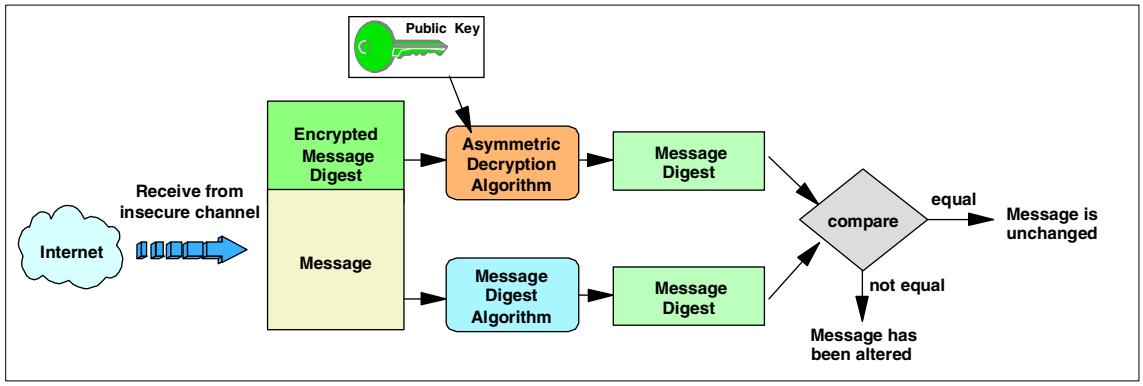

Figure 6. Digital signature verification

With digital signatures, only public-key cryptosystems can be used. If secret-key cryptosystems would be used to encrypt the signature, it would be very difficult to make sure that the receiver (having the key to decrypt the signature) could not misuse this key to forge a signature of the sender. The

private key of the sender is known to nobody else, so nobody is able to forge the sender's signature.

Note the difference between encryption using public-key cryptosystems and digital signatures:

- With encryption, the sender uses the receiver's public key to encrypt the data, and the receiver decrypts the data with his private key. This means everybody can send encrypted data to the receiver that only the receiver can decrypt. See Figure 7 for a graphical representation.

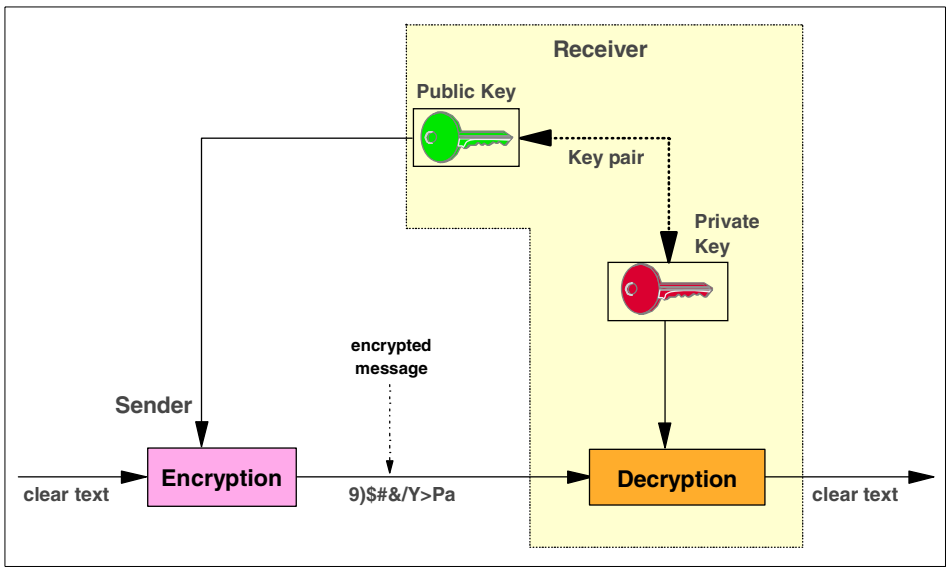

Figure 7. Encrypting data with the receiver's public key

- With digital signatures, the sender uses his private key to encrypt his signature, and the receiver decrypts the signature with the sender's public key. This means that only the sender can encrypt the signature, but everybody who receives the signature can decrypt and verify it.

The tricky part with digital signatures is the trustworthy distribution of public keys, since a genuine copy of the sender's public key is required by the receiver. A solution to this problem is provided by digital certificates, which are discussed next.

2.1.6 Public Key Infrastructure

A Public Key Infrastructure (PKI) offers the basis for practical usage of public key encryption. A PKI defines the rules and relationships for certificates and Certificate Authorities (CAs). It defines the fields that can or must be in a certificate, the requirements and constraints for a CA in issuing certificates, and how certificate revocation is handled.

PKI has been exploited in many applications or protocols, such as Secure Sockets Layer (SSL), Secure Multimedia Internet Mail Extensions (S/MIME), IP Security (IPSec), Secure Electronic Transactions (SET), and Pretty Good Privacy (PGP). PKI is described here, only insofar as its use with Web serving and Secure Sockets Layer (SSL) is concerned. For more information on PKI, see the redbook *Deploying a Public Key Infrastructure*, SG24-5512.

2.1.6.1 Digital certificates

When using a PKI, the user must be confident that the public key belongs to the correct remote person (or system) with which the digital signature mechanism is to be used. This confidence is obtained through the use of public key digital certificates. A digital certificate is analogous to a passport: the passport certifies the bearer's identity, address and citizenship. The concepts behind passports and other identification documents (for instance, drivers' licenses) are very similar to those that are used for digital certificates.

Passports are issued by a trusted authority, such as a government passport office. A passport will not be issued unless the person who requests it has proven their identity and citizenship to the authority. Specialized equipment is used in the creation of passports to make it very difficult to alter the information in it or to forge a passport altogether. Other authorities, for instance, the border police in other countries, can verify a passport's authenticity. If they trust the authority that issued the document, they implicitly trust the passport.

A digital certificate serves two purposes: it establishes the owner's identity and it makes the owner's public key available. Similar to a passport, a certificate must be issued by a trusted authority, the CA; and, like a passport, it is issued only for a limited time. When its expiration date has passed, it must be replaced.

Trust is a very important concept in passports, as well as in digital certificates. In the same way as, for instance, a passport issued by the governments of some countries, even if recognized to be authentic, will probably not be trusted by the US authorities, each organization or user has to determine whether a CA can be accepted as trustworthy.

For example, a company might want to issue digital certificates for its own employees from its own Certificate Authority; this could ensure that only authorized employees are issued certificates, as opposed to certificates being obtained from other sources such as a commercial entity such as VeriSign.

The information about the certificate owner's identity is stored in a format that follows RFC 2253 and the X.520 recommendation, for instance: CN=George Baker O=IBM Corporation; the complete information is called the owner's distinguished name (DN). The owner's distinguished name and public key and the CA's distinguished name are digitally signed by the CA; that is, a message digest is calculated from the distinguished names and the public key. This message digest is encrypted with the private key of the CA.

Figure 8 shows the layout of a digital certificate.

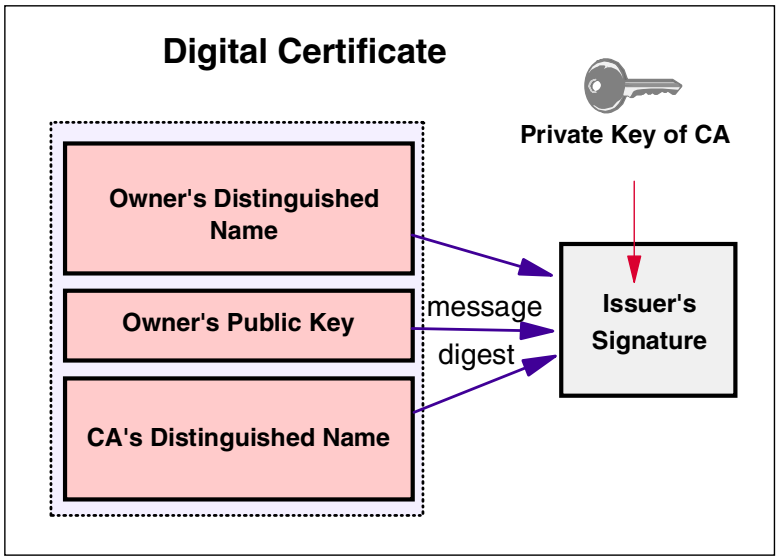

Figure 8. Simplified layout of a digital certificate

The digital signature of the CA serves the same purpose as the special measures taken for the security of passports such as laminating pages with plastic material: it allows others to verify the authenticity of the certificate. Using the public key of the CA, the message digest can be decrypted. The message digest can be recreated; if it is identical to the decrypted message digest, the certificate is authentic.

Security considerations for certificates

If I send my certificate with my public key in it to someone else, what keeps this person from misusing my certificate and posing as myself? The answer is: my private key.

A certificate alone can never be proof of anyone's identity. The certificate just allows the identity of the certificate owner to be verified by providing the public key that is needed to check the certificate owner's digital signature. Therefore, the certificate owner must protect the private key that matches the public key in the certificate. If the private key is stolen, the thief can pose as the legitimate owner of the certificate. Without the private key, a certificate cannot be misused.

An application that authenticates the owner of a certificate cannot accept just the certificate. A message signed by the certificate owner should accompany the certificate. This message should use elements such as sequence numbers, time stamps, challenge-response protocols, or other data that allow the authenticating application to verify that the message is a "fresh" signature from the certificate owner and not a replayed message from an impostor.

2.1.6.2 Certificate Authorities and trust hierarchies

A user of a security service requiring knowledge of a public key generally needs to obtain and validate a certificate containing the required public key. To verify that the certificate is authentic, the receiver needs the public key of the CA that issued the certificate.

Most Web browsers come configured with the public keys of common CAs (such as VeriSign). However, if the user does not have the public key of the CA that signed the certificate, an additional certificate would be needed in order to obtain that public key. In general, a chain of multiple certificates may be required, comprising a certificate of the public key owner signed by a CA, and possibly additional certificates of CAs signed by other CAs. Many applications that send a subject's certificate to a receiver send, not only just that certificate but also all the CA certificates necessary to verify the certificate up to the root.

2.1.6.3 Obtaining and storing certificates

As has been discussed, certificates are issued by a CA. Clients usually request certificates by going to the CA's Web site. After verifying the validity of the request, the CA sends back the certificate in an e-mail message or allows it to be downloaded.

Requesting server certificates

Server certificates can be either self-signed or they can be signed by an external CA. The server environment will determine which kind of certificate should be used: in an intranet environment, it is generally appropriate to use self-signed certificates. In an environment where external users are accessing the server over the Internet, it is usually advisable to acquire a server certificate from a well-known CA, because the steps needed to import a self-signed certificate might seem obscure, and most users will not have the ability to discern whether the action they are performing is of trivial consequence or not. It should also be noted that a root CA certificate received over an untrusted channel such as the internet does not deserve any kind of trust.

2.2 Firewall concepts

A firewall machine is a computer used to separate a secure network from a non-secure network (Figure 9). Such networks are typically based on the TCP/IP protocol, but the concept of a firewall concept is not restricted to just TCP/IP.

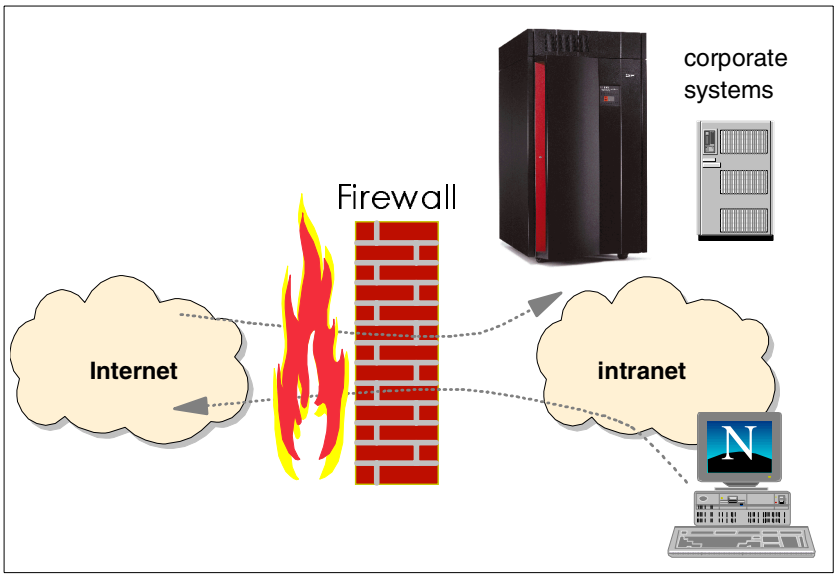

Figure 9. The firewall concept

Firewalls have become an important concept in TCP/IP-based networks, because the global Internet is a TCP/IP-based network and is often perceived as being a non-secure place to enter or traverse. Yet you still want your

intranet (perceived as being a secure place) to be connected to the non-secure Internet.

The reasons for establishing connections between an intranet and the Internet are many, but generally fall into two categories:

- You want to provide a service to the Internet community or want to conduct business on the Internet.
- You want to allow your internal employees to access the vast amount of services on the Internet, as well as the ability to exchange or share information with other users on the Internet or through the Internet.

At this point, it might be useful to define the following terms:

- The term *intranet* refers to an internal TCP/IP network.
- The term *Internet* refers to the World Wide Web, and the associated infrastructure of news groups, e-mail, chat rooms and other services.
- The term *extranet* refers to TCP/IP networks of different companies connected with a secure connection, perhaps using virtual private network technology (VPN).

Doing e-business on the Internet is very different from just serving static information out of a Web server. Doing e-business means that you have to establish an environment where users on the Internet are able to interact with the applications and data that your daily existence as a company is based on and relies upon.

That data and those applications are likely, to a large extent, to be located in your environment, which means that you probably already are, or in the near-term future will be, challenged with the request to establish Internet access to your production environment.

When you connect your intranet to the Internet and define a strategy for how your firewall should function, you may think that it is sufficient to block all types of traffic that represent a risk, and allow the remaining traffic to pass through the firewall. However, such a strategy is based on the assumption that all risks are known in advance and that existing well-behaving traffic will remain well-behaving; such an assumption is a mistake. New ways of exploiting existing applications and well-known application protocols are being found every week, so an application that may be considered harmless today may be the instrument of an attack tomorrow.

2.2.1 General guidelines for implementing firewalls

A few general guidelines for implementing firewall technologies are worth including.

Before you start connecting your intranet to the Internet, define a security policy for how your firewall should function and how demilitarized zones should be configured. Decide what type of traffic is allowed through the firewall, and under what conditions, what kind of servers are to be placed in demilitarized zones, and what type of traffic is allowed between the demilitarized zone and the intranet.

When actually configuring your firewall, start by disallowing everything and then proceed by enabling those services you have defined in your security policy. Everything that is not specifically allowed should be prohibited.

If you establish more than a single gateway between your internal network and the Internet, make sure that all gateways implement the same level of security. It is common practice to use different firewall products in a vertical setup (product A between the Internet and the demilitarized zone and product B between the demilitarized zone and the intranet). That way, a hacker exploiting a vulnerability in product A is still stopped by product B. Of course, it does not make sense to use this concept in a horizontal setup (one gateway uses product A, the other one product B) because a hacker will get in at the weakest link.

If you build a perfect firewall on one end of your network while users on the other end dial in to the Internet from their LAN-attached PCs, enabling those PCs to act as IP routers between your internal network and the Internet, a hacker is soon going to exploit that back door into your network instead of wasting his time trying to break through your firewall.

One of the most important aspects of a firewall is its ability to log both successful and rejected access events. However, these logs are worth nothing if you do not set up daily administrative procedures to analyze and react to the information that can be derived from these logs.

By analyzing the firewall logs, you should be able to detect if unauthorized accesses were attempted and if your firewall protection succeeded in rejecting such attacks, or if it failed and allowed an intruder to gain access to resources that should not have been accessed. In addition, it might be a good idea to install an intrusion detection system.

This list is not all-inclusive, but merely points out some of the most important aspects of implementing firewall technologies in your network.

So far, the Internet has been considered to be the non-secure place, while your internal network has been considered the secure place. However, that may in some situations be an oversimplification. For example, consider a research department that works with highly confidential information. In such an environment, you may want to protect that research department from your regular users by implementing a firewall between your regular internal network and the network in your research department.

2.2.2 Firewall categories

There are many firewall technologies available, but they can in general be grouped into two major categories:

- Those that allow IP packets to be routed between two or more networks, namely packet-filtering routers.
- Those that disable IP routing, but relay data through specialized application programs, namely application-level gateways or proxies.

2.2.2.1 Packet filtering

A packet-filtering router, as shown in Figure 10 on page 27, is a special type of IP router. What differentiates a firewall packet-filtering router from a normal IP router is that it applies one or more technologies to analyze the IP packets and decide if a packet is allowed to flow through the firewall or not. Such a firewall is sometimes also referred to as a screening filter, or router firewall.

Some packet-filtering techniques only act on data in the headers of individual packets, while others also look at data depending on the type of packet. The traditional packet-filtering router is stateless (each packet is handled independently) but there are products that save state over multiple packages and base their actions on the state information.

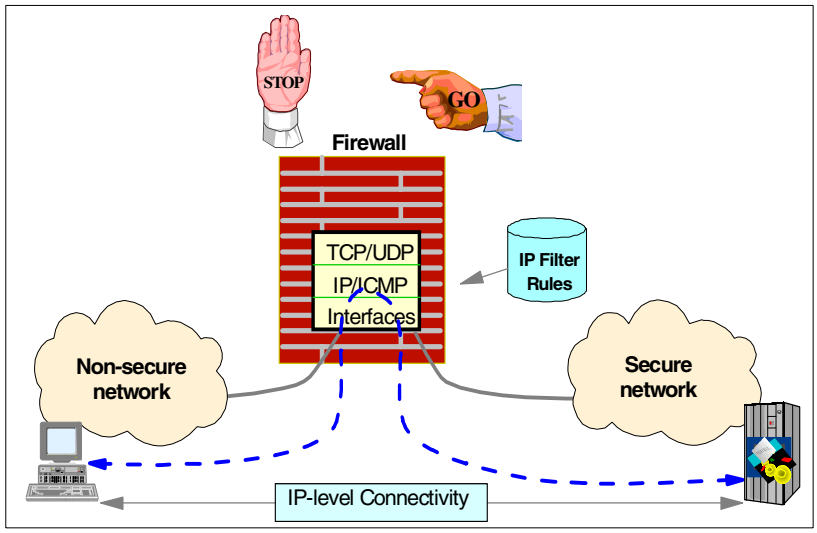

Figure 10. Packet-filtering firewall

2.2.2.2 Application-level gateway

An application-level gateway, sometimes referred to as a bastion host, is a machine that disables IP-level routing between the non-secure network and the secure network, but allows specialized application gateway programs (termed proxies) that run on the firewall to communicate with both the secure network and the non-secure network. See Figure 11.

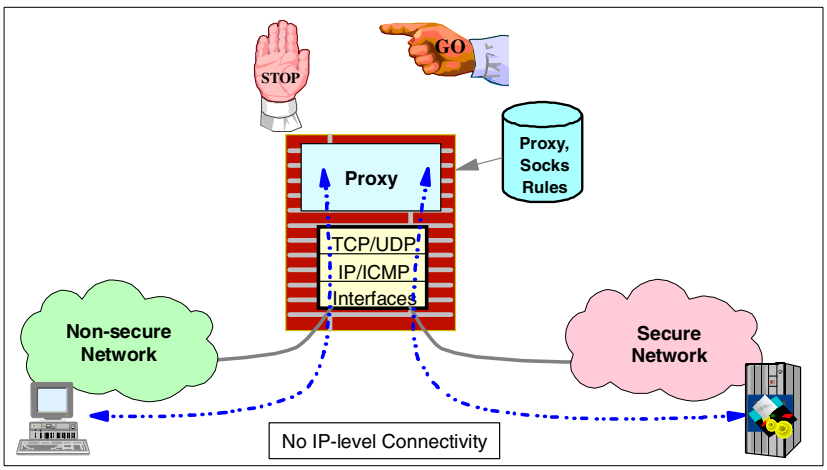

Figure 11. Application gateway firewall

The proxy applications on the firewall act as relay applications between users or applications on the secure and the non-secure networks. Examples of such proxy applications are HTTP or FTP proxy servers. The SOCKS server is also an application-level gateway, but a special kind, sometimes referred to as a circuit level gateway. A SOCKS server can relay all TCP and UDP connections, not just HTTP or FTP sessions. It does not provide any extra packet processing or filtering, and unlike proxy servers, it is often used for outbound connections through a firewall.

A firewall may not always have to be configured as either a packet-filtering router or as a proxy; it may be configured to perform the following functions:

- IP filtering
- Network address translation (NAT)
- Virtual private networks (VPN)
- FTP proxy server
- SOCKS server
- Domain name services

An excellent discussion of firewall technologies can be found in the redbook, *TCP/IP Tutorial and Technical Overview*, GG24-3376.

2.2.2.3 The demilitarized zone

The demilitarized zone (DMZ) is a term often used when describing firewall configurations. Figure 25 shows a typical example. A DMZ is an isolated subnet between your secure network and the Internet. Much as the no-man's land between two entrenched armies, anyone can enter it, but the only things present are those that you want to allow access to anyway. Nowadays, a demilitarized zone is an area in which you place the Web servers and other servers for public access, but which you also wish to protect to some degree.

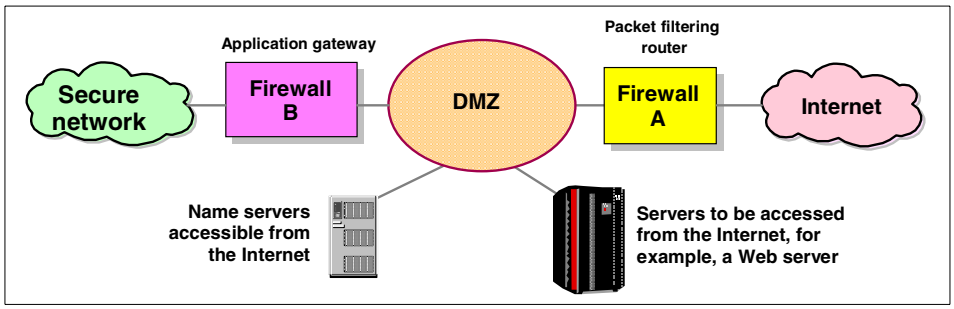

Figure 12. A demilitarized zone

This is achieved by placing an outer firewall (often a packet-filtering router) between the Internet and the servers in the DMZ, and another firewall (often an application-level gateway) between your secure network and the DMZ. The outer firewall is designed to allow into the DMZ only those requests you wish to receive at your Web servers, but could also be configured to block denial-of-service attacks and to perform network address translation of the servers in your DMZ. The inner firewall is designed to prevent unauthorized access to your secure network from the DMZ and also perhaps to prevent unauthorized access from your secure network to the DMZ or the connected non-secure network.

When you put a server into a DMZ, it is strongly recommended that you use firewall technologies. You should use firewall technologies to block all traffic into and out of your server that does not belong to the services you are going to offer from this server. This control should be in place even if you already have a filtering router or firewall between the insecure network and this server.

2.2.3 HardenIng

Hardening is a process done to firewalls to make them more secure. All unnecessary services, user accounts, and software on the operating system are removed or disabled.

An operating system is designed to fit computers with different configurations doing different tasks. To accomplish this, extra items are installed with the operating system that will only be used in certain situations. Much of the time, these extra items just take up resources and might cause the computer to run slower. On a firewall, these items become more of a problem. They become unnecessary, and potential security exposures.

Almost everything on a firewall is a potential security exposure. By disabling and removing the unnecessary items, there will be less exposure for a hacker to exploit. All services, user IDs, and software on a firewall should be required only by the operating system or the firewall. Everything else should be removed.

Many firewalls will perform a limited amount of hardening. However, it is the responsibility of the firewall administrator to finish the task.

2.3 Virtual private network (VPN) and IPSec

A virtual private network (VPN) provides secure connections across the Internet, by establishing a “tunnel” between two secure networks. It is a

generic solution that is application- and protocol-independent. A VPN encapsulates the IP datagram into another IP datagram in order to maintain data privacy. It can be used by two disparate parts of a corporation to connect their internal private networks by means of a non-secure network such as the Internet. An example of a VPN configuration is shown in Figure 13.

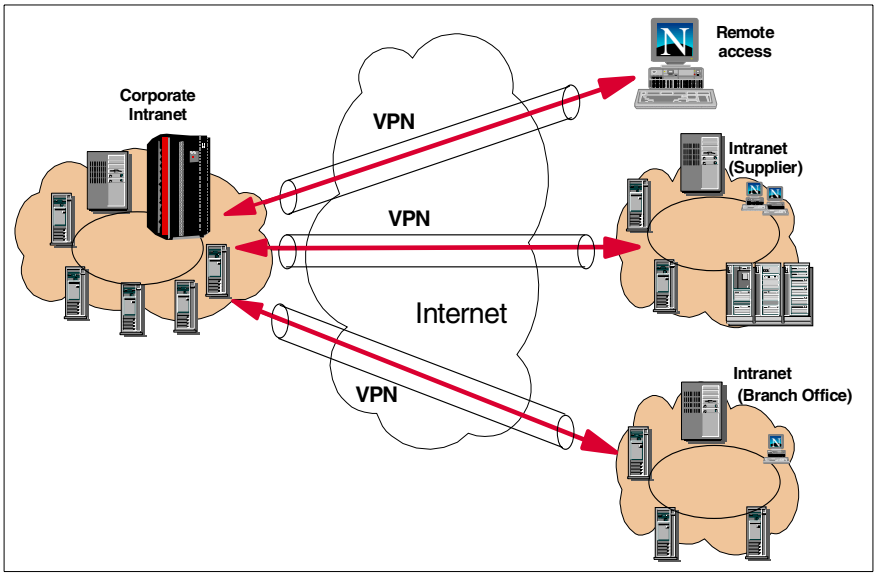

Figure 13. Virtual private networks

2.3.1 IPSec

In Figure 14 the TCP/IP layered protocol stack is shown, with the security-related protocols associated with each layer:

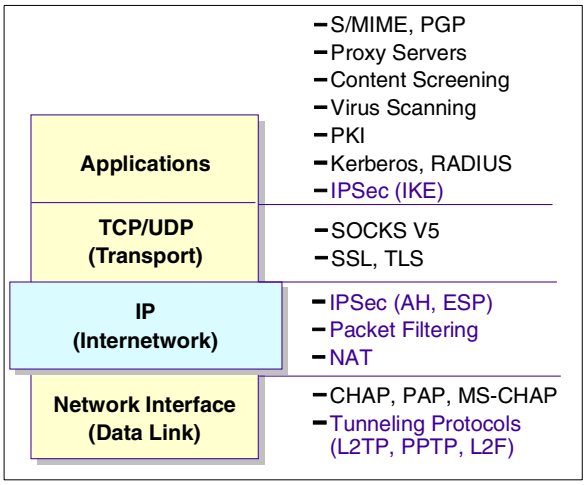

Figure 14. The TCP/IP protocol stack and the security-related protocols

Within the layered communications protocol stack model, the network layer (IP in the case of the TCP/IP stack) is the lowest layer that can provide end-to-end security. Network-layer security protocols provide blanket protection for all upper-layer application data carried in the payload of an IP datagram, without requiring a user to modify the applications.

The IP Security Architecture (IPSec) open framework is defined by the IPSec Working Group of the IETF. IPSec is called a framework because it provides a stable, long-lasting base for providing network layer security. It can accommodate today's cryptographic algorithms, and can also accommodate newer, more powerful algorithms as they become available. IPv6 implementations must support IPSec, and IPv4 implementations are strongly recommended to do so.

IPSec is comprised of a number of components described in individual RFCs that are designed to operate together:

- Security Protocols - IP Authentication Header (AH) provides data origin authentication, data integrity, and replay protection while IP Encapsulating Security Payload (ESP) provides data confidentiality, data origin authentication, data integrity, and replay protection.
- Security Associations - an SA is a kind of session between two hosts defining the protocols to be used when transmitting data. ISAKMP (Internet Security Association and Key Management Protocol) is a generic framework for negotiating SAs and keys.

- Key Management - Internet Key Exchange (IKE) provides a method for automatically setting up security associations and managing and exchanging their cryptographic keys.

Security Associations

An IPSec Security Association (SA) corresponds to a session between two hosts. It defines the set of protocols and, with these, the negotiated algorithms and keys that are to be used when transmitting data between two hosts. An SA for data traffic is always unidirectional, so for a pair of hosts that are to communicate securely, at least two SAs, one for each direction, are needed. This differs from other protocols that make use of sessions such as, for instance, SSL. An SSL session covers the transmission in both directions.

2.3.1.1 Negotiating Security Associations (ISAKMP and IKE)

Before any data can be sent between two hosts using IPSec, a SA needs to be established. The IPSec architecture provides two methods for establishing an SA: a manual tunnel or ISAKMP/IKE.

With manual tunnels, the SA and keying material are generated on one of the hosts (the tunnel owner), transferred to the other host (the tunnel partner) with an out-band transport mechanism, and then imported. This procedure needs to be repeated whenever the validity of the keys has expired and new keying material needs to be generated.

Contrary to manual tunnels, ISAKMP and IKE provide automatic management of sessions and keys. ISAKMP provides a generic framework for the negotiation of SAs and keying material. It defines the procedures and packet formats to establish, negotiate, modify and delete SAs, but it does not provide any specific key-generation techniques or cryptographic algorithms.

Internet Key Exchange (IKE) is based on two protocols: Oakley (*The Oakley Key Determination Protocol*, by H. Orman; RFC 2412, November 1998) and SKEME (*SKEME: A Versatile Secure Key Exchange Mechanism for the Internet*, by H. Krawczyk; IEEE Proceedings, 1996). For the key exchange, Diffie-Hellman (DH) shares are used and the shared key thus obtained is used to derive the keys for data encryption and message authentication. Authentication can be performed with one of three alternatives:

- Digital signatures
- Public key encryption
- A shared secret (a key previously known to both parties)

The use of DH shares causes the connection to have a property called “perfect forward secrecy”. This means that even if the keys for one session are completely compromised, the keys for previous sessions are still safe.

Phases: it takes two

Two hosts can communicate with each other in many different ways that may need different sorts of protection. For instance, some traffic may need encryption and authentication, while other traffic may only need authentication.

IKE uses a two-phase approach to be able to meet these different needs with minimal overhead. In phase 1, an ISAKMP SA is negotiated to create a secure, authenticated channel between the two hosts. The ISAKMP SA is a single, bidirectional security association. In phase 2, the SAs for the individual type of traffic (one SA for each direction) are negotiated using the authenticated channel established in phase 1.

Due to the Diffie-Hellman key exchange and the authentication, phase 1 is computationally rather expensive. Phase 2 does not involve key exchange nor authentication and is much less expensive. Performing phase 1 just once for a pair of hosts and then multiple phase 2 operations for the individual connections is a concept that can improve performance considerably.

Identity protection

In phase 1, certificates and authentication data are exchanged between the hosts. IKE offers *identity protection*, meaning that all information that could identify a host to an attacker or eavesdropper can be encrypted. Depending on whether identity protection is really required, IKE supports two modes for phase 1: *main mode* offers identity protection, while *aggressive mode* does not. In main mode, a shared, secret key is established before the identification information (for instance, the host’s digital certificate) is sent. For a diagram showing IKE main mode, see Figure 15.

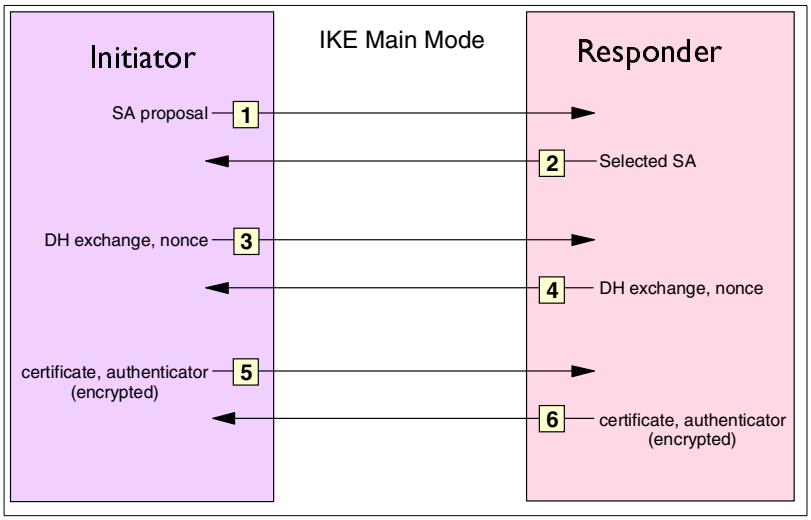

Figure 15. IKE phase 1 main mode

Aggressive mode does not require the DH key exchange to be completed before sending the remaining information. Therefore, there is only one exchange of messages in aggressive mode (see Figure 16).

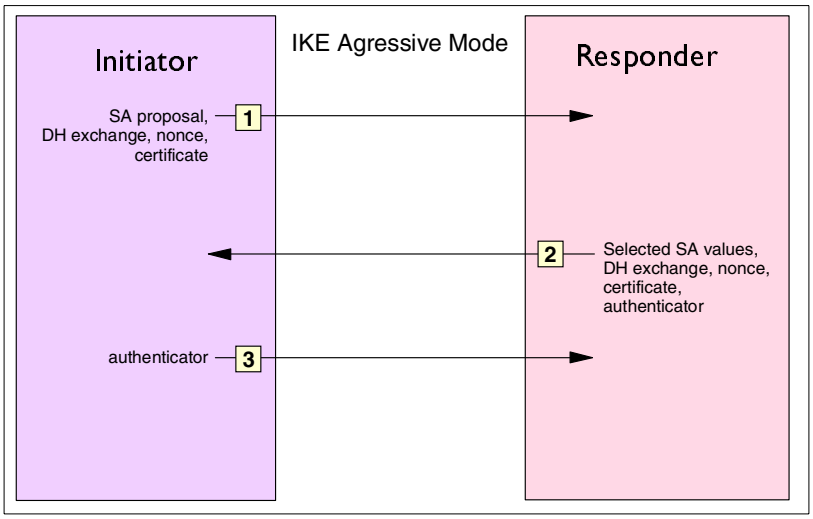

Figure 16. IKE phase 1 aggressive mode

The exchange of messages taking part in phase 2 (negotiation of the SAs for the individual type of traffic) is called *quick mode*. In this mode, the pair of

SAs for the intended type of communication is set up. The required keys for encryption and message authentication are generated from the shared key obtained in phase 1.

2.3.1.2 Transmitting data with IPSec

When a host wants to transmit one or more packets to another host it had not contacted before, it will perform the necessary IKE exchanges to set up the required SAs with the other hosts. Once this has all been performed and the necessary keys are generated, the host proceeds to send the first packet.

IPSec has two formats for sending data, which serve slightly different purposes. *Authentication Header* (AH) provides for message authentication and replay protection, whereas *Encapsulating Security Payload* (ESP) provides for data encryption in addition to message authentication and replay protection. The SA for a communication selects whether AH, ESP, or a combination of both is to be used.

Depending on the type of VPN connection between the two hosts, there are two modes, *tunnel mode* and *transport mode*, that are to be used.

ESP and AH in transport mode

If a VPN connection is being established between two hosts that are the endpoints for the packets transmitted between them, transport mode should be used. Figure 17 shows the format of an Authentication Header (AH) in transport mode.

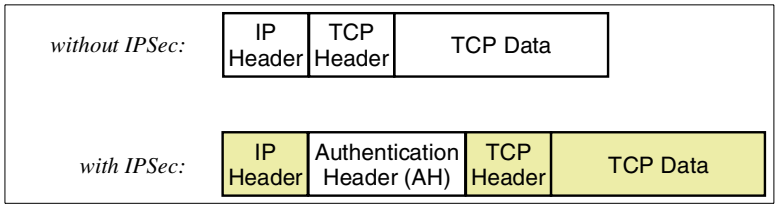

Figure 17. AH in transport mode

The message authentication applied by AH protects the parts of the packet that are shaded in Figure 17. Note that although the IP header is shaded in the diagram, parts of it are not authenticated because they can change in transit between sender and receiver.

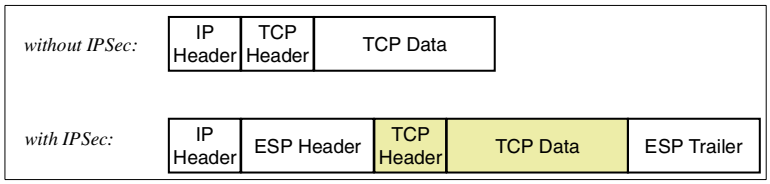

Figure 18. ESP in transport mode

With the Encapsulating Security Payload (ESP) format in transport mode, the TCP header and data are encrypted and, optionally, authenticated. But as can be seen in Figure 18 (the protected areas are shaded), the IP header is afforded no protection at all. However, this should not be a problem because sending and receiving hosts have been authenticated and verified in the SA.

ESP in tunnel mode

A common application of VPNs is the use of a protected tunnel between two secure networks. IPSec-capable firewalls at each end of the tunnel encrypt the packets they send from the secure network through the tunnel; they decrypt the packets they receive from the tunnel and route them to the destination hosts. In this scenario, the SAs do not authenticate the destination hosts (just the firewalls) and an attacker's modification of the IP headers could go undetected.

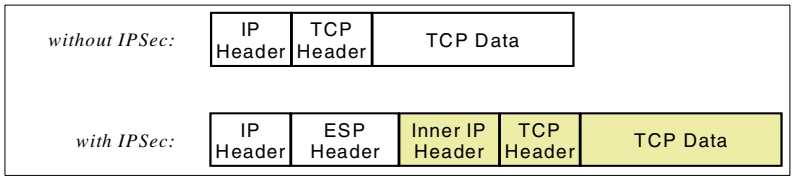

Figure 19. ESP in tunnel mode

In this environment, tunnel mode is to be used. Figure 19 shows the format of ESP packets in this mode; again, protected areas are shaded. The complete original packet, including the original IP header, is used as payload for an ESP packet. The inner IP header has the address of the destination host while the outer IP header addresses the firewall at the end of the tunnel. In this way, the complete packet including the IP header is protected.

In some cases, the AH and ESP formats are combined (applied one after the other) in order to reap both the benefits of IP header authentication with AH and payload (data) encryption with ESP.

For detailed information read:

- *A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions,* SG24-5201
- *Secure e-business in TCP/IP Networks on OS/390 and z/OS*, SG24-5383

2.3.2 Alternative VPN solutions: Layer 2 Tunnel Protocol

A remote access dial-up solution for mobile users is a very simple form of a virtual private network, typically used to support dial-in access to a corporate network whose users are all company employees. To eliminate the long-distance charges that would occur if a remote user were to dial in directly to a gateway on the home network, the IETF developed a tunneling protocol, Layer 2 Tunnel Protocol (L2TP). This protocol extends the span of a PPP connection: instead of beginning at the remote host and ending at a local ISP's point of presence, the virtual PPP link now extends from the remote host all the way back to the corporate gateway. In effect, the remote host appears to be on the same subnet as the corporate gateway.

Since the host and the gateway share the same PPP connection, they can take advantage of PPP's ability to transport protocols other than just IP. For example, L2TP tunnels can be used to support remote LAN access as well as remote IP access. Figure 20 outlines a basic L2TP configuration:

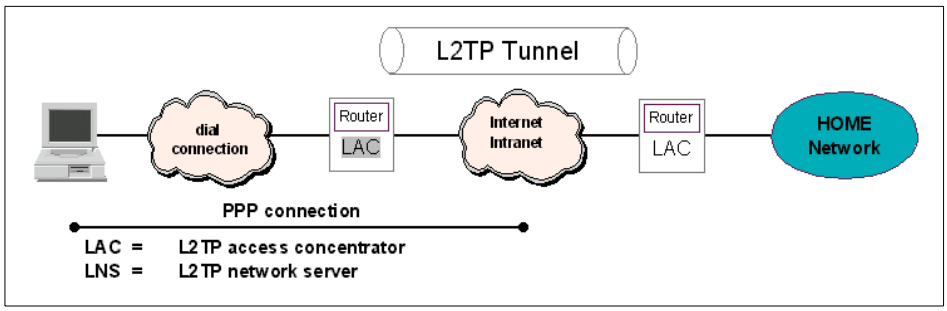

Figure 20. Layer 2 Tunnel Protocol (L2TP) scenario

Although L2TP provides cost-effective access, multiprotocol transport, and remote LAN access, it does not provide cryptographically robust security features. For example:

- Authentication is provided only for the identity of tunnel endpoints, but not for each individual packet that flows inside the tunnel. This can expose the tunnel to various attacks.
- Without per-packet integrity, it is possible to mount denial-of-service attacks by generating bogus control messages that can terminate either the L2TP tunnel or the underlying PPP connection.

- L2TP itself provides no facility to encrypt user data traffic. This can lead to embarrassing exposures when data confidentiality is an issue.
- While the payload of the PPP packets can be encrypted, the PPP protocol suite does not provide mechanisms for automatic key generation or for automatic key refresh. This can lead to someone listening in on the wire to finally break that key and gain access to the data being transmitted.

2.4 Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol developed by the Netscape Communications Corporation that uses encryption to provide privacy and authentication between two applications in TCP/IP. SSL can be regarded as a *transport layer* equivalent of IPSec. Like IPSec, it uses asymmetric cipher algorithms (RSA is normally used) to authenticate users and sign messages, and symmetric algorithms to ensure confidentiality. Unlike IPSec, it is used to protect sessions between particular applications on particular ports; IPSec provides blanket protection between two hosts.

HTTP can use SSL to secure its communications. This allows Web browsers and servers to pass confidential or sensitive data through the Internet or intranet. SSL is also implemented by the Lightweight Directory Access Protocol (LDAP) for secure connections between LDAP clients and LDAP servers, by Telnet/3270, and by a Telnet client such as Host On-Demand for connections between the client and the host system.

2.4.1 SSL overview

SSL was originally developed to protect traffic between a client and a server communicating across the Internet. The latest version of SSL from Netscape (and final version from Netscape) is SSL 3.0. At time of writing, it is by far the most commonly used SSL protocol. According to the latest SSL standard (RFC 2246, *The TLS Protocol Version 1.0*, Appendix E) SSL 2.0 should be phased out “with all due haste”. The IETF TLS-Based Telnet Security document (see 2.5, “Transport Layer Security Protocol (TLS)” on page 42) goes a step further to say that SSL 2.0 is not an acceptable protocol at all. See Figure 21 on page 39 for an outline of some of the SSL protocols and standards.

The use of SSL for Web access is through a protocol called HTTPS. HTTPS is a unique protocol that combines SSL and HTTP. You need to specify https:// instead of http:// as an anchor in HTML documents that link to SSL-protected documents. A client user can also open a URL by specifying https:// to request SSL-protected documents.

Because HTTPS and HTTP are different protocols and use different ports (443 and 80, respectively), you can run both SSL and non-SSL requests at the same time. As a result, you can elect to provide information to all users using no security, and specific information only to browsers that make secure requests. This is how a retail company on the Internet can allow users to look through the merchandise without security, but then fill out order forms and send their credit card numbers using security.

SSL relies on digital certificates and a hierarchy of trusted authorities, as described in 2.1.6.1, “Digital certificates” on page 20, to ensure authentication of clients or servers.

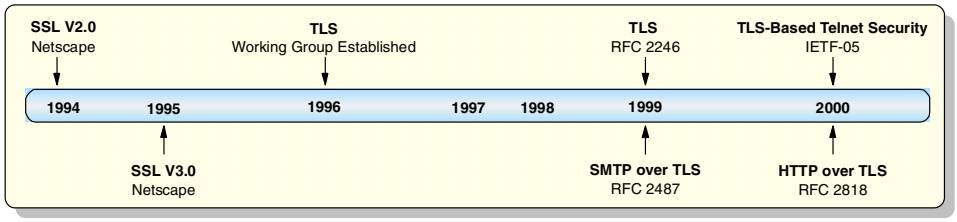

Figure 21. Evolution of SSL

2.4.2 Establishing secure communications with SSL

To use SSL, both the client and the server need to have the software to support this protocol. Because SSL started with HTTP communication, it is used as an illustration.

The latest Netscape and Microsoft browsers support SSL 3.0 and all its features on the client. SSL is composed of two subprotocols:

- SSL Handshake Protocol
- SSL Record Protocol

The SSL Handshake Protocol initializes a secure session, with authentication of the server (and optionally, the client), agreement of encryption scheme, and transfer of encryption keys. A public-key algorithm, usually RSA, is used for the exchange of the symmetric encryption key and for digital signatures. With the server certificate, the client is also able to verify the server's identity. With SSL Version 3.0, the possibility of authenticating the client identity by using client certificates in addition to server certificates was added. The overall flow of these steps is shown in Figure 22:

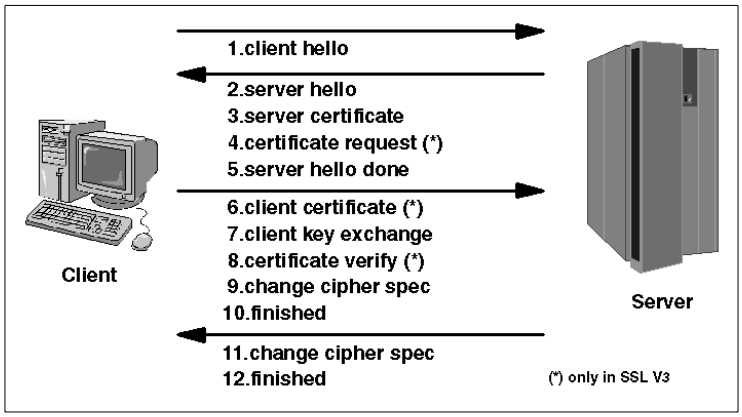

Figure 22. Overview of SSL Handshake Protocol

1. First, the client sends a `client hello` message which lists the cryptographic capabilities of the client (sorted in client preference order) and contains the SSL/TLS protocol version desired. It also contains a random value, a nonce. A nonce is a random value used in communication protocols, typically for replay protection.
2. The server responds with a `server hello` message which contains the cryptographic method (cipher suite) selected by the server, the session ID, another random number and the SSL/TLS protocol version acceptable. The client and server must support at least one common cipher suite or the handshake will fail.
3. Following the server hello message, the server sends its certificate. With Secure Sockets Layer, X.509 V.3 certificates are used.
4. If SSL Version 3 is used and the server application (for example, the Web server) requires a certificate for client authentication, the server sends a `certificate request` message. In the certificate request message, the server sends a list of the types of certificates supported and the distinguished names of acceptable certification authorities.
5. The server then sends a `server hello done` message and waits for a client response. Upon receipt of the `server hello done` message, the client (the Web browser) verifies the validity of the server's certificate and checks that the server hello parameters are acceptable.
6. If the server requested a client certificate, the client sends a certificate or, if no suitable certificate is available, a `no certificate` alert. This alert is only a warning, but the server application can fail the session if client authentication is mandatory.

7. The client then sends a `client key exchange` message. This message contains the so-called pre-master secret, a 46-byte random number that will be used in the generation of the symmetric encryption keys and the Message Authentication Code (MAC) keys, encrypted with the public key of the server.
8. If the client sent a certificate to the server, the client will now send a `certificate verify` message, which is signed with the client's private key. By verifying the signature of this message, the server can explicitly verify the ownership of the client certificate.
9. A similar process to verify the server certificate is not necessary. If the server does not have the private key that belongs to the certificate, it cannot decrypt the pre-master secret nor create the correct keys for the symmetric encryption algorithm, and the handshake must fail.
10. Now, the client uses a series of cryptographic operations to convert the pre-master secret into a master secret, from which all key material required for encryption and message authentication is derived. Then, the client sends a `change cipher spec` message to make the server switch to the newly negotiated cipher suite.
11. The `finished` message immediately following is the first message encrypted with this cipher method and keys.
12. After the server responds with a `change cipher spec` and a `finished` message of its own, the SSL handshake is completed and encrypted application data can be sent.

The SSL Record Protocol transfers application data using the encryption algorithm and keys agreed upon during the handshake phase. As explained above, symmetric encryption algorithms are used, because they provide much better performance than asymmetric algorithms.

2.4.3 SSL considerations

As discussed, security functions such as SSL are needed to send sensitive data safely if you connect your system to an insecure network such as the Internet. On the other hand, using such security functions has performance impacts, including utilizing additional CPU cycles and degrading Web server performance.

Furthermore, SSL does not satisfy every security requirement. While it protects against eavesdropping and alteration of data, it cannot protect the server from an attacker masquerading as a trusted user. For these security concerns, the risk can be minimized by the use of access controls or firewalls.

To maintain SSL security you have to manage the key carefully, especially when using self-signed certificates, because the whole system environment is affected by the security of the Certificate Authority's key database.

2.5 Transport Layer Security Protocol (TLS)

SSL 3.0, has outgrown the scope of being a Netscape standard. Continued development of the protocol fell into the hands of the Internet Engineering Task Force in 1996. The result was that SSL 3.0 evolved into the proposed standard for Transport Layer Security, RFC 2246.

TLS is the latest in the continuing evolution of SSL. TLS 1.0 might just as readily been titled SSL 3.1. In fact, when negotiating a TLS handshake, the client and server hello messages will use version specification 3.1 (SSL 3.0 uses version specification 3.0).

Note

As of the time of writing, TLS (RFC 2246) is not supported on the TN3270 server. Only IETF Internet-Draft TLS-Based Telnet Security (negotiated Telnet) is supported.

So, what exactly is new in TLS? Not very much. The protocol syntax and handshake flow remains virtually unchanged. The significant difference is that the hello message for TLS must contain Version 3.1. Once it has been agreed by both client and server that 3.1 is to be used, cipher suite exchanges will use a prefix of `TLS_` instead of the SSL 3.0 prefix of `SSL_`.

Finally, TLS 3.1 is a protocol designed with the intent of allowing enhancements for future improvements to privacy over TCP connections.

2.5.1 Negotiated Telnet 3270

Negotiated Telnet is an implementation of IETF TLS-based Telnet Security. The name TLS-based Telnet Security is a little misleading because this IETF Internet draft functions equally well with TLS 1.0 and SSL 3.0. In other words, the actual security protocol used with TLS-based Telnet Security is SSL 3.0. When used with Host On-Demand the assumption is that, most likely, if an application has been written to use negotiated Telnet, it will probably also have been written to support TLS 1.0.

What does the TLS-based Telnet Security define? It adds a new IAC (Interpret As Command) option and suboption. The `START_TLS` option allows the client (`WILL START_TLS`) or the server (`DO START_TLS`) to initiate a request for

a secure session. Once TLS has been agreed upon, the session immediately drops into negotiation of either SSL or TLS. Negotiation of other Telnet IAC options is suspended until the security negotiation has successfully completed.

If the client and server cannot agree upon the `START_TLS` option, then the Telnet server can opt to drop into native TLS/SSL security negotiation (`CONNTYPE SECURE` in the TCP/IP profile data set).

Why add a new IAC option? The foremost advantage is that this option places the control of session security into the TN3270 world (instead of leaving it up to the transport layer). If a Telnet client won't accept a `DO START_TLS` option, the Telnet server can choose to end the session (`CONNTYPE NEGTSECURE` in the TCP/IP profile data set).

The other significant advantage of placing encryption negotiation into the TN3270 option data stream is that a single port can be used for encrypted and non-encrypted sessions. Prior to negotiated Telnet, a separate port for secure and non-secure sessions had to be used. Since all TN3270 clients default to port 23, this was not an ideal situation.

A typical TLS-based Telnet SSL flow is shown in Figure 23.

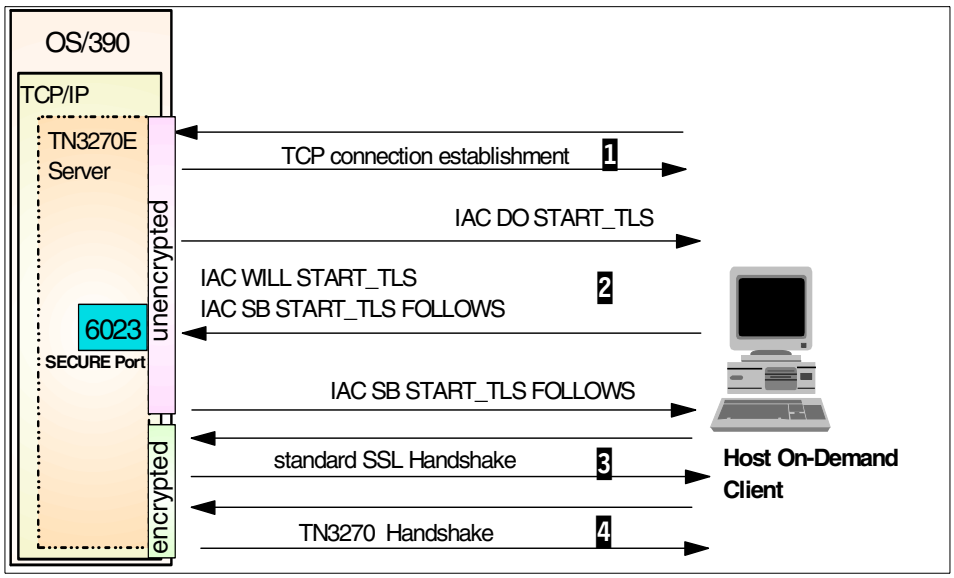

Figure 23. TLS-negotiated security session negotiation

1. IP connection establishment.

2. The Telnet server sends the `IAC DO START_TLS` command to the client to verify if it wants to perform the SSL negotiation.
3. If a positive response is received, then Telnet begins a normal SSL handshake.
4. If no positive response is received, the connection will be dropped.

The `IAC DO START_TLS` Telnet command, sent from the server, activates TLS at the beginning of a Telnet connection. The client can respond to this command by sending the `IAC WILL START_TLS` command, if the negotiation of a TLS connection is required. With the `IAC DONT START_TLS` command, the client can refuse the TLS connection negotiation. Sending the `IAC SB START_TLS FOLLOWS IAC SE` command initiates a TLS negotiation. When this subcommand has been sent and received, the TLS negotiation will begin.

If Enable Security (SSL) is Yes and Telnet-negotiated is Yes, then the Telnet connection will be started normally without SSL. However, the 3270 session will not start until the SSL negotiation completes successfully. If the server `WONT STARTTLS`, then the session will not start, and an error message will be issued stating `Security was requested, but the server does not support security`.

If Enable Security (SSL) is No and the server requests to start the session using TLS-negotiated security, Host On-Demand will not start the session and an error message will be displayed on the status bar stating `The server requested security, but Security is not enabled.`.

To understand the data flows in more detail, refer to Appendix D, “Sample Telnet-negotiated traces” on page 253 for sample traces.

2.6 SOCKS server

SOCKS is a standard for circuit-level gateways. As with the other proxy types, the connection session is broken at the server; when the user starts a client application with the destination server, the client initiates a connection to the SOCKS server.

The SOCKS server validates that the source address and user ID are permitted to establish the connection. From then on, the SOCKS server completes the connection. Unlike application-level gateways, SOCKS is not application-specific. However, it does require clients whose TCP/IP stacks have been made SOCKS server aware, called SOCKSified. For older operating systems, this means a modification of the network stack. Newer

operating systems now come with SOCKS server-enabled TCP/IP stacks, so SOCKS is an even more appealing option than before.

A SOCKS server has the same objectives as the proxy application servers, that is, to break the session at the firewall and provide a secure gateway for access control. It also has the advantage of greater simplicity for the user, at a cost of a little extra administrative work. This is due to the modification required on the protocol stack of the client machines. However, once the TCP/IP protocol stack is SOCKS server-enabled, users can use services with no changes in their usual procedure.

2.7 Network address translation

With the incredible growth of the Internet over the past few years it is getting difficult to get as many unique IP addresses as there are hosts connected to it. The best solution is to use reserved IP addresses for intranets. But how do you communicate to the Internet?

Network address translation (NAT) does three things for you:

1. Allows you to communicate with the Internet by using private addresses

2. Hides your internal network structures

3. Routes traffic to normally non-routable IP addresses

Though NAT looks like a good choice, you have to remember that you have a direct connection to the Internet. There is no breaking of connections as in application-level gateways. NAT comes in handy if you have some protocol that is not supported by any proxy or SOCKS server.

Chapter 3. IBM WebSphere Host On-Demand security

Whether you are implementing Host On-Demand purely within your corporate network, or you are using it to provide access to your host systems via the Internet, you should be concerned about security.

Host On-Demand has the following security capabilities built into the product:

1. All clients are implemented as signed applets
2. Support for Secure Sockets Layer (SSL)
3. Support for Telnet-negotiated sessions
4. Support for smart cards
5. Support for the configuration servlet

This chapter will explore these security capabilities, plus what you need to be concerned about in order to do administration securely, understand what protection HTTPS provides you, and what ports Host On-Demand uses, in order to plan for firewall implementations.

3.1 Signed applet support

The original Java security model prevented a Java applet from:

- Communicating with servers other than the one from which it had originated
- Accessing system resources such as hard disks, printers and the clipboard

These constraints are often referred to as the sand box. Their purpose was:

- Prevent an applet from causing harm on the Internet, which it might be able to do if it were allowed to connect to any destination
- Prevent an applet from doing harm to the machine to which it was downloaded

This was found to be too restrictive in practice, and Java Development Kit (JDK) Version 1.1 introduced the notion of a signed or trusted applet. Such an applet has an embedded X.509 certificate, which identifies the creator of the applet. A user can instruct the browser to allow certain signed applets to operate outside of the sand box.

To sign an applet, the developer must first obtain a certificate from a Certificate Authority (CA). He can then sign his applet with a special signing tool, which embeds the certificate in the file that contains the applet code. There will usually be two of these: a JAR file for use by Netscape and a CAB file for use by Internet Explorer.

Browsers are pre-configured with public-key certificates from well-known CAs such as VeriSign. When a browser encounters a signed applet from a new source, it checks the embedded certificate to see if it has been signed by one of its pre-configured CAs. If it has, the browser tells the user who the developer is and asks if he trusts the applet (and whether the decision is to be remembered). It also asks if all applets from that developer are to be trusted. If the user agrees, the applet continues to load.

This is a much-simplified description of signed applet security. The following Web sites contain further details:

```
http://www.suitable.com/Doc_CodeSigning.shtml
http://developer.netscape.com/docs/manuals/signedobj/trust/index.htm
```

3.2 Host On-Demand SSL support

Host On-Demand can ensure the privacy of communications through the use of the Secure Sockets Layer (SSL) Protocol when connecting to SSL-capable Telnet servers. Host On-Demand implements SSL Version 3 to provide message privacy and integrity. This section describes how Host On-Demand has implemented SSL.

The key part of SSL negotiation is the client's ability to trust the certificate presented by the server, and the server's ability to trust the certificate presented by the client.

For Host On-Demand clients the public certificates of the trusted CAs are stored in one of three places:

1. WellKnownTrustedCAs.class file
2. CustomizedCAs.class file
3. Microsoft cryptographic database

Host On-Demand now has two places to look for the client certificate if required:

1. A password-protected PKCS12 file accessed via the local file system, or in a URL.

   This is the same level of support as was provided by Host On-Demand Version 4 and the initial release of Host On-Demand Version 5. The URL protocols you can use depend on the capabilities of your browser. Most browsers support HTTP, HTTPS, FTP, and FTPS.

2. A client certificate accessible through the Microsoft cryptographic API (CAPI).

   The Microsoft cryptographic API is the security interface used by Internet Explorer to access its certificates, client or server, and is only available only on Windows platforms. Microsoft introduced this API with Internet Explorer Version 5.

3.2.1 Java class files

There are two key Java class files that are used by the Host On-Demand emulation clients when negotiating SSL sessions with Telnet servers: WellKnownTrustedCAs.class and CustomizedCAs.class. The WellKnownTrustedCAs.class file contains the public certificates of all the CAs that Host On-Demand trusts. The CustomizedCAs.class file contains the certificates of unknown CAs and self-signed certificates.

The WellKnownTrustedCAs.class file is supplied by Host On-Demand and is not to be modified by the customer. If a self-signed certificate or a certificate from a unknown authority (CA) is to be used, the CustomizedCAs.class must be created or updated by the customer.

Both the WellKnownTrustedCAs.class file and the CustomizedCAs.class files are stored in the publish directory. All Host On-Demand download clients, including cached clients, obtain or refresh these files from the server when the applet is loaded.

Locally installed clients have the WellKnownTrustedCAs.class file installed on the workstation during product installation. A CustomizedCAs.class file is not installed by default, so if a locally installed client requires a certificate from a unknown CA or self-signed certificate, it must be created. The recommended method is to send the certificate to the client and have the user create the CustomizedCAs.class file at the client. Refer to 3.11, “Making server certificates available to clients” on page 112 for further details.

3.2.2 Microsoft cryptographic service provider database

Starting with Version 5.03, Host On-Demand provided an enhancement that allows the administrator to enable Host On-Demand to use the cryptographic API interface to store client certificates and public key certificates for CAs into the Microsoft cryptographic service provider database, hereafter referred to as the cryptographic database. This function has been tested and is supported on the following platforms:

- Windows 98
- Windows NT 4.0
- Windows 2000
- Windows Millennium Edition

The use of this interface provides simplification and usability improvements. All user prompting and card access for client authentication can be performed by the CAPI software. When selected, the Host On-Demand client receives a list of available client certificates and security providers, then presents the list to the user for selection to send to the server. As long as the Microsoft cryptographic database is installed, the option is available on both Netscape and Internet Explorer browsers, and is the preferred interface for Microsoft Internet Explorer.

Through the use of the Microsoft cryptographic service provider database, not only do you have access to the client certificates, but to the CA certificates trusted by the browser as well. Therefore, if the Telnet server is using a certificate by a CA unknown to Host On-Demand, but known to the cryptographic database, then you can use the certificate located in the cryptographic database, thus eliminating the need to add the signer certificate to the CustomizedCAs.class file.

Many of the smart card readers are CAPI-compliant. By leaving hardware-level smart card processing to the CAPI and vendor interfaces, IBM is able to support new security devices without changing the Host On-Demand code. For instance, if a new thumbprint reader device becomes available, Host On-Demand will be able to access it through the use of the CAPI or the vendor interfaces without realizing it is not a smart card.

3.2.2.1 Viewing certificates

Certificates that are registered in the cryptographic database can be displayed in the following way:

1. Start the Internet Explorer 5.x browser.
2. Select **Tools -> Internet Options**.

3. Select the **Content** tab in the Internet Options window, as shown in Figure 24.

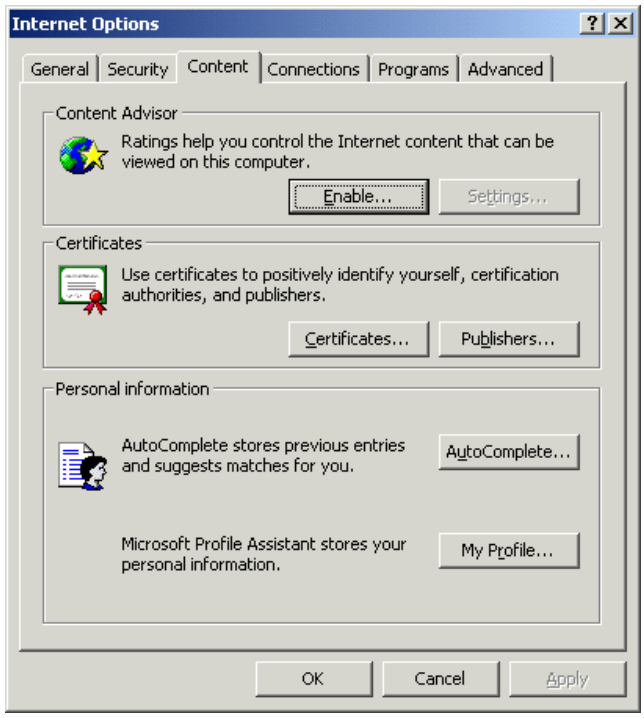

Figure 24. Internet Explorer Content

4. In the Content window, click **Certificates**.
5. In the Certificates window shown in Figure 25 on page 52, select the **Personal** tab. Displayed will be the certificates that will appear in the drop-down list on the Host On-Demand session configuration window and the Server Requesting Certificate window. If the certificate is not in this list, it will be obtained from either the WellKnownTrustedCAs.class file or the CustomizedCAs.class file.

3.2.2.2 Adding a personal certificate

There are some security issues you need to be aware of when you add your personal certificate to the cryptographic database. The following instructions will assist you in the process. Please note that this procedure was developed using Microsoft Internet Explorer Version 5.5.

1. Start the Internet Explorer Version 5 browser.
2. Select **Tools -> Internet Options**.

3. Select the **Content** tab in the Internet Options window.
4. In the Content window, click **Certificates**.
5. Make sure you have selected the **Personal** tab, as shown in Figure 25, then click **Import** to start the process of adding your personal certificate to the database.

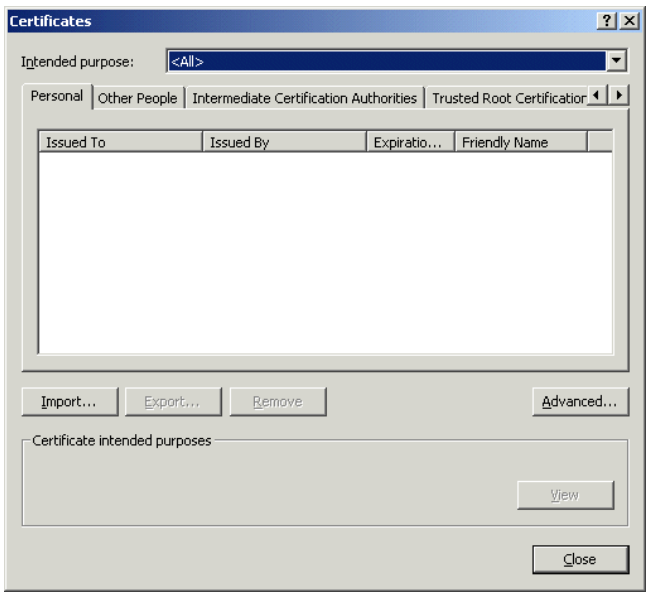

Figure 25. Personal certificates window

6. Click **Next** on the wizard startup window.
7. Enter the location of your personal certificate. You may click **Browse** to navigate to and select the file, or you may just type the location into the input field (see Figure 26). When completed, click **Next**.

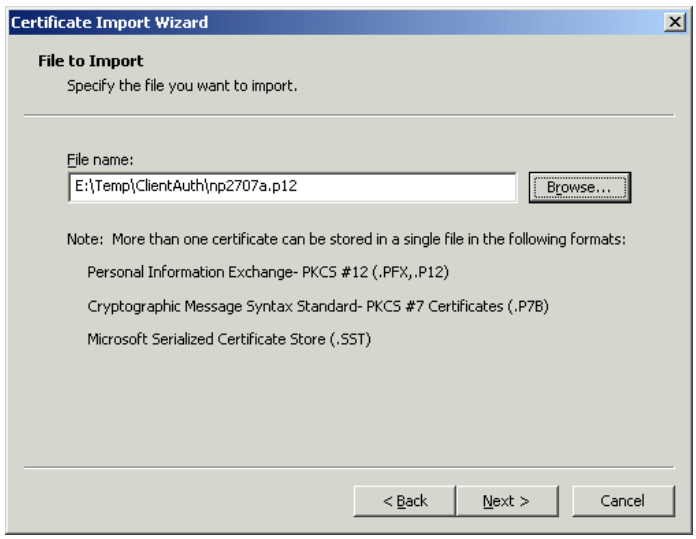

Figure 26. Select certificate

8. You must enter your password for your personal certificate in the entry field as shown in Figure 27. If you select the first check box, then the browser will take an active role in prompting you for permission to use the certificate. Click **Next** to continue to the next window shown in Figure 28.

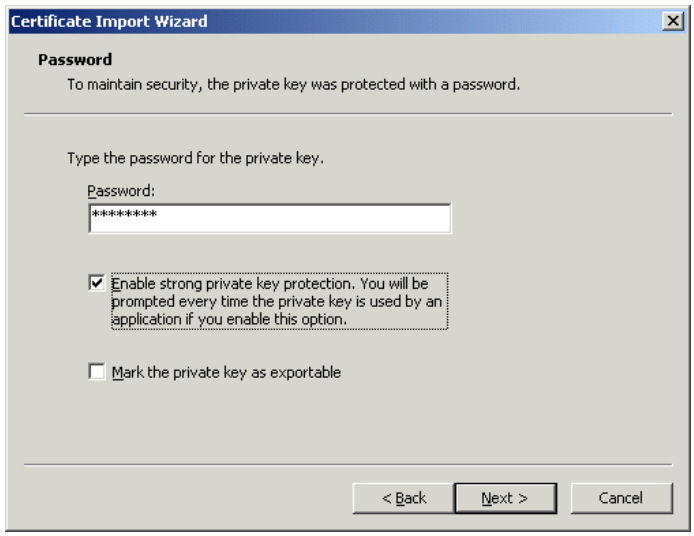

Figure 27. Prompt for certificate password

9. You should specify that the certificate is to be stored in the Personal store by selecting the second radio button, clicking **Browse**, and selecting **Personal** from the resulting list. Click **Next** to proceed to the last window of the wizard where you will click **Finish**.

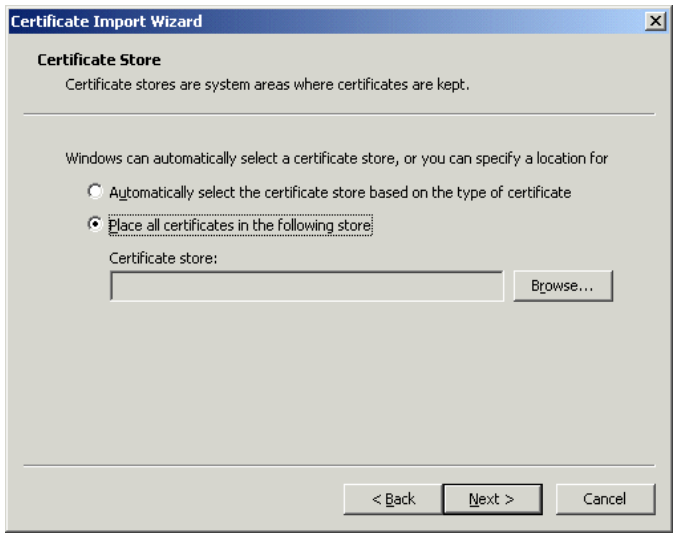

Figure 28. Certificate store

10. If you selected the second check box in Figure 27, then you are finished.

11. If you selected the first check box, then you will be presented with the window shown in Figure 29.

Figure 29. Security wizard

12. Click **Set Security Level** in the window shown in Figure 29 to set the desired security level for this certificate. You will be presented with three choices as shown in Figure 30.

Figure 30. Security level

- If you select the default of **Medium**, you will be notified with a pop-up window (shown in Figure 31 on page 56) that an application (Microsoft Internet Explorer) is requesting access to the protected file. When you click **OK**, this will allow Host On-Demand to present the certificate to the Telnet server. Clicking **Cancel** will not allow Host On-Demand access to the certificate and Host On-Demand will present an error window. When you clear the error window, Host On-Demand will then prompt you for the certificate password.

Figure 31. Notification of certificate use

- If you choose **High** security then you will be prompted, as shown in Figure 32, to provide a common name for your certificate and the password to be used when accessing it. Remember, this password is not the same as the certificate password.

Figure 32. Database password

- If high security is selected, the user will be prompted at run time to provide a password (see Figure 33) in order to release the certificate. Notice that there is a check box to remember the password.

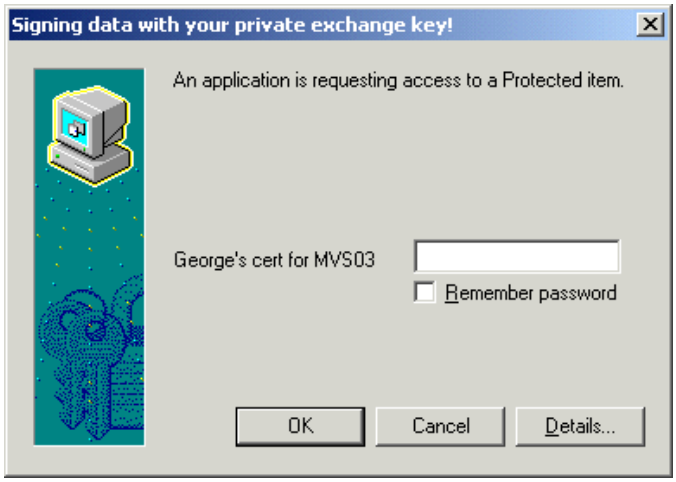

Figure 33. Cryptographic password prompt

- If the check box is selected, then the system will remember the password and the next time the certificate is accessed, the prompt window shown in Figure 34 will appear, and all the user needs to do is click **OK**.

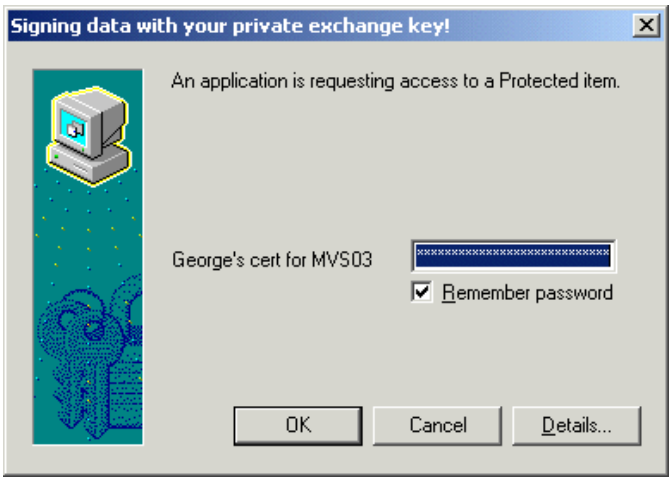

Figure 34. Cryptographic remembered password

3.2.2.3 Recommendation

Implementation of the cryptographic database depends upon a user model that requires each user of the system to log in to the Windows system with a unique ID. If multiple users use the same Windows user ID, then they will share the same copy of the cryptographic database and thus each user will have access to all client certificates.

3.2.3 Host On-Demand SSL implementations

SSL is supported by all Host On-Demand emulator clients, 3270, 5250 and VT, whether they are locally installed, cached, or downloaded clients. There are three ways to use SSL with the emulator clients:

1. Basic SSL
2. Server authentication
3. Client authentication

3.2.3.1 Basic SSL

By default, when SSL is enabled for the Host On-Demand client, a basic SSL session is established. As documented in 2.4.2, “Establishing secure communications with SSL” on page 39, the server will present its certificate to the client during the negotiation process. With basic SSL enablement all that is required is that the client recognize that the certificate is signed by an authority that it trusts.

If the client is running on a Windows platform and the session properties have the MSIE browser key ring file enabled, then the Microsoft cryptographic database is checked first to determine if the signer is trusted. If the signer certificate is not found in the cryptographic database, the cryptographic database is not enabled, or the client is not running on a Windows platform, then the WellKnownTrustedCAs.class file followed by the CustomizedCAs.class files will be checked. If the signer is not found in any of these repositories, the session is rejected. If the signer is found, the session is established.

3.2.3.2 Server authentication

Server authentication is not enabled by default and must be selected on the Security tab in the session properties definition as shown in Figure 37 on page 64.

When server authentication is selected, a secure session is negotiated as described in 2.4.2, “Establishing secure communications with SSL” on page 39. However, immediately the Host On-Demand client looks at the Common

Name field of the server's certificate to determine if the host name of the server presenting the certificate is stored in the Common Name field of the certificate.

Using one or more Java virtual machine (JVM) calls, the client obtains all IP numeric addresses associated with the Common Name in the server's certificate. Next, JVM calls are made requesting all IP numeric addresses associated with the server as specified in the destination field of the session properties definition. When the results of both searches are complete, the client compares the two lists of addresses looking for at least one IP address that appears in both lists. If any IP address appears in both lists, the connection continues and data can be sent; however, if no IP address appears in both lists, then the connection is terminated, and an error generated to the session status line. For server authentication to work, a DNS must be available that can resolve these addresses, or the server address must be defined in the TCP/IP hosts file.

For server authentication to be valid and to give a positive result, two conditions must be met if you are not using the cryptographic database:

1. The client must be locally installed.

   A client downloaded using HTTP cannot be trusted for server authentication because the WellKnownTrustedCAs.class file and the CustomizedCAs.class file are downloaded from the server.

2. The Common Name in the server's certificate must match its Internet name.

The crucial step in the process is when the client checks its list of trusted CAs and self-signed certificates. For a locally installed client, the list is kept on the local hard disk. This is considered adequately secure. However, for a download client, on which the client is a browser that downloads all its code from the server using HTTP(S), the only place the browser can look for the list of trusted CAs or self-signed certificates is on the server from which it has just downloaded the certificate. If that server is an intruder, or if an intruder can intercept and alter data passed from the server to the client, security is breached.

3.2.3.3 Client authentication

Client authentication is similar to server authentication except that with client authentication the Telnet server requests a certificate from the client to verify that the client is who it claims to be. The certificate must be an X.509 certificate and signed by a Certificate Authority (CA) trusted by the server. You can only use client authentication when a server requests a certificate

from a client. Not all servers support client authentication, including the Host On-Demand Redirector.

In order to use client authentication you must:

- Obtain a client certificate.
- Transfer the certificate available to the client by either sending it directly, or making it available via a shared LAN drive or a secure HTTPS connection.
- Always send the password for the certificate via a separate out-of-band secure method so as not to compromise the certificate.

The certificate can be kept in the client's browser, a dedicated security device such as a smart card, or in a local or network-accessed file in PKCS12 or PFX format, which is protected by a password.

When a certificate expires, follow the renewal procedures specified by the CA for that certificate.

3.2.4 FTP client

The FTP client was introduced with Host On-Demand Version 5. This client does not support secure FTP file transfer.

3.2.5 TN3270 client

The 3270 display and printer clients support SSL. The Host On-Demand 3270 display session supports all three types of SSL sessions:

- Basic
- Server authentication
- Client authentication

The 3270 printer session supports the following two types of file transfer:

- Host File Transfer (IND$FILE)

  The IND$FILE mode uses the 3270 data stream to transfer the data; therefore, if the emulator session is encrypted so is the file transfer data.

- FTP

  The FTP option is the same method as deployed with the FTP client described in 3.2.4, "FTP client" on page 60; therefore, SSL is not supported when using this file transfer method.

3.2.6 TN5250 client

The TN5250 emulator client supports SSL sessions. The Host On-Demand 5250 emulator client supports all three types of SSL sessions: basic, server authentication, and client authentication. The 5250 emulator supports two types of file transfer:

1. host file transfer
2. FTP

Host file transfer with TN5250 does not use the 5250 data stream to do the file transfer as does the TN3270 host file transfer (IND$FILE). It uses a file transfer method derived from Client Access. When configuring host file transfer, the window shown in Figure 35 is displayed.

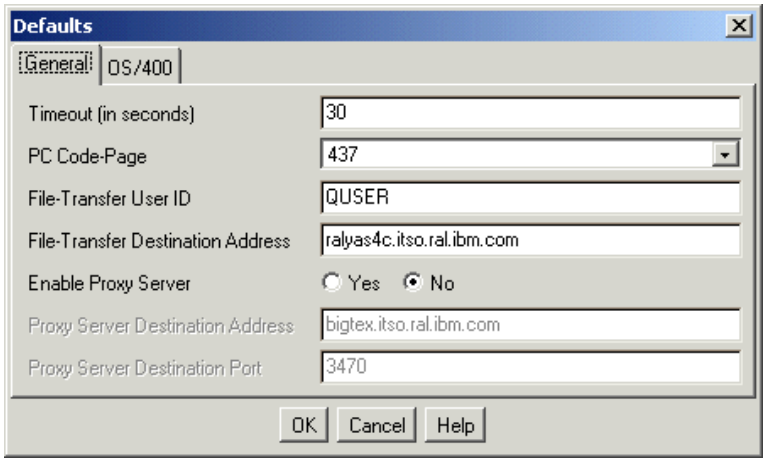

Figure 35. Configure 5250 host file transfer

If you select **No** for Enable Proxy Server, then the file transfer operation will occur to the destination file transfer address using the same security as the 5250 client. This means if the 5250 session is not encrypted, then file transfer data will not be encrypted, but if the 5250 session is encrypted, the file transfer data will also be encrypted using server authentication.

If you select **Yes** for Enable Proxy Server, then the file transfer operation from the client to the proxy server will not be encrypted. Refer to 3.5, “The OS/400 proxy server” on page 82 for details on the operation and configuration of the OS/400 proxy server.

The FTP option is the same method as deployed with the FTP client described in 3.2.4, “FTP client” on page 60; therefore, SSL is not supported when using this file transfer method.

3.2.7 VT client

The VT client supports SSL. Unless the system you will be connecting to supports SSL on the VT session you must use a redirector that does, such as the Host On-Demand Redirector or the Communications Server for AIX Telnet Redirector.

3.2.8 AS/400 Database On-Demand client

The Database On-Demand client is provided as part of the OS/400 toolkit. The MOD 3 version of the toolkit introduced with Host On-Demand Version 5.0 introduced the ability to use SSL with the Database On-Demand client.

Figure 36 illustrates the `;Secure=TRUE` parameter that may be added to the Database URL when initiating a database query.

Note

If you are using the OS/400 proxy for port reduction, the session between the client and the proxy will not be encrypted even if the secure parameter is specified.

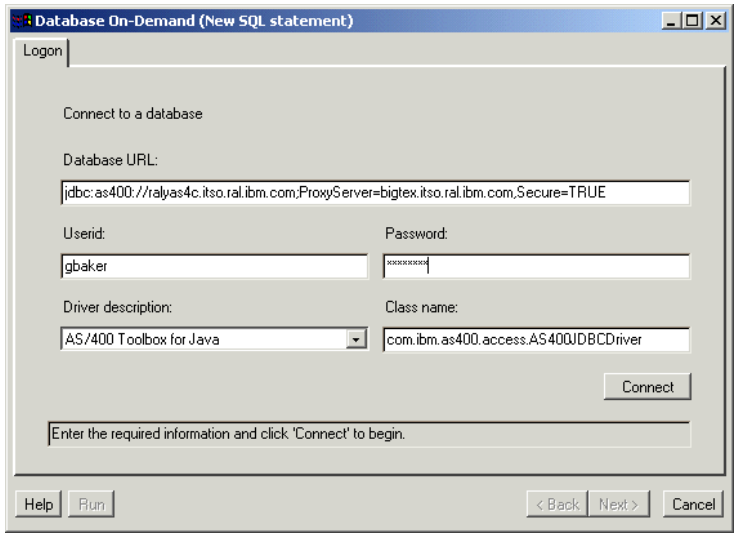

Figure 36. Database On-Demand configuration

If you wish to use the OS/400 proxy server you need to add `;Proxy Server=ralyas4c.itso.ral.ibm.com` to the Database URL, where `ralyas4c.itso.ral.ibm.com` is the OS/400 proxy server destination address.

Note

If you are using the Netscape browser and you see this message when logging on:

```
Please disable the JIT compiler and restart the browser.
```

you must stop your browser, rename the Netscape jit*.dll file so that it is not a .dll file type and restart your browser. This file is located in the \program files\netscape\communicator\program\java\bin\ directory.

3.3 Defining a secure Telnet session

There are many options available when enabling security for a Telnet session. In order to understand the interrelationships and operational implications, each of the settings for each option as shown in the Session Security window (Figure 37) is examined.

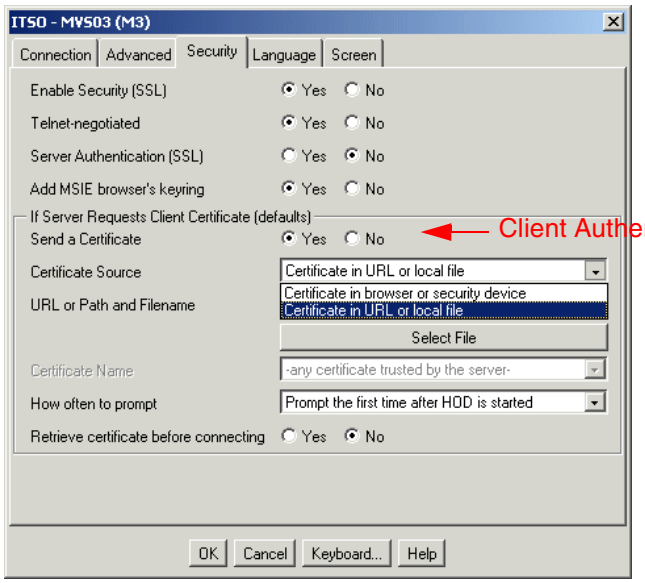

Figure 37. Session configuration - security

3.3.1 Enable Security (SSL)

The first and most important option is whether or not to enable security (SSL). If you click **No**, the remaining options in this configuration window will be unavailable. Thus, the client will not attempt to do any SSL and all transmissions will be in the clear. If you clicked **Yes**, then the remainder of the options become available, and at a minimum basic SSL (see 3.2.3.1, “Basic SSL” on page 58) will be attempted.

3.3.2 Telnet-negotiated session

This option is not available unless you first enable SSL. Selecting Telnet-negotiated determines if the SSL negotiation between the client and the server is done on the Telnet connection or on an SSL connection prior to the Telnet negotiations. The other SSL options are valid regardless of whether the Telnet-negotiated radio button is Yes or No.

If you click **Yes**, then the Telnet protocol defined in IETF Internet-draft TLS-based Telnet Security will be used to negotiate the SSL security after the Telnet connection is established. This support is only applicable with a Telnet server that supports TLS-based Telnet Security. Communications Server for OS/390 V2R10 is the only IBM Telnet server at this time that supports this function.

If you click **No**, the traditional SSL negotiations will be done on an SSL connection with the server, and subsequently the Telnet negotiations with the server will be done. Since this is not yet an RFC, few Telnet servers have this support, so the default is **No**.

3.3.3 Server authentication

The default here is **No**, but should you click **Yes**, then the client will perform the server authentication process as documented in 3.2.3.2, “Server authentication” on page 58.

3.3.4 Add MSIE browser’s key ring

Clicking **Yes** for this parameter allows the client, when running on a Windows platform, to search the cryptographic database when validating the server’s certificate. If the CAs public certificate is not found in the cryptographic provider database, then the applet will look in the WellKnownTrustedCAs.class file followed by the CustomizedCAs.class file if necessary to validate the certificate.

This setting is valid only on a Windows platform. The cryptographic database is available to the Host On-Demand client regardless of the browser being used by the client. This setting has no effect on the client authentication process.

3.3.5 Client authentication

There are many options available if client authentication is to be deployed. First and foremost the session must be configured to respond to the Telnet server with a client certificate. This is done by clicking **Yes** to the Send a Certificate option as shown in Figure 37. Once you click **Yes**, then the remaining options in this section of the window are enabled.

3.3.5.1 Certificate Source

There are two places that the Host On-Demand client will look for the X.509 certificate:

1. The client’s local file system, which includes any configured LAN, NFS, AFS, etc. drives, or from a standard URL
2. A security device, such as a smart card, or from the cryptographic database

If you select **Certificate in URL or local file system**, then you may enter the location where the certificate is found into the URL or Path and Filename field. You may use **Select File** to browse your file system to find the file.

3.3.5.2 Certificate Name

This option will become active when you indicate the certificate is in the browser or security device. Host On-Demand will read the cryptographic database from the machine on which this function is being performed. If you select the default entry, **-any certificate trusted by the server-**, Host On-Demand will search the list returned at run time and select the first certificate recognized by the server. The operator also has the option to view the certificates found in the cryptographic database at configuration time and select one of them as well. This option is fine if the operator is operating on the settings for his own session, but if the operator is an administrator there is no way to know beforehand what certificates will be available on any given client that will use this settings. For administrator operations, refer to 3.3.5.3, “Selecting the client certificate” on page 66.

When a server requests a certificate, the client will check the status of the Send Certificate option set in the session properties file. If it is set to no, a certificate will not be sent and the session may be denied. If the option is set to yes, then the certificate will be located as per the settings in the session configuration file (see Figure 37) and the certificate sent to the server. The user may be prompted for the password of the certificate before it is sent. Finally, the server makes a connection if the client's certificate can be trusted.

3.3.5.3 Selecting the client certificate

When defining the sessions and selecting that the certificate shall be found in the browser or security device (see Figure 38), the administrator has the ability to set a mask that will be used to identify the proper certificate from the cryptographic database.

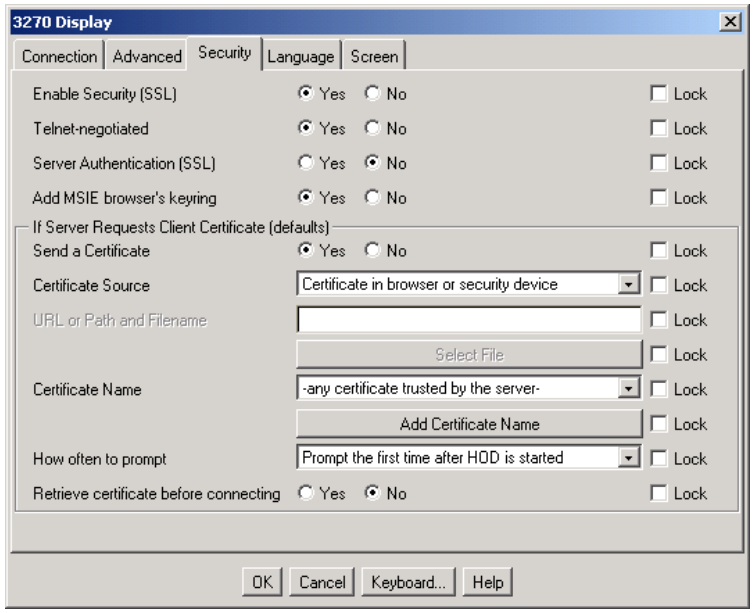

Figure 38. Using certificate from browser

The administrator sets the mask by clicking **Add Certificate Name.** This results in a window (Figure 39) that allows the administrator to specify a mask that will be used in selecting which certificate from the clients cryptographic database will be selected.

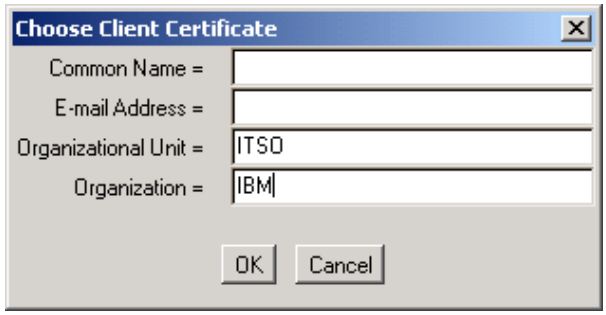

Figure 39. Set up client certificate mask

The mask is not case sensitive and wild cards are not allowed. When the certificate is requested, the cryptographic database is searched and the first valid certificate to fit all the components specified is sent. If no certificates are

found, the client displays the window shown in Figure 40, prompting the end user for further action.

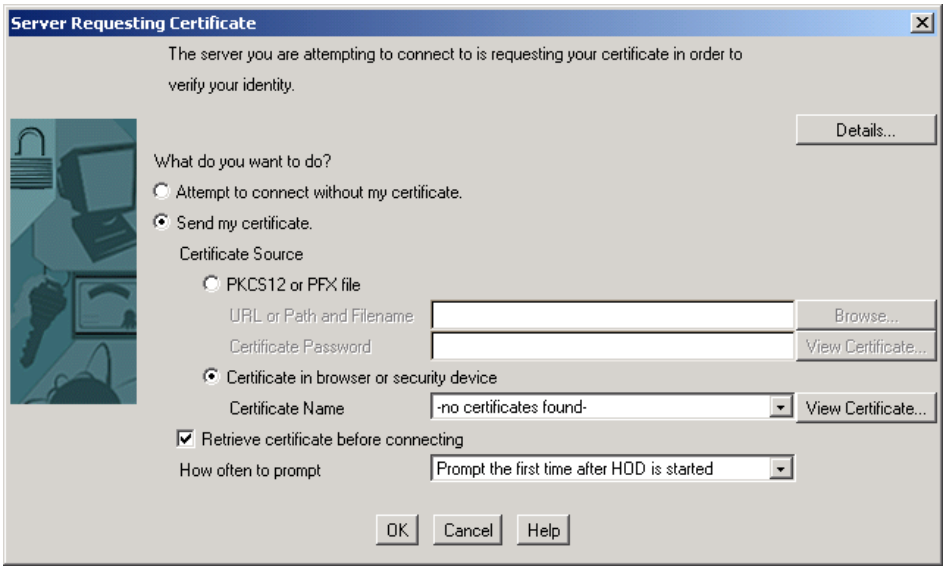

Figure 40. No certificates found

The option to specify a mask is only available to the administrator because if you are the user you should see the list of all of your certificates, but if you are the administrator, there is no way you can see all the certificates on the client's computer.

Note: Using the mask is one technique that may be used to restrict access to valid certificate holders based upon an organizational requirement.

3.3.5.4 How often to prompt

This drop-down box allows you to control the timing of Host On-Demand prompts for client certificates. You can choose to prompt each time a connection is made to the server, or only the first time after starting Host On-Demand. In addition, if your client stores preferences locally (specified when the client HTML file was created via the Deployment Wizard), you can choose to be prompted only once, and have all subsequent connections use the information stored in the local preferences. This option is only available on the client's configuration window.

If you specified browser or security device for the certificate source, then a do-not-prompt option will also be available to you. If the certificate is stored in the cryptographic database (browser), no Host On-Demand password prompt

is required and you may select the no-prompt option. Host On-Demand will not prompt you for your certificate password; however, depending upon the options you selected when you stored your certificate, the cryptographic database may prompt/notify you. Refer to 3.2.2.2, “Adding a personal certificate” on page 51 for more details.

3.3.5.5 Retrieve certificate before connecting

If you click **Yes** for this option, the client will access its certificate before connecting the server, whether the server requests a certificate or not. If you click **No**, the client will access the certificate only after the server has requested it; depending on other settings, this may force the client to abnormally terminate the connection to the server, prompt the user, and then re-connect. It is recommended that you choose Yes if you will be authenticating with a Communications Server for OS/390 system; otherwise, unnecessary error messages may be generated.

3.3.6 Express Logon Facility

The Express Logon Facility (ELF) allows a user running a 3270 client session to log on to a host system without entering a user ID and password. ELF uses digital certificates in place of user IDs and passwords to log on to RACF-enabled applications. Using ELF reduces the number of user IDs and passwords that users need to remember.

To use Express Logon Facility, the following are required:

1. The host session must be configured for SSL with client authentication.
2. The connection must be to one of the supported Telnet servers.
3. Each user must have his own unique digital certificate because ELF and RACF will associate each digital certificate to the user’s RACF user ID and password.
4. You must record a macro that the user will use to log on to the host application. The macro record function steps you through the process for creating an express logon macro.
5. Distribute that macro to the clients.

Some configuration needs to be done on the Telnet servers and on the S/390 system that you are accessing. The information in this book will assist you in configuring the Host On-Demand component. For a complete tutorial and examples of all supported platforms, refer to:

`ftp://ftp.software.ibm.com/software/network/library/whitepapers/elf.pdf`

For additional configuration information, you may also refer to the documentation for the server platform you have implemented:

- *Communications Server for OS/2 Warp - What's New*
- *Communications Server for Windows NT - Readme*
- *Communications Server for AIX - Readme*
- Communications Server for OS/390:
  - The following information APARs:
    - II12362 *V2R10: IP Configuration Guide*, SC31-8725-00
    - II12363 *V2R10: IP Configuration Reference*, SC31-8726-00
    - II12364 *V2R10: IP Quick Reference,* SX75-0121-0)
    - II12365 *V2R10: IP User's Guide,* GC31-8514-04
    - II12366 *V2R10: IP Diagnosis Guide,* SC31-8521-04
    - II12369 *V2R10: IP Messages Volume 3,* SC31-8674-05
    - II12370 *V2R10: IP and SNA Codes*, SC31-8571-04
  - *OS/390 IBM Communications Server Express Logon User's Guide,* found at ftp://ftp.software.ibm.com/software/network/library/whitepapers/elf.pdf
  - *z/OS V1R1.0 CS: IP Migration*, SC31-8773

3.3.6.1 Configuring the client - basic definitions

Before you can start recording a macro using the Express Logon Facility, you have to define a session that is able to provide express logon support. When recording the macro, the session definitions are not checked, that is, you can record a macro for express logon support that might not work correctly when played.

The session must be configured for SSL and client authentication. A client certificate must have been installed on the client or must be accessible from a server. The destination IP address must specify a server that has been set up to support the Express Logon Facility.

Note

If the connection to the TN3270 server is through the Host On-Demand Redirector or the Communications Server for AIX Telnet Redirector, the security option for the Redirector must be set to pass-through.

3.3.6.2 Recording the macro

Recording the macro is started the normal way by clicking **Record** on the session window's tool bar or by selecting **Actions -> Record Macro**. The session itself may have been started from a client by logging in as a user and then opening the intended session window. You may also record an express logon macro as an administrator customizing an HTML page that, when referenced, automatically opens the session window, starts the macro, logs the user on to his host application, and navigates the user to the applications start window.

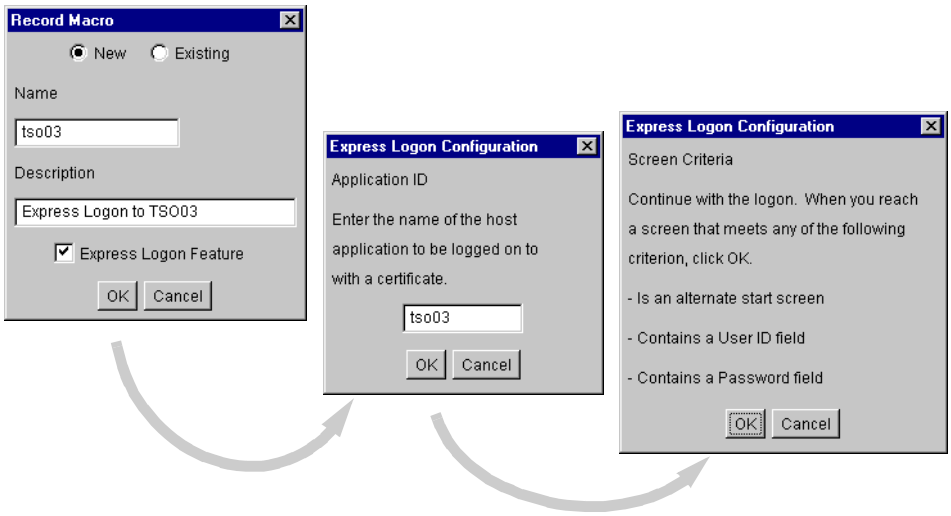

Figure 41. Recording the express logon macro - getting started

Figure 41 shows the sequence of the first three windows that appear when you start recording an express logon macro. On the first window you have to specify the name of the new macro (of course, you may also append to or overwrite an existing macro). Select the **Express Logon Feature** check box to indicate that you want to use express logon. Clicking **OK** causes the second window in Figure 41 to appear, prompting you to enter the application ID of the application you are logging on to with this macro. This is the name of the application that was used when it was defined to RACF on the OS/390 host.

Pick the right application

What you have to enter as the application ID is the LU name of the application as defined in VTAM and to RACF. This name might (and in most installations will) be different from what you enter when VTAM prompts you with an USSMSG10. What you enter there in most cases is an Unformatted System Services (USS) logon command that is translated by VTAM into `LOGON APPLID(applname)`. This *applname* then is what has to be entered in this Express Logon Configuration window as the application ID.

3.3.6.3 Recording the macro - user ID and password

After having entered the application ID and clicking **OK**, the third window in Figure 41 appears, prompting you to actually start recording your actions on the session window.

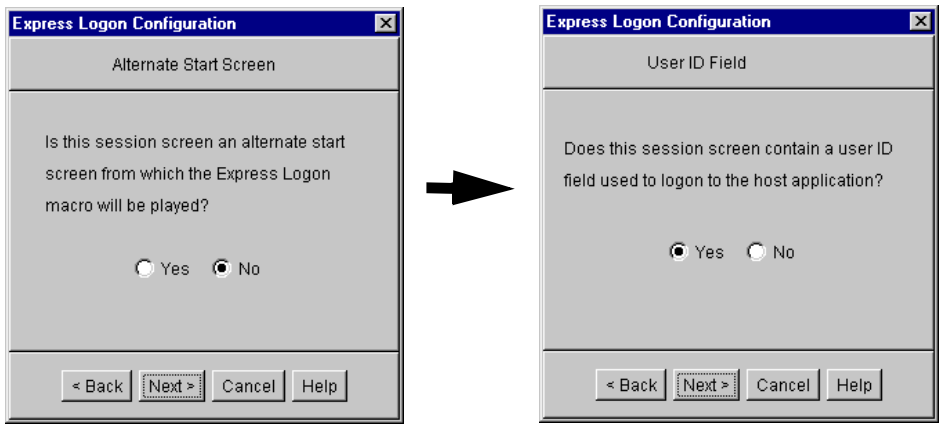

Figure 42. Recording the express logon macro - getting to the user ID field

Once you have reached the window prompting you for the user ID, click **OK** on the third window in Figure 41. The next window (Figure 42) then will ask if this is an alternate start window.

You can define alternate start windows, which can be more than one, in the first or a follow-on editing pass through the macro. This will allow the user to start the macro (or have it started automatically) when the host session is initialized. After having logged off from the application, a different logon window might be presented to the user (for example, the application's logon window and not VTAM's USSMSG10). This then will allow the user to use the same macro for one application, independent of where he starts.

The next window, when not defining an alternate start window, asks if there is a user ID field on the current host window. Clicking **Yes**, then **Next** from the following window leads you to a window that lets you define the position of the user ID field on the host window as shown in Figure 43.

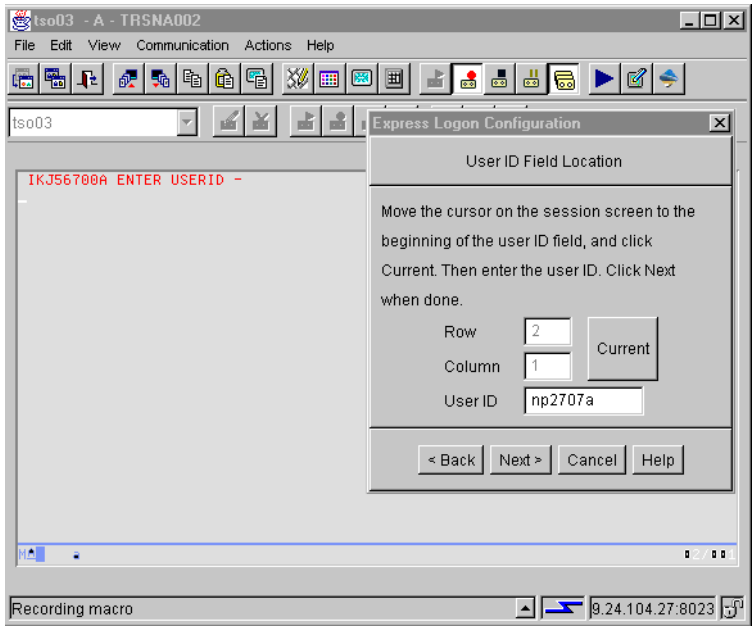

Figure 43. Recording the express logon macro - user ID field

The simplest way of getting the correct row and column is by positioning the cursor on the user ID input field (normally it will already be correctly positioned) and clicking **Current**. This will update the input fields in the window with the current cursor position. In the user ID field, fill in a valid user ID. This user ID then will only be used to log on to the host application while recording the macro; it will not be recorded in the macro. Instead, a placeholder variable, `)USR.ID(`, will be placed in the macro and actually filled into the host window's user ID field when the macro is played. The TN3270 server then will replace this variable with the user's correct user ID.

The next window presented will ask if there is also a password field on the host window that prompts for the user ID. If you answer Yes, the password field is on the same window as the user ID field, or after having navigated to the window prompting for the password, you have to define the position of the password field on the window on a window similar to the one used for the password field as shown in Figure 43. Also, the password you are entering

here is not recorded in the macro. It is only used to actually log on when recording the macro. The macro will again contain the placeholder variable,)PSS.WD(, that will be replaced with the PassTicket by the TN3270 server when playing the macro.

3.3.6.4 Recording the macro - finishing steps

When you click **Finish**, the left window shown in Figure 44 is displayed giving instructions on how to continue. Only when you really want the user to press the Enter key, or whichever PF key is used for the logon, do you follow the instructions to stop the macro immediately. Otherwise, click **OK** to remove the window and continue recording your macro until you have reached the application's start window where you want to leave the user.

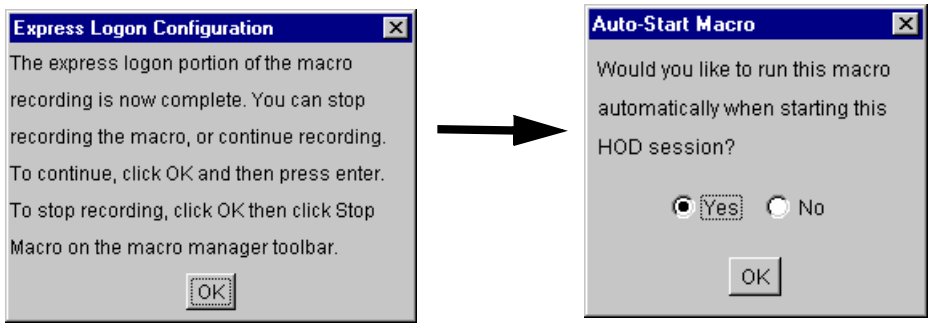

Figure 44. Recording the express logon macro - finishing steps

When you stop recording the macro, a final window (shown on the right in Figure 44) will appear asking you whether you want this macro to be automatically started when the session window is initialized. If you click **Yes**, the corresponding session definitions will be updated.

3.3.6.5 ELF process flow

Refer to Figure 45 on page 75 for a description of the process flow for the Express Logon Facility.

1. The user references a Host On-Demand HTML page that downloads the Host On-Demand code (or loads the cached client from its local disk) and starts an emulator session. The session definitions are being retrieved either from Host On-Demand's configuration server or directly from the HTML page referenced. The session definitions must specify that this session uses SSL encryption with client authentication and an express logon macro must be available to the user.

2. The certificate file is unlocked with a PIN and the user's (X.509) certificate is retrieved to be used during the SSL handshake flows.
3. The IBM WebSphere Host On-Demand client starts a TN3270 connection to its TN3270 server requesting SSL encryption with client authentication using an X.509 certificate. The certificate is sent to the TN3270 server and, after having been validated, is saved for later reference at the TN3270 server for this session. During the Telnet function negotiation, the Express Logon Facility (ELF) is agreed upon using the Handshake Protocol described in RFC 1572.

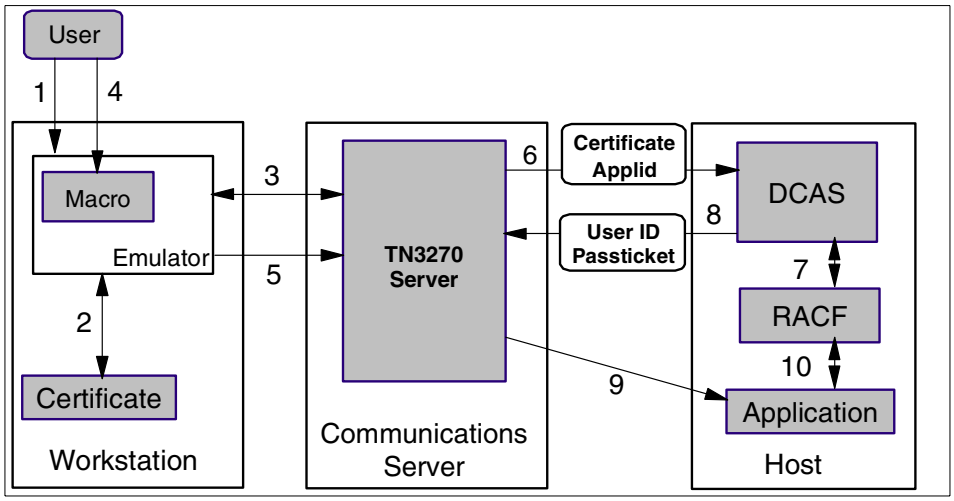

Figure 45. Express logon process flow

4. A macro recorded for this session supporting the Express Logon Facility is started either explicitly by the user or automatically when the host connection is initialized. Before recording the macro, an application ID had to be specified for this macro. This application ID must be the name under which the destination application is known to RACF on the application host.
5. The application ID is sent to the TN3270 server using the Telnet Handshake Protocol in order to make the TN3270 server aware of which application the user intends to connect to using an express logon macro. The logon macro is then played and eventually the host application sends the window(s) designed to prompt for user ID and password (can be on different windows). Instead of filling in a previously recorded user ID and password or prompting the user to provide them in a separate window (as normal macro processing would be), the macro inserts placeholder strings

into the fields for user ID and password,)USR.ID(and)PSS.WD(, respectively.

The TN3270 server intercepts the windows prompting for a user ID and password until it can replace the placeholder strings with a valid user ID and password.

6. If this is the first user requesting assistance for the Express Logon Facility, the TN3270 server establishes a secure and trusted TCP/IP connection to its configured Digital Certificate Access Server (DCAS) using SSL V3 for encrypting the data exchange flows. If this connection has already been established for an earlier request, the existing connection is used. On this connection, the TN3270 server sends a request for a user ID and PassTicket providing the destination application ID and the user's certificate (which was saved for this session during connection establishment).
7. DCAS is a function of Communications Server for OS/390 and interacts with RACF on the S/390 host to verify the validity of the user's certificate. Only certificates for which a user ID has been defined are accepted. If the certificate's associated user ID is, in addition, authorized to access the requested application a PassTicket is generated and, together with the user ID, returned to DCAS.
8. The user ID and PassTicket are sent to the TN3270 server over the secure (SSL-encrypted) TCP/IP connection in response to the previous request.
9. The TN3270 server then replaces the placeholder variable for the user ID,)USR.ID(, on the withheld host logon window and releases the window for transmission also over the SNA LU-LU session towards the host application. It subsequently replaces the placeholder variable,)PSS.WD(, for the password with the PassTicket when the host window requesting the password shows up, if it is not already replaced on the primary logon window together with the user ID.

The host application presents the user ID and PassTicket (received in the 3270 data stream) to RACF in order to check if the user is authorized to log on to this application. The PassTicket should still be valid (unless you have severe performance problems in your system) and RACF will hence grant access to the application.

Important

The initial release of ELF used the variables `$USR.ID$` and `$PSS.WD$`, but because of national language translation issues these variables were changed to `)USR.ID(` and `)PSS.WD(`. This change for Host On-Demand is introduced in Version 5.04. There are plans to update the mid-tier communications servers. The APARs for each platform are as follows:

- JR16019 for Communications Server for OS/2
- JR16009 for Communications Server for Windows NT and Windows 2000
- IY19871 for Communications Server for AIX

The original implementation of Express Logon Facility required a middle-tier server, such as IBM Communications Server on AIX, Windows NT or OS/2. APAR PQ47742 was written to enable ELF directly on the OS/390 system.

As of the writing of this book none of the above-mentioned APARs were available.

3.4 The Host On-Demand Redirector

The Redirector is a Telnet proxy that is written primarily in Java. The Redirector is able to accept connections from clients and pass them on, through a different port, to the next stage in the link to the host. The Redirector has the following main functions:

- Hide the real host system address and port number from the client, a common requirement when providing Internet-attached clients access to secure host systems.
- Provide SSL support for all emulator clients when the Redirector is running on Windows NT or AIX.

The Redirector when running on either a Windows NT or AIX server is capable of supporting SSL sessions in one of the following ways:

- Client-side: SSL is enabled between the Redirector and the client.
- Host-side: SSL is enabled between the Redirector and the host Telnet server.
- Both: the Redirector will support SSL sessions on both the client and the host side simultaneously, managing each SSL session separately.

On all platforms the pass-through mode is supported. The pass-through mode allows the client and the server on the other side of the Redirector to communicate in the clear, or to negotiate a secure session directly, including the use of Telnet-negotiated session, basic SSL, and client authentication.

3.4.1 Redirector certificates

When performing SSL the Host On-Demand Redirector relies on two files for certificate management. These files are found on the Host On-Demand server in the \hostondemand\bin directory, and are:

- HODServerKeyDb.kdb

  This is the server's key database file, created during SSL configuration (described in 3.10.1, "Creating a self-signed certificate" on page 109). It contains:

  - Root certificates for well-known CAs (these are inserted when the file is created)
  - A self-signed certificate (when one exists)
  - Certificates that you have imported from authorities you trust
  - Public keys of all the above certificates
  - Private keys of the self-signed certificate, and of any of your own certificates that have been validated by a CA

- HODServerKeyDb.sth

  This is the password-stash file for the key database. It is used to store the password in an encrypted form that can be used by the Redirector to open the key database file.

These files are not created at installation. They are created by the key-management utility when you install the servers certificate or any unknown CA certificate you may be using.

When using the Redirector and configuring any connection in anything other than pass-through mode, you must install a public key (site) certificate on your server. There are three choices:

1. Use a certificate from one of the well-known CAs whose root certificate is already in the WellKnownTrustedCAs.class.
2. Use a certificate from a CA whose root certificate is not in the file (a unknown CA).
3. Use a self-signed certificate.

3.4.1.1 Obtain a certificate from a CA

To use a certificate from one of the well-known CAs or some other CA, you must request the certificate from the CA, receive the certificate, then store it into the key ring database, HODServerKeyDb.kdb file. Nothing else needs to be done to allow the Host On-Demand client to recognize the signer of certificates from the following CAs:

- RSA Data Security, Inc.
- VeriSign, Inc.
- Thawte Consulting

However, if the certificate is from any CA other than one of these well-known CAs that Host On-Demand recognizes, then the public certificate of that signer must be made available to the client in the CustomizedCAs.class file. Refer to 3.11.1, “Downloaded and cached clients” on page 113 for instructions on this process, or if you are using the Microsoft Internet Explorer 5.0 or above browser, you may authenticate the server’s certificate from the list of CAs that Microsoft recognizes. For further details refer to 3.11.3, “Using a cryptographic database” on page 114.

3.4.1.2 Self-Signed Certificate

If your security requirements do not warrant the purchase of a commercial certificate, if you need a temporary certificate while you are waiting for your permanent certificate, or if you need one just for testing, you can create your own (self-signed) certificate by using the key management utility to create a self-signed certificate. Refer to 3.10.1, “Creating a self-signed certificate” on page 109 for details.

3.4.2 Configuring the Host On-Demand Redirector

Use the sample configuration shown in Figure 46 to illustrate configuring the Redirector.

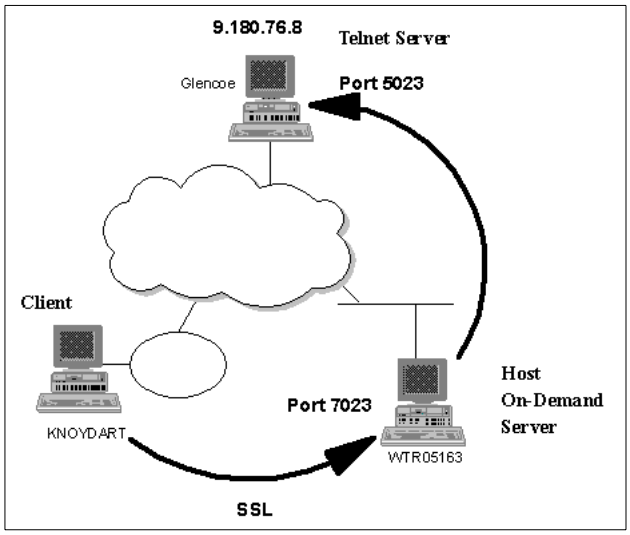

Figure 46. Implementing SSL on the Host On-Demand Redirector

The client-side connection will be SSL enabled and connect to the Redirector over port 7023. The server-side connection will connect to the Telnet Server, Glencoe, over port 5023. The host-side connection will not support SSL, as would be required if Glencoe were a UNIX server or other system supporting VT emulation.

Sign on as the administrator and select the **Redirector Service** option in the navigation frame on the left to display the window shown in Figure 47.

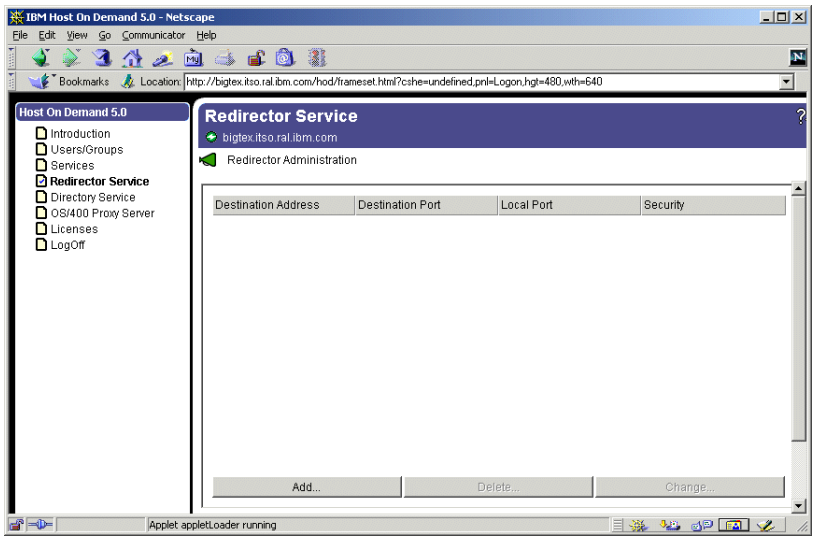

Figure 47. Redirector service

To add an entry, click **Add**. The Add Configuration window shown in Figure 48 will appear. In this window you must enter the destination address and port number of the target Telnet server, Glencoe. In addition, you must select the port on the Redirector to which the client will communicate and the security level required, client-side.

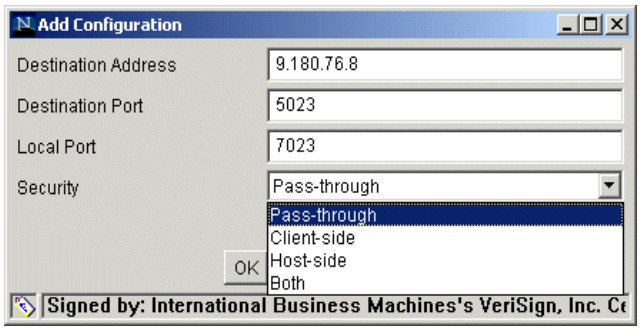

Figure 48. Redirector configuration

Finally, click **OK** to save and enable the configuration. You do not need to stop/restart the Redirector service or the Service Manager.

3.5 The OS/400 proxy server

The OS/400 MOD 3 toolbox delivered the OS/400 proxy server to Host On-Demand. The OS/400 proxy server is a service that runs on the Host On-Demand server and provides the ability of a Database On-Demand client and OS/400 file transfer to operate through a single port rather than the standard multiple-port implementation.

Figure 49 illustrates how you may select the port you wish to use for the proxy server. The default OS/400 proxy server port is 3470.

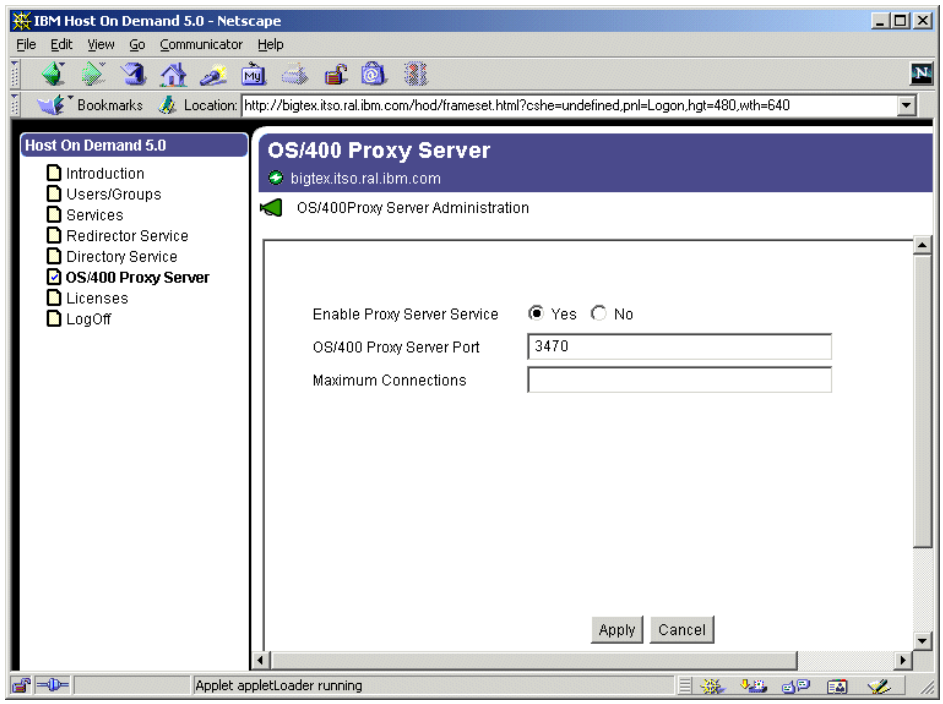

Figure 49. OS/400 proxy server

For information on how to configure and use the TN5250 file transfer client with the OS/400 proxy server, refer to 3.2.6, “TN5250 client” on page 61. For information on how to configure and use the Database On-Demand client with the proxy server, refer to 3.2.8, “AS/400 Database On-Demand client” on page 62.

3.5.1 OS/400 proxy limitations

When SSL is enabled, server authentication is used for encrypting data between the client and the host for transferring files to an AS/400. However, if you enable both SSL and the OS/400 proxy server, encryption is done only from the proxy server (Host On-Demand server) to the host (AS/400). Data is not encrypted on the client side (from the client to the proxy server).

If you are trying to avoid opening additional ports on the firewall by using the configuration servlet, you still must enable the configured proxy server port for 5250 file transfer and Database On-Demand.

Transferring save files (SAVF) is not supported with the OS/400 proxy server enabled.

3.6 Secure OS/400 Database On-Demand and file transfer configuration

Host On-Demand supports a secure Database On-Demand connection with an AS/400. In addition, the OS/400 file transfer capability within the TN5250 client may also transfer data over a secure connection. The OS/400 Database On-Demand client and the S/400 FTP file transfer component of the TN5250 client obtain use the KeyRing.class (refer to 3.2.1, “Java class files” on page 49) file to authenticate supported servers.

3.6.1 Updating the KeyRing.class file

You need to do no further configuration if your AS/400 has a certificate from one of the supported CAs:

- VeriSign, Inc.
- Integrion Financial Network
- IBM World Registry
- Thawte Consulting
- RSA Data Security, Inc.

If your AS/400 is using a certificate from an unknown CA or a self-signed certificate, use the following procedure for updating the KeyRing.class file and making it available to clients. This procedure is to be followed from the Host On-Demand server or the locally installed client.

1. From a command prompt, change to your Host On-Demand lib directory:

   ```
   cd /usr/local/hostondemand/lib
   ```

2. Run:

   ```
   keyrng com.ibm.as400.access.KeyRing connect <systemname>:<port>
   ```

where <systemname> is the address of AS/400 server you wish to connect to, and <port> is the secure port number on that server. For example if your secure server is 9.24.104.162 and the secure port is 9476 (the default port), you would enter the following string:

```
keyrng com.ibm.as400.access.KeyRing connect 9.24.104.162:9476
```

3. Type `toolbox` as a password.
4. Select the number of the Certificate Authority (CA) certificate that you want to add to your AS/400. Be sure to add the CA certificate and not the site certificate. A message is issued stating that the certificate is being added to com.ibm.as400.access.KeyRing.class.
5. Repeat step 2 for each certificate that you would like to add, downloading a separate certificate for each CA you would like to add to the KeyRing.class file.
6. When all certificates are loaded, copy the KeyRing.class file to the \hostondemand\hod\com\ibm\as400\access directory.

3.7 The configuration servlet

The traditional technique for retrieving and saving user preferences is for the Host On-Demand client to talk directly to the Host On-Demand configuration server via a predefined port, 8999 by default. Although efficient, it has two drawbacks when used in an environment that demands security:

1. It requires an additional port to be opened through a firewall.
2. The data is not encrypted, but flows in the clear.

To resolve these issues, IBM WebSphere Host On-Demand Version 5 introduced a servlet to tunnel the configuration information between the client and the servlet over an HTTP(S) connection, and then to pass that information on to the Host On-Demand configuration server of choice over the defined configuration port. This resolves both of the above-mentioned issues by using the existing HTTP(S) port already open through the firewall, and the encryption of the data by using HTTPS.

The implementation of the configuration servlet requires either a Web server that can manage servlets, such as Lotus Domino Go Server, or a Web application server such as WebSphere Application Server.

There are many products that are capable of running the configuration servlet, and the configuration procedure for each is different. However, a procedure for configuring the IBM WebSphere Application Server Version 3.5

running on Windows NT is provided. Note that the windows and navigation instructions may be different if you are using a different release level of WebSphere Application Server.

3.7.1 Enabling clients

There are two ways to enable a client to use the configuration servlet:

1. Set the ConfigServerURL parameter in the config.properties file. When this parameter is detected in the config.properties file, all clients, including the administration client, will use this method of communication with the Host On-Demand configuration server. Figure 50 shows a sample file.

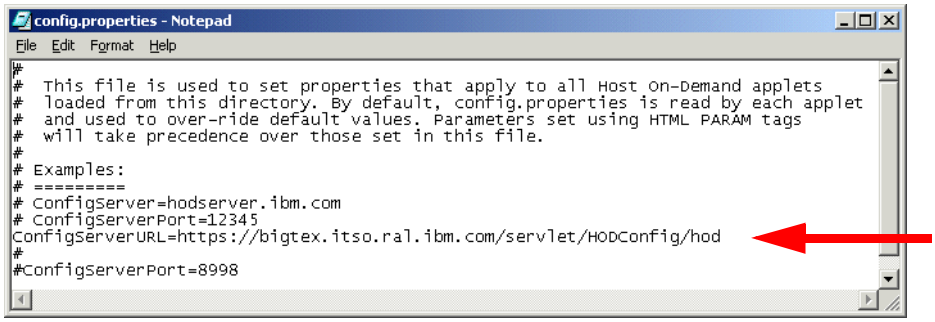

Figure 50. Config.properties file

2. Set the ConfigServerURL parameter in the HTML file used to launch the IBM WebSphere Host On-Demand client. This technique allows the administrator to specify which clients use the configuration servlet, such as external users, and which clients use the configuration server directly, such as internal users. Figure 51 illustrates how to specify the parameter in the Deployment Wizard to communicate securely with a specific server.

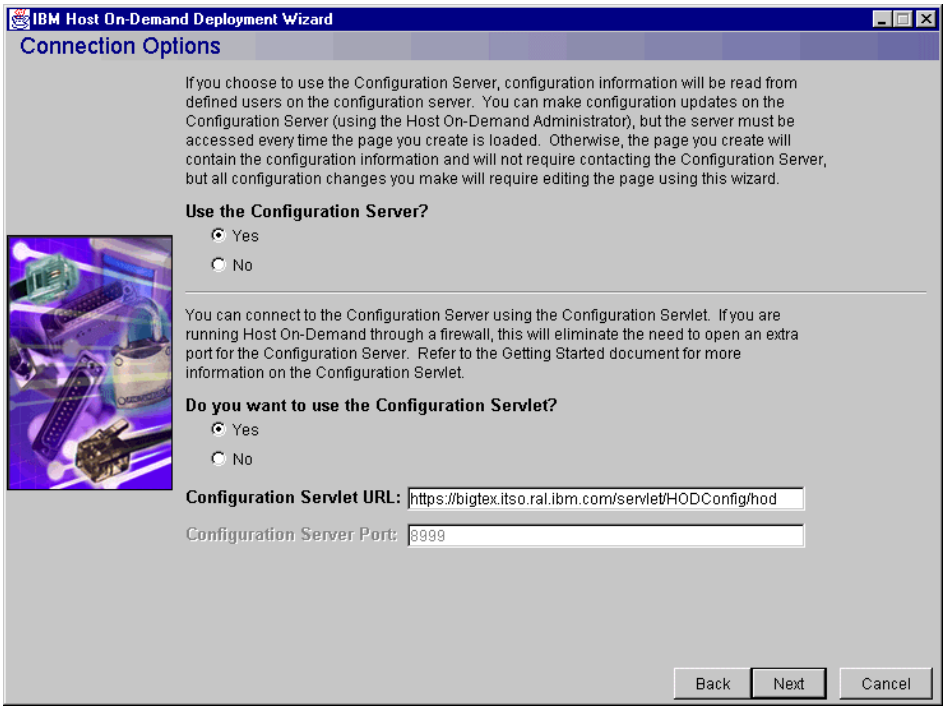

Figure 51. Configuration servlet URL - Deployment Wizard

3.7.2 Accessing the configuration servlet

There are two ways to specify the location of the configuration servlet: the direct reference and the indirect reference.

3.7.2.1 Direct reference

The direct reference is a complete URL. It includes the HTTP or HTTPS designation. For example, if you specify:

```
https://hodserver.raleigh.ibm.com/servlet/HODConfig
```

you force the applet to use an encrypted HTTP connection to contact the Host On-Demand configuration servlet running on hodserver.raleigh.ibm.com over the default port 443. If this reference is used, the configuration servlet information will flow over an encrypted session even if the URL used to load the Host On-Demand client specified an unencrypted session, for example `https://hodserver.raleigh.ibm.com/hod/HOD.html`.

3.7.2.2 Indirect reference

An indirect reference specifies only a path name on the server that launched the Host On-Demand client. Using this method results in the ConfigServerURL being appended to the host portion of the Host On-Demand applet's URL. For example, if the configuration servlet reference is:

```
/servlet/HODConfig
```

and the Host On-Demand applet is loaded using the following URL:

```
https://hodserver/hod/HOD.html
```

then the resulting URL used to contact the configuration servlet would be:

```
https://hodserver/servlet/HODConfig/hod
```

This method is more flexible, allowing the reference to be used for HTTP and HTTPS connections from a single specification.

3.7.3 Configuring the configuration servlet

WebSphere Application Server installs a default servlet engine. An example of how to install the configuration servlet using the graphical interface on a Windows platform, and how to install the servlet using the XMLConfig utility that was introduced with WebSphere Application Server V3.5, are both provided. The graphical interface scenario shows how to install the configuration servlet running under the default servlet engine. For the scenario using the XMLConfig utility, the sample provided defines a new application to host the configuration servlet.

3.7.3.1 IBM WebSphere graphical configuration

Open the WebSphere Administrator's Console by selecting **Start -> Programs -> IBM WebSphere -> Application Server V3.5 ->Administrator's Console**. Once the Administrator's Console is up, you will see an icon that contains the name of your server. This is your node name. You must expand that tree by clicking the **+** sign and then expand the Default Server and the Default Servlet Engine icons.

Set WebSphere alias

A WebSphere Application Server can provide a platform for multiple hosts. Each of these hosts is represented by a virtual host name and a list of one or more DNS aliases by which it is known. When a servlet request is made, the server name and port number component of the URL is compared to a list of all known aliases in an effort to locate the correct virtual host and serve the servlet. If no match is found, an error is returned to the browser. When no port

number is specified in the URL, port 80 is assumed. If you will use any port other than port 80, including port 443 for HTTPS, you must add an alias statement with that port number specified.

There are several conditions that may not be obvious that will require you to add an alias:

If your URL specifies a port number, then you must define an alias that includes the port number.

If you will use HTTPS to connect with your WebSphere Application Server, you must define an alias with the port number that HTTPS is using, even if you are using the default port of 443.

If your Web server is host for multiple IP addresses, each IP address must have an alias and appropriate port number(s).

Let us illustrate with the environment shown in Figure 69. Assume the Web server has two network cards and two addresses (meaning two virtual hosts) and either address may use HTTP and HTTPS. The internal address is 9.24.105.38 (bigtex.itso.ral.ibm.com), and the external address is 205.223.100.15.

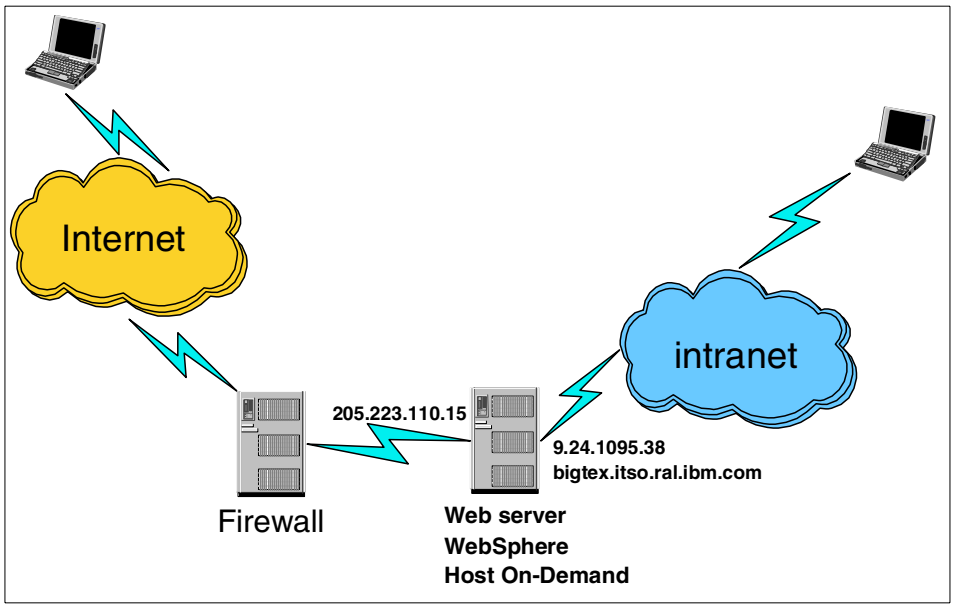

Figure 52. WebSphere alias environment

Table 1 illustrates the required alias rules.

Table 1. WebSphere alias examples

Reference URL	Required alias
http://127.0.0.1/servlet/HODConfig (usable only from the WebSphere machine)	127.0.0.1
http://localhost/servlet/HODConfig (usable from the WebSphere machine)	localhost
http://bigtex.itso.ral.ibm.com/servlet/HODConfig	bigtex.itso.ral.ibm.com
https://bigtex.itso.ral.ibm.com/servlet/HODConfig	bigtex.itso.ral.ibm.com:443
http://bigtex/servlet/HODConfig	bigtex
https://bigtex/servlet/HODConfig	bigtex:443
http://9.24.105.38/servlet/HODConfig	9.24.105.38
http://205.223.100.15/servlet/HODConfig	205.223.100.15
https://205.223.100.15/servlet/HODConfig	205.223.100.15:443

If the Web server is properly configured for all the connections and ports prior to the installation of the WebSphere Application Server, the WebSphere Application Server will add all the appropriate aliases; however, if anything changes, you must update the aliases manually.

To set the required aliases you must first select **default_host** then select the **Advanced** tab as shown in Figure 53 on page 90. Next, scroll down to an empty alias field and from there enter the required alias. Repeat this process until all aliases are entered, then click **Apply**.

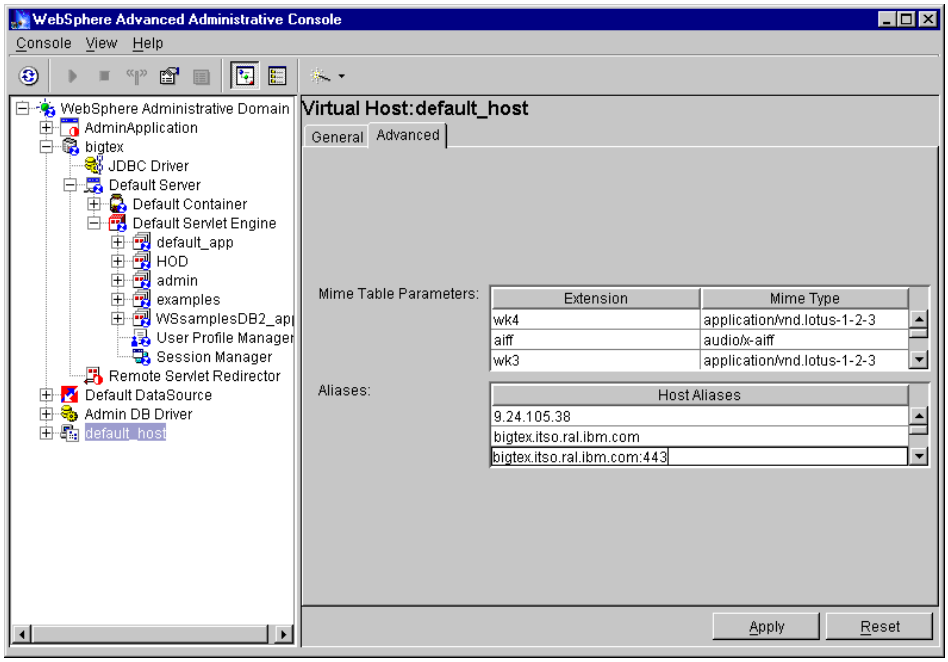

Figure 53. WebSphere default_app alias

Adding the classpath

You must now add an entry to the classpath for the application that hosts the configuration servlet, default_app in this example. Select the **default_app** entry in the left-hand pane, then select the **Advanced** tab from the resulting right-hand pane (see Figure 54).

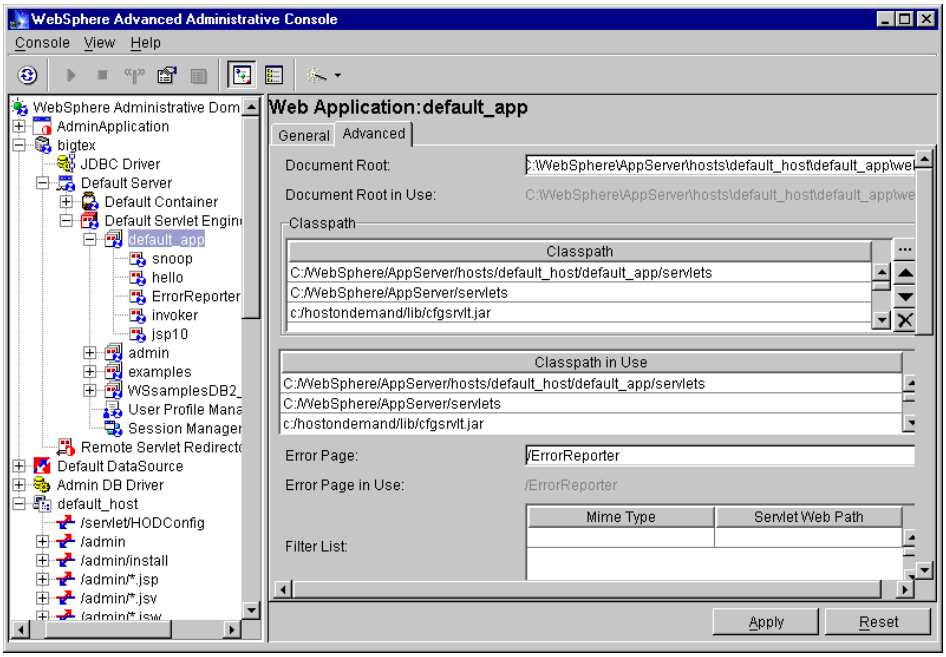

Figure 54. Select default application

You will see a frame called Classpath. This is where you will add the location of the cfgsrvlt.jar file. Select one of the empty entry boxes under Classpath and type the location of the cfgsrvlt.jar file, for example C:\hostondemand\lib\cfgsrvlt.jar. You must click **Apply** to update the classpath.

Adding the servlet

The next step is to add the IBM WebSphere Host On-Demand configuration servlet, so select the default application using the right mouse button to display the context menu. From that menu, click **Create** to display the next context menu, where you then click **Servlet** (see Figure 55).

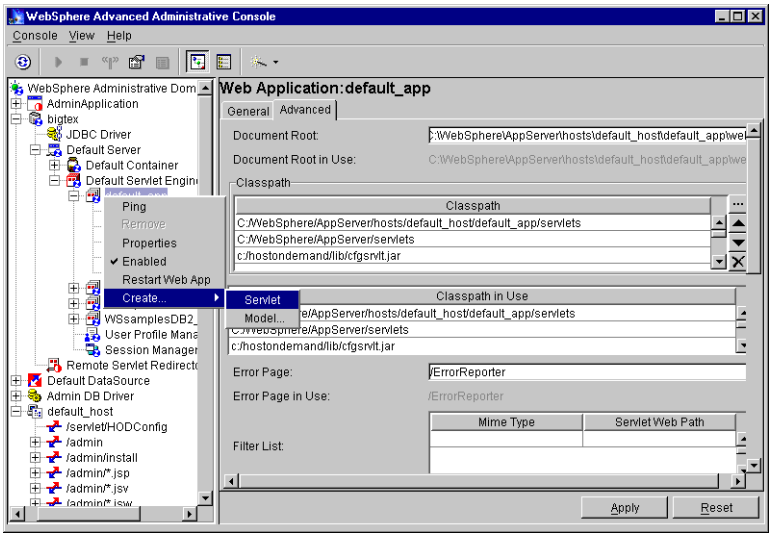

Figure 55. Create servlet, step 1

The resulting window is shown in Figure 56.

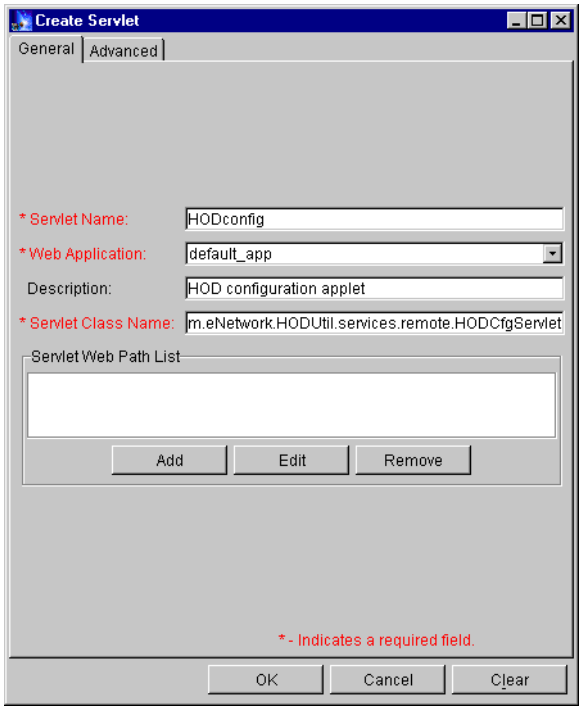

Figure 56. Create servlet, step 2

In this window there are three required fields and one optional field:

- Servlet Name: the name of the servlet as it will be known to WebSphere (required).
- Web Application: the name of the Web application will be displayed (required).
- Description: a textual description of the servlet, for example Host On-Demand configuration servlet (optional, but recommended).
- Servlet Class Name: the full name of the class for this servlet. This is a required field and the value must be `com.ibm.eNetwork.HODUtil.services.remote.HODCfgServlet;`
- Servlet Web path list: the string that will be used in the URL to identify the servlet.

The servlet name may be any name you wish to use in your URLs; the example used here is HODConfig. The Description field is optional and is used only as comments. The Servlet Class Name field is critical and must be specified exactly. It is recommended that you cut and paste it directly from the

help file. The value is
`com.ibm.eNetwork.HODUtil.services.remote.HODCfgServlet`.

Lastly, you must click **Add**, which will display the Add Web Path to Servlet window (see Figure 57). Here you enter the alias of this servlet, for example /servlet/HODConfig. Note the complete string:

```
/servlet/HODConfig
```

This will be the value that must be specified in the URL when accessing the configuration servlet. To leave this window, click **OK** to return to the previous window (Figure 56) where you must then click **OK**.

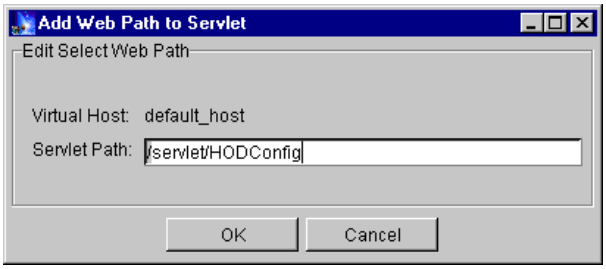

Figure 57. Create servlet, step 3

Upon return to the main Create Servlet window, you must select the **Advanced** tab. This will display the window shown in Figure 58 on page 95 into which you may specify the parameters as described in Table 2 on page 95. These parameters are optional. You need to specify them only if they differ from the defaults shown in Table 2. It is recommended that you specify at least the ConfigServer, ConfigServerPort, and the ShowStats parameters as shown in Figure 58.

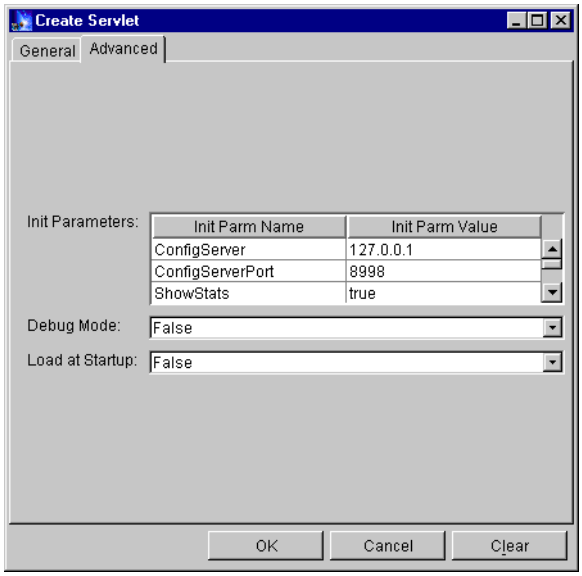

Figure 58. Servlet parameters

Specifying the ShowStats as true is recommended so that you can easily verify if the servlet is working properly before deploying the servlet.

Once all windows are completed, click **Finished**.

The IBM WebSphere Host On-Demand applets will recognize the parameters shown in Table 2, which are specified in the definition of the configuration servlet.

Table 2. Configuration servlet parameters

Parameter	Default Values	Description
ConfigServer	127.0.0.1	Host name or address of the Host On-Demand configuration server.
ConfigServerPort	8999	Port Number of the Host On-Demand configuration server. This must match the port on which the target configuration server is listening.
Trace	false	When set to true, the configuration servlet writes servlet messages to the servlet engine log file, and to the browser when requested, for debugging purposes.

Parameter	Default Values	Description
ShowStats	false	When set to true, this option allows the configuration servlet to return configuration information and statistics to browser requests. To invoke this option, specify *info* as the parameter passed to the applet. See 3.7.3.3, “Testing the servlet” on page 97.
BufferSize	4096	Size of the buffer to use on buffered input or output streams.
PoolSize	5	Size of the buffer or socket pool to maintain. To turn off pooling, set PoolSize to 0.

3.7.3.2 Start the default server

To enable the configuration servlet, the default server must be stopped and started. To stop the default server, highlight it and either click **Stop** on the tool bar, or click the right mouse button to display the context menu and click **Stop**. You must wait until you receive the information window shown in Figure 59, indicating that the server has stopped. This process could take a while.

Figure 59. Default server stopped

Click **OK** to clear the information window, then from the context menu of the default server click **Start** to start the server. Again, you must wait for the information window (shown in Figure 60) for the process to complete.

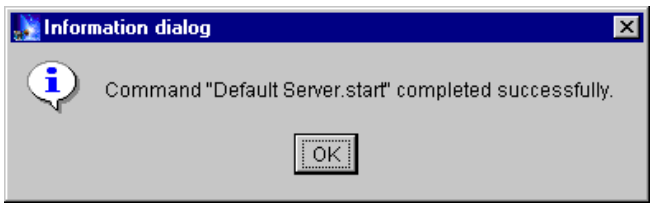

Figure 60. Default server started

3.7.3.3 Testing the servlet

After restarting the default server, it is recommended that you test the servlet by invoking the ShowStats function. This is done by specifying the following URL from a browser:

```
http://server_name/servlet_location/HODConfig/info
```

Using the example just created, the URL will look like the following:

```
http://server_name/servlet/HODConfig/info
```

When successful, your browser will return a window similar to that shown in Figure 61.

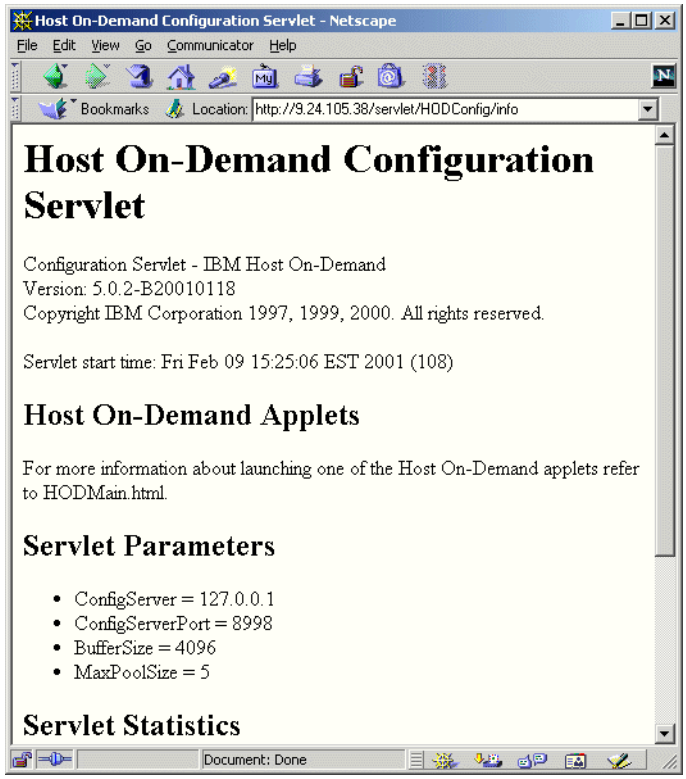

Figure 61. Servlet information

3.7.3.4 XMLConfig Utility

If you are using WebSphere Application Server 3.5 there is a batch utility, XMLConfig, that may be used to add the configuration servlet. This utility is available on all platforms and is located in the \AppServer\bin directory. To

use the utility you must create an XML file that defines the changes that you wish to implement. The general syntax is to invoke the utility:

```
XMLConfig -import filename.xml
```

A complete description of how to use the XMLConfig utility may be found in Chapter 21 of the *WebSphere V3.5 Handbook,* SG24-6161. The remainder of this section provides sample XML files to add a configuration servlet to the server.

Add configuration servlet to default_app

This example configures the configuration servlet to run under the default_app. The objective is to define aliases to allow secure connections to the configuration servlet, and to add the configuration servlet under the default_app. The result will allow you to specify one of the following URLs to access the servlet:

- /servlet/HODConfig (a relative URL may be used with HTTP or HTTPS)
- http://bigtex.itso.ibm.com/servlet/HODConfig
- https://bigtex.itso.ibm.com/servlet/HODConfig

Note that even though the port number is not specified in the URL, it is still required in the definition.

The XML input file is shown below:

```
<?xml version="1.0"?>
<!DOCTYPE websphere-sa-config SYSTEM
"$server_root$$dsep$bin$dsep$xmlconfig.dtd" >

<websphere-sa-config>
  <virtual-host name="default_host" action="update">

   <alias-list>
      <alias>localhost</alias>
      <alias>127.0.0.1</alias>
      <alias>bigtex</alias>
      <alias>bigtex.itso.ral.ibm.com</alias>
      <alias>9.24.105.38</alias>
      <alias>bigtex:443</alias>
      <alias>bigtex.itso.ral.ibm.com:443</alias>
      <alias>9.24.105.38:443</alias>
    </alias-list>
  </virtual-host>
  <node name="bigtex" action="locate">
    <application-server name="Default Server" action="locate">
      <servlet-engine name="Default Servlet Engine" action="locate">
```

```
        <web-application name="HOD" action="create">
          <description>Host On-Demand</description>

<document-root>C:\WebSphere\AppServer\hosts\default_host\HOD\web</document
-root>
          <classpath>
            <path
value="C:/WebSphere/AppServer/hosts/default_host/HOD/servlets"/>
            <path value="c:/hostondemand/lib/cfgsrvlt.jar"/>
          </classpath>
          <error-page>/ErrorReporter</error-page>
          <filter-list/>
          <group-attributes/>
          <auto-reload>true</auto-reload>
          <reload-interval>9000</reload-interval>
          <enabled>true</enabled>
          <root-uri>default_host/</root-uri>
          <shared-context>false</shared-context>

<shared-context-jndi-name>SrdSrvltCtxHome</shared-context-jndi-name>

          <servlet name="HODConfig" action="create">
            <description>Configuration Servlet</description>

<code>com.ibm.eNetwork.HODUtil.services.remote.HODCfgServlet</code>
            <init-parameters>
              <parameter name="ConfigServerPort" value="8998"/>
              <parameter name="ShowStats" value="true"/>
              <parameter name="Trace" value="true"/>
              <parameter name="ConfigServer" value="127.0.0.1"/>
            </init-parameters>
            <load-at-startup>false</load-at-startup>
            <debug-mode>false</debug-mode>
            <uri-paths>
              <uri value="/HODConfig"/>
              <uri value="/hodconfig"/>
            </uri-paths>
            <enabled>true</enabled>
          </servlet>

        </web-application>
      </servlet-engine>
    </application-server>
  </node>
</websphere-sa-config>
```

Notes

It is always recommended that you back up (export) the existing WebSphere Application Server definition prior to proceeding.

It was discovered during testing that for the alias list, the actual action performed was to replace, when an update was specified. Therefore, it is recommended that you copy the virtual host definiitions form your exported backup file, paste them into the new file, and then add any additional aliases you require.

Add configuration servlet to new application

This scenario defines a new application, HOD, to run under the Default Servlet Engine, and to define the configuration servlet to run under the new application, HOD. Aliases are also defined to allow secure connections to the configuration servlet as was done in the scenario in “Add configuration servlet to default_app” on page 98. The result is the same except that the URL will be similar to one of the following:

- /HOD/HODConfig
  - http://bigtex.itso.ral.ibm.com/HOD/HODConfig
  - https://bigtex.itso.ral.ibm.com/HOD/HODConfig

The XML file used is as follows:

```
<?xml version="1.0"?>
<!DOCTYPE websphere-sa-config SYSTEM
"$server_root$$dsep$bin$dsep$xmlconfig.dtd" >

<websphere-sa-config>
  <virtual-host name="default_host" action="update">

   <alias-list>
      <alias>localhost</alias>
      <alias>127.0.0.1</alias>
      <alias>bigtex</alias>
      <alias>bigtex.itso.ral.ibm.com</alias>
      <alias>9.24.105.38</alias>
      <alias>localhost:443</alias>
      <alias>127.0.0.1:443</alias>
      <alias>bigtex:443</alias>
      <alias>bigtex.itso.ral.ibm.com:443</alias>
      <alias>9.24.105.38:443</alias>
    </alias-list>
  </virtual-host>
```

```
  <node name="bigtex" action="locate">
    <application-server name="Default Server" action="locate">
      <servlet-engine name="Default Servlet Engine" action="locate">
        <web-application name="HOD" action="update">
          <description>Host On-Demand</description>

<document-root>C:\WebSphere\AppServer\hosts\default_host\HOD\web</document
-root>
          <classpath>
            <path
value="C:/WebSphere/AppServer/hosts/default_host/HOD/servlets"/>
            <path value="c:/hostondemand/lib/cfgsrvlt.jar"/>
          </classpath>
          <error-page></error-page>
          <filter-list/>
          <group-attributes/>
          <auto-reload>true</auto-reload>
          <reload-interval>9000</reload-interval>
          <enabled>true</enabled>
          <root-uri>default_host/HOD</root-uri>
          <shared-context>false</shared-context>

<shared-context-jndi-name>SrdSrvltCtxHome</shared-context-jndi-name>
          <servlet name="HODConfig" action="update">
            <description>HOD Configuration Servlet</description>

<code>com.ibm.eNetwork.HODUtil.services.remote.HODCfgServlet</code>
            <init-parameters>
              <parameter name="ConfigServerPort" value="8998"/>
              <parameter name="ShowStats" value="true"/>
              <parameter name="ConfigServer" value="127.0.0.1"/>
              <parameter name="Trace" value="true"/>
            </init-parameters>
            <load-at-startup>false</load-at-startup>
            <debug-mode>false</debug-mode>
            <uri-paths>
              <uri value="/HODConfig"/>
              <uri value="/hodconfig"/>
            </uri-paths>
            <enabled>true</enabled>
          </servlet>
        </web-application>
      </servlet-engine>
    </application-server>
  </node>
</websphere-sa-config>
```

3.8 License use tracking

To assist the administrator is monitoring and insuring that the company is in compliance with their licensing agreement, Host On-Demand has the ability to track the number of concurrently active clients. The methods used by Host On-Demand are called:

1. License use counting (LUC): tracking is performed by a Host On-Demand server.
2. License use management (LUM): tracking is performed by a license use management server.

When tracking is enabled, there are two primary security implications:

- All license use tracking communications is over unencrypted sessions.
- The protocol and port used is different depending upon which method of license use tracking is selected. Communication to a LUM server is via the HTTP protocol, while communication to a LUC server is to the Host On-Demand configuration server port using the Host On-Demand private protocol. The configuration servlet is not used.

3.8.1 License use administration

License use administration is configured as part of Host On-Demand administration and enabled when you select **Enable** in the License-Use Count (settings for clients) section, shown in Figure 62.

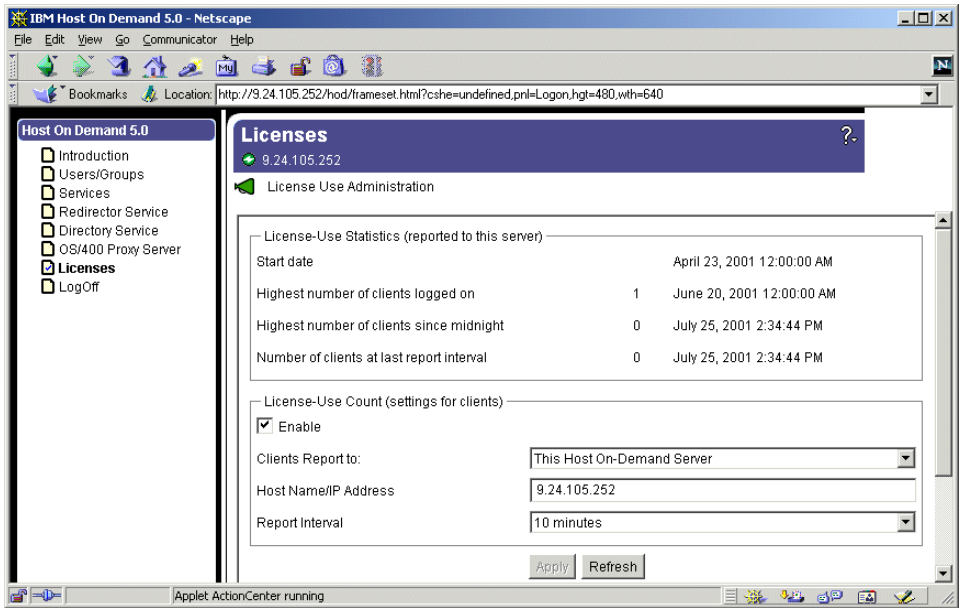

Figure 62. License use administration

The fields are as follows:

- Clients Report to

  This field specifies the location and type of License use count server. The choices are:

  - This Host On-Demand Server
  - Other Host On-Demand Server
  - This License Use Management Server
  - Other License Use Management Server

- Host Name/IP Address

  This field specifies the host name or IP address of the selected reporting server. If the server is on this machine, the field will be automatically completed and may not be altered. If the server is on another machine, you are required to enter the DNS host name or IP address of the other server.

  When this field specifies a server not on the machine being configured, the administrator may specify the port number to be used when communicating with the server, as follows:

- This Host On-Demand Server

  The port number that the client will use will be the default port (8999) if not overridden from the HTML file, or the config.properties file.

- Other Host On-Demand Server

  In addition to specifying the host name/IP address of the Host On-Demand server, you may specify the port number being used by that server. For example, if the server is hodserver.itso.ibm.com and the port being used is 8998, you would specify `hodserver.itso.ibm.com:8998` in the host name/IP address field.

- This License Use Management Server

  The server name used by the administrator's browser when accessing the administration utility is automatically inserted in the host name/IP address field. The administrator may not alter the contents of this field, and port 80 will be used.

- Other License Use Management Server

  If the Web server on the LUM server machine is listening on a port other than the standard port 80, the administrator must append the port number to the end of the host name/IP address. For example, if the Web server is lumserver.itso.ibm.com and the HTTP port being used is 8080, you would specify `lumserver.itso.ibm.com:8080` in the host name/IP address field.

- Reporting interval

  This field identifies how frequently the client will attempt to contact the reporting server. This field has no security impact.

3.8.2 Disabling license use tracking

You may disable the client from performing license use tracking activities in one of the two ways:

1. For those users who log into the Host On-Demand configuration server for preferences, license use tracking may be disabled by clearing the Enable check box on the Licenses pane of the administrator's console window (shown in Figure 62 on page 103).
2. Add the following parameter to the HTML page of any Host On-Demand client.

   ```
   <Param Name=Disable Value=LUM>
   ```

   Beginning with Host On-Demand Version 5.04, custom clients created with the Deployment Wizard may disable license use tracking by adding the

HTML parameters on the User-defined parameters window as shown in Figure 63. Using this technique with Host On-Demand Version 5.03 or earlier does not work, and you must add the HTML parameter to the HTML file manually.

When using the Deployment Wizard it is also recommended that you clear the License Use Management check box from the preload configuration window. This stops the downloading the class file(s) that perform the license use tracking. This step makes for a smaller client.

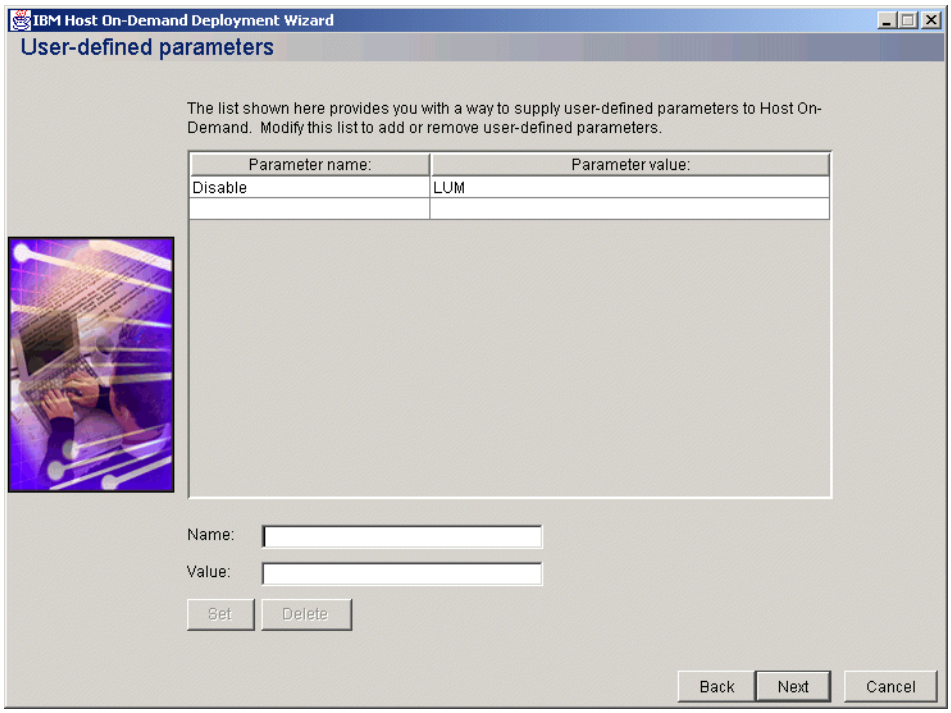

Figure 63. Disable license use tracking with Deployment Wizard

3.9 Administration

The administrator has complete control over the setup of the system, including user IDs and passwords and system definitions; therefore, only authorized personnel should have access to the administrative functions of Host On-Demand. Restricting access to the administrative functions of Host On-Demand is especially important when the server is located outside a secure area. Before discussing how to secure the administrative functions, you must first understand how administration works.

Administration of a Host On-Demand server is done through the administration applet, HODAdmin.html, which is downloaded into the browser and then communicates to the server over the defined configuration port either directly or via the configuration servlet, see 3.7, “The configuration servlet” on page 84. There are several items that are required for someone to perform administrative functions, and blocking access to any of them blocks access to the administrative functions.

1. One or more administrative accounts must be defined, and the user must supply the appropriate user ID and password.
2. The Host On-Demand Service Manager must be running on the server.
3. The user must have access to an HTML page in order to download the administrative functions.

To maintain secure communications, the administrator should access the applet, HODAdmin.html, using HTTPS, and communicate with the configuration server via a secure connection via the configuration servlet, see 3.7, “The configuration servlet” on page 84.

3.9.1 Administrator account

An administrator account is provided by default; the user ID is `admin` and the password is `password`. As an administrator, you can change either of these through the Add User window, but you cannot delete this account, nor can you create another account that has administrator privilege.

You can, however, create more administrator accounts (or recover the original) by using the following procedure:

1. Open the ConfigAgentParms data file, in the \hostondemand\private subdirectory of your server, with a text editor (see Figure 64):

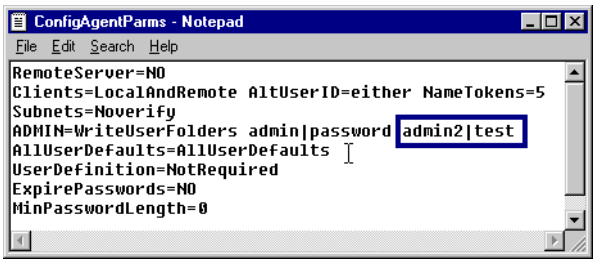

Figure 64. ConfigAgentParms data file

2. In the `ADMIN=WriteUserFolders` line, you will find `admin|password`

3. `admin` is the default administrator ID, `password` the default password. Even if you change the password, for example `to newpwd`, (through the administrator windows) you will see only the `admin|password` statement in the file.
4. If you use the administrator windows to change the admin name to, for example, `admin1`, you will find `admin1|` but no password. The new password is encrypted and written to another file.
5. To add an administrator, use an editor and type a second name and password for example, `admin2|test,` behind the first, separated by a blank, as shown in Figure 64.
6. Log off.
7. Stop and restart the Host On-Demand Service Manager.

Note

Changes you make to the ConfigAgentParms file will only be activated if you stop and restart the Service Manager.

8. Reload the Administrator page:
   - Shift + Reload (point and click with the mouse) with Netscape.
   - Ctrl + Refresh (point and click with the mouse) with Internet Explorer.
9. Log on, using the *new* administrator name and password (admin2 and test in our example).
   - You can now change the new administrator name and password through the normal Change User window. Once you do so, the new password will disappear from the ConfigAgentParms file.

Notes

Every time you want to change between your administrator IDs, you must log off and reload HODAdmin.html.

If you forget the password for the original administrator ID, you can use this procedure to re-create it.

3.9.1.1 Restrict access to Administrative functions

You also need to secure or restrict access to the administrative client. There are several ways that this may be done. Below is a list of the some of the techniques that may be used. These techniques may be used individually or in combinations for more security.

1. Change the password for the administrative account. You cannot delete the admin account; all you can do is change the password. Make the password non-trivial. Longer passwords that include alpha and numeric characters are more secure.
2. Use HTTPS to encrypt all administrative traffic and Web server authentication.
3. Use Web server authentication and authorization to restrict access to the Host On-Demand administrative client to only authorized people.
4. Use the configuration servlet, see 3.7, “The configuration servlet” on page 84, to encrypt all configuration information between the client and the Host On-Demand configuration server.
5. Use the Web server security functions to restrict launching of the administration client to the local machine. This will force the administrator to be physically present at the server in order to run the administrative applet.
6. Remove all administrative functions from any non-secure server. All administration should be performed on a server in a secure location, then transfer the changes to the servers in the DMZ or other non-secure environment.

3.10 Certificate Management Utility

At the time that this book was written, the only platforms on which the IBM Certificate Management Utility was distributed were Windows NT, Windows 2000 Server, and AIX. As a result, users on other platforms have no Host On-Demand utility for adding unknown CAs to the HODServerKeyDb.kdb file or the CustomizedCAs.class file. To update these files, the administrator is required to use the utility on a supported platform (including the locally installed client) or some other utility that can perform the same functions. There are plans to update this utility to run on other platforms in a future release.

The Certificate Management Utility on AIX is an X-Windows Java application. Before running the Certificate Management Utility on AIX you must be in the /hostondemand/bin directory, and the JAVA_HOME environment variable must be set to the full path of your Java installation. For example, if the default installation options were chosen and your Java system is installed in /usr/JDK1.1.8, you would run the Certificate Management Utility by doing the following:

```
cd /usr/opt/hostondemand/bin
```

```
export JAVA_HOME=/usr/JDK1.1.8
CertificateManagement
```

On a Windows system you start the Certificate Management Utility by selecting **Start -> Programs -> IBM Host On-Demand -> Administration -> Certificate Management**. The graphical interface on Windows and AIX is the same.

3.10.1 Creating a self-signed certificate

Follow these instructions to create a self-signed certificate:

1. Select **Key Database File - New**.
2. Accept **CMS key database file**.
3. Accept HODServerKeyDb.kdb for the file name and \hostondemand\bin for the location.
4. Click **OK**. The Password window appears.
5. Enter your password twice and, if you wish, set an expiration time.
6. Select **Stash the Password** to cause the password to be held, encrypted, in \hostondemand\HODServerKeyDb.sth. The Host On-Demand server needs to be able to access it at run time.
7. Click **OK**. Your Key Management window will look like Figure 65.

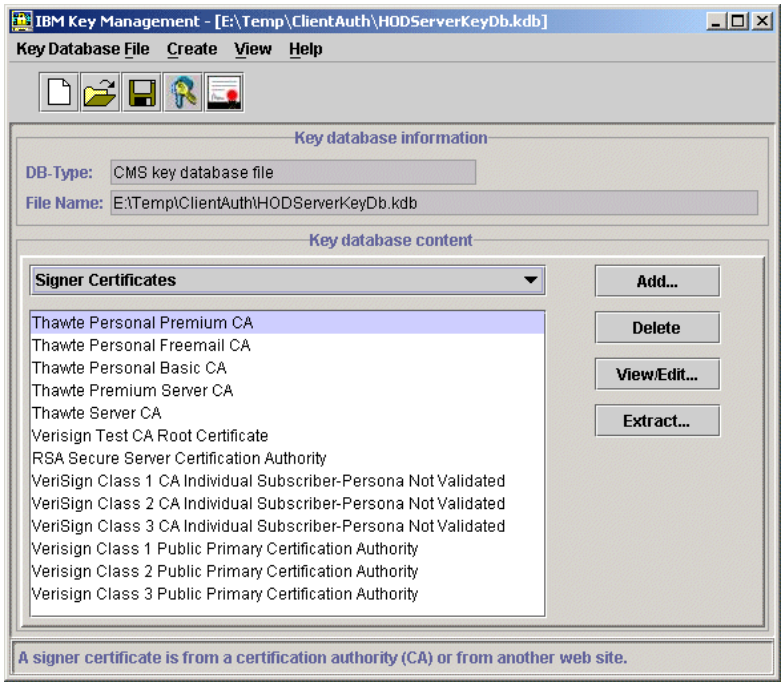

Figure 65. IBM Key Management Utility

8. Select **Personal Certificates** from the drop-down list.
9. Select **Create - New Self-signed Certificate**. The Create New Self-Signed Certificate window appears.
10. Use Table 3 to fill in the fields.

Table 3. Self-signed certificate information

Field name	Value
Key Label	This field is for identification use only, so use a meaningful text string.
Version	Use X509 V3.
Key Size	This should default to the encryption level of your installation. The larger value is more secure, but you must take into account the capabilities of your browsers to decrypt.

Field name	Value
Common Name	This should be the fully qualified DNS name of the server. It will be used if any client requests server authentication, and when resolved by the DNS server it must match the IP address the client uses to establish the session (refer to 3.2.3.2, “Server authentication” on page 58).
Organization	Fully identify the name of your organization, for example IBM Corp.
Organizational Unit	This optional field can further identify the server or department operating the server within the organization, for example ITSO.
Locality	This optional field should contain the city where the server is located
State/Province	This is an optional parameter.
Zipcode	This is an optional parameter. Some Netscape browser versions have been known to crash when this field is used; therefore, it is recommended to omit this field.
Country	This will default to the native country code.
Validity	The maximum recommended value is 365 days.

When complete, the window should look like Figure 66.

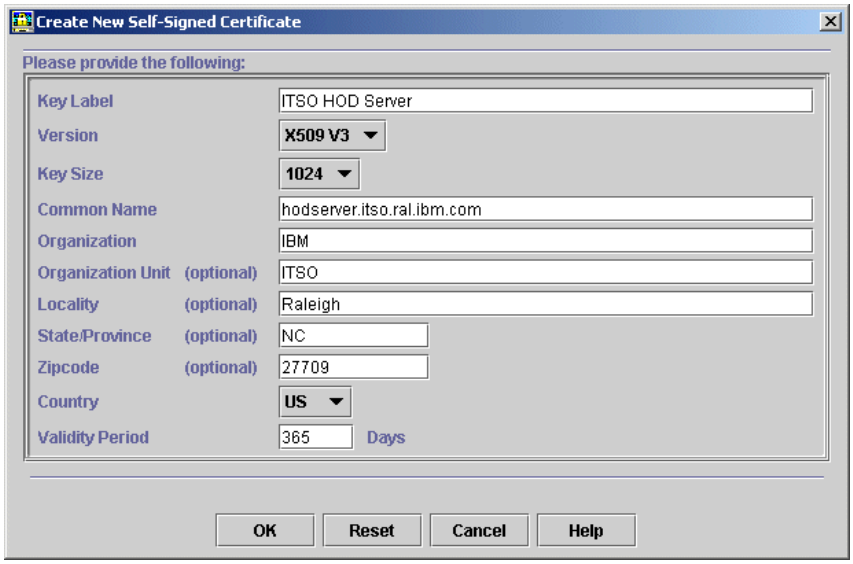

Figure 66. Create New Self-Signed Certificate window - completed

11. Click **OK**. Your certificate will appear in the list of Personal Certificates.
12. Use View/Edit to check that the values are correct and that the certificate is the default.
13. In order for the Host On-Demand service manager to use this new certificate, you must stop and restart the Host On-Demand service manager.

3.11 Making server certificates available to clients

All clients must be able to authenticate the signer of the server certificate. If the signer exists in the WellKnownTrustedCAs.class file, nothing more needs to be done, since all Host On-Demand clients on all platforms have access to this file either locally installed or downloaded from the server.

If you use a certificate from a CA that is not contained in the Host On-Demand WellKnownTrustedCAs.class file, or you use a self-signed certificate, you must provide the client with the signer's public certificate.

The traditional method is to add the signer certificate to the CustomizedCAs.class file. Download clients receive this file from the server when they load the applet, while locally installed clients must have this file either created or installed on each client separately.

Beginning with Host On-Demand Version 5.03, there is another method for Windows platform users: the cryptographic database, which contains many more signer certificates than does the WellKnownTrustedCAs.class file.

3.11.1 Downloaded and cached clients

The CustomizedCAs.class file is always downloaded to all download clients and available for use even if they are configured to use the cryptographic database. This insures that all clients will have the certificate when required regardless of platform.

The following illustrates the procedure for running the IBM Certificate Management Utility as executed on a Windows system. Once the Certificate Management Utility is launched (see 3.10, “Certificate Management Utility” on page 108) the procedure is the same for both the Windows and AIX Host On-Demand installations. The certificate management is not installed on any other operating environment. You may also create and update the CustomizedCAs.class file using the Windows locally installed Host On-Demand client and copy the file to the published directory of the server.

1. Start Key Management on the Host On-Demand server and open the key database file, HODServerKeyDb.kdb, which is located in the \hostondemand\bin directory.
2. Select **Personal Certificates** and highlight your self-signed certificate.
3. Select **Extract Certificate**, and the Extract Certificate to a File window appears.

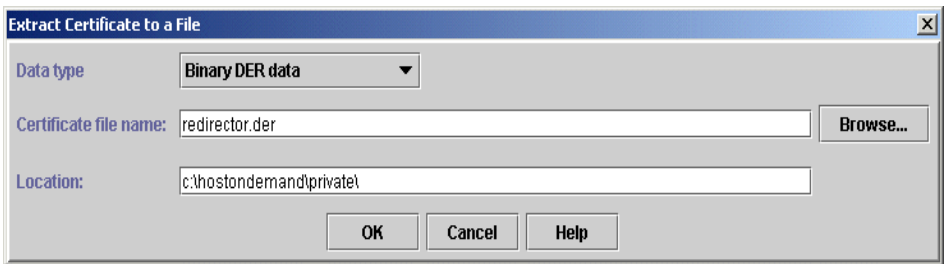

Figure 67. Extract certificate

4. Set the **Data type** to Binary DER.
5. Select a name and location to store this file, for example \hostondemand\HOD\private\redirector.der.
6. Click **OK** and the file will be created.
7. Close the key database file.

8. Open the CustomizedCAs.class file located in the Host On-Demand publish directory. If the file does not exist, create it.
9. Select **Signer Certificates**.
10. Click **Add**.

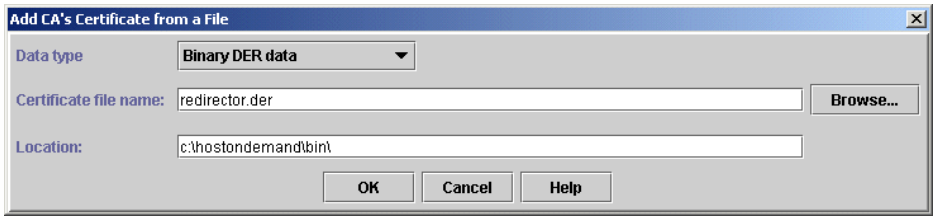

Figure 68. Add signer certificate

11. Select the **Binary DER** file from the Data type pull-down.
12. Enter the file name and location where you stored it, or you may click **Browse** to locate it.
13. Click **OK** to insert the certificate.
14. Close and save the CustomizedCAs.class file.

3.11.2 Locally Installed Clients

There are two ways to enable a locally installed client to recognize certificates signed by unknown CAs:

1. Build the CustomizedCAs.class file at the central site and transfer it to the client via a secure method that meets your security policy. The user needs only to store it in the \hostondemand\lib subdirectory of his system to make it available.
2. Update the locally installed client on the workstation. The administrator must transfer the binary DER file of the certificate to each user that has a locally installed client. The user of that machine must then run the Certificate Management Utility from the local Host On-Demand installation. For security purposes, it is recommended that the binary DER file and the password be sent via separate out-of-band methods.

3.11.3 Using a cryptographic database

The cryptographic database contains a much larger list of recognized CAs than does the Host On-Demand WellKnownTrustedCAs.class file. You may use the cryptographic database to authenticate server certificates. This

database may be used in addition to the WellKnownTrustedCAs.class and CustomizedCAs.class files.

If you are running on a Windows system and you are using certificates from CAs that are not in the WellKnownTrustedCAs.class file that exist in Microsoft's cryptographic database, then you may wish to use this method for authenticating the server to avoid administrative overhead. If your CA is not listed in the cryptographic database or you are using a self-signed certificate, you may add it to that database. Follow the instructions that are provided by the browser.

3.12 LDAP directory considerations

When LDAP directory services is enabled, the Host On-Demand user IDs and passwords stored in the directory are not encrypted; therefore, anyone with administrative access to the directory server could view the Host On-Demand user IDs and their passwords.

Host On-Demand does not support an SSL connection to the LDAP directory server. All communications will be in the clear; therefore if you use the LDAP directory server, the Host On-Demand server and the LDAP directory server should both reside in a secure network.

3.13 Using Host On-Demand with a firewall

If you are configuring Host On-Demand to go through a firewall, it is recommended that the firewall administrator open only those ports required for the clients to function. At a minimum, you will need to open the following ports:

1. HTTPS port

   This port will be used for downloading the applet and for obtaining configuration information via the configuration server.

2. Telnet ports

   There may be one or more ports open, depending upon the Telnet server requirements. These ports should allow SSL-encrypted session traffic.

3. OS/400 proxy server port

   If you are using Database On-Demand or TN5250 file transfer, you should utilize one or more OS/400 proxy server ports to pass all traffic over a single port per proxy server.

3.13.1 TCP/IP port used by Host On-Demand

Table 4 identifies all the ports that Host On-Demand will use in its default configuration. Remember, that many of the ports are configurable, such as the configuration server port and the OS/400 proxy server port. Use this table to determine how to configure your firewall.

Table 4. Ports used by Host On-Demand

Host On-Demand Function	Unsecure Port(s) used	Secure Port(s) used
3270 and 5250 Display Emulation	23 (Telnet) 80 (HTTP) 8999 (config server)[3]	992 (Telnet) 443(HTTPS)
3270 and 5250 Printer Emulation	23 (Telnet) 80 (HTTP) 8999 (config server)[3]	992 (Telnet) 443(HTTPS)
3270 File Transfer (IND$FILE)	23 (Telnet) 80 (HTTP)	992 (Telnet) 443(HTTPS)
5250 file transfer - SAVF file	80 (HTTP) 8999 (config server)[3] 21 (FTP)[4] >1024 (FTP)[4] 446 (drda)[4] 449 (as-svrmap)[4] 8470 (as-central)[1 2 4] 8473 (as-file)[1 4] 8475 (as-rmtcmd)[1 4] 8476 (as-signon)[1 4]	
5250 file transfer - database	80 (HTTP) 8999 (config server)[3] 446 (drda)[4] 449 (as-svrmap)[4] 8470 (as-central)[1 2 4] 8473 (as-file)[1 4] 8475 (as-rmtcmd)[1 4] 8476 (as-signon)[1 4]	
5250 file transfer - stream file	80 (HTTP) 8999 (config server)[1 2 4] 449 (as-svrmap)[4] 8470 (as-central)[1 2 4] 8473 (as-file)[1 4] 8476 (as-signon)1 4	443 (HTTPS) 9470 (as-central)[1 2 4] 9473 (as-file)[1 4] 9476 (as-signon)1 4

Host On-Demand Function	Unsecure Port(s) used	Secure Port(s) used
Host On-Demand Administration	80 (HTTP) 8999 (config server)[3]	443 (HTTPS)
Database On-Demand	80 (HTTP) 8999 (config server)[3] 449 (as-svrmap) [4] 8470 (as-central)[1 2 4] 8471 (as-database)[1 4] 8476 (as-signon)[1 4]	
License Use Count License Use Management (LUM)	8999 (config server)[3] 80 (HTTP)	

Notes: port numbers listed are the default values.
1. You can change the port numbers with the OS/400 `WRKSRVTBLE` command. The port numbers listed are the default values.

2. The port for as-central is used only if a code-page conversion table needs to be created dynamically (EBCDIC to/from unicode). This is dependent on the JVM and the locale of the client.

3. You can change the config server port.

4.These ports do not need to be opened on the firewall if you are using OS/400 proxy server support. You will need to open the default proxy server port 3470. You can change this port.

Chapter 4. IBM WebSphere Host Publisher security

There are three areas in the use of IBM WebSphere Host Publisher where security needs to be monitored:

1. Transferring the applications from the development environment, the IBM WebSphere Host Publisher Studio, to the server
2. Communicating between the end user (client) and the server
3. Communication from the server to the host application

Many IBM WebSphere Host Publisher applications will be created using static user IDs and passwords that will be managed within the Integration Object. Since these user IDs and passwords are stored in the clear within the Integration Objects, there is an opportunity for exposing these as they are transferred from the IBM WebSphere Host Publisher Studio to the IBM WebSphere Host Publisher server for deployment. 4.1.1, “Encrypted passwords” on page 120 discuss the technique designed to protect these user IDs and passwords while in transit.

For communications between the end-user client and the IBM WebSphere Host Publisher server, the Web browser and the Web servers may use the Secure Hypertext Transfer Protocol (HTTPS).

IBM WebSphere Host Publisher is enabled to support Secure Sockets Layer (SSL) which allows you to secure the upstream TN3270E or TN5250 sessions, thus providing server authentication and data encryption for the communication from the server to the host application. This process is documented in 2.4, “Secure Sockets Layer” on page 38.

IBM WebSphere Host Publisher uses the IBM WebSphere Host On-Demand Java Beans to provide host connection support. The current version of IBM WebSphere Host Publisher uses the Java Beans from Version 4 of Host On-Demand. Host Publisher Version 2 does not support Transport Layer Security (TLS) or Express Logon (ELF), introduced in Host On-Demand Version 5; however, Host Publisher Version 3.5 is including support for ELF.

4.1 Transferring applications to the server

Transferring files to a remote system uses the File Transfer Protocol (FTP). Therefore, you need FTP client support in the system where IBM WebSphere Host Publisher Studio is running, and FTP server support (daemon) on your target server. For cases where there is a firewall between the Host Publisher

Studio and the Host Publisher server you must enable the firewall to pass FTP traffic.

Most Integration Objects will contain imbedded user IDs and passwords, and the FTP transfer of these Integration Objects is not encrypted. IBM WebSphere Host Publisher provides a mechanism to protect this sensitive data when the Integration Objects are transferred to the Host Publisher Server. These passwords can be encrypted with a strong algorithm for better security if so required. This function is available in both the export and non-export versions of Host Publisher.

4.1.1 Encrypted passwords

The data protection scheme works as follows:

1. When using passwords in your database or host Integration Objects, you will be prompted to select the type of security for these fields during the *transfer-to-server* operation. The following options are available:
   - No security.

     Selecting this option, passwords will flow in the clear (unencrypted) and will not be protected.
   - Scramble data.

     Indicates that passwords will be scrambled only. A weak protection but it does not require a password.
   - Encrypt data.

     Password fields will be encrypted using the triple-DES (3DES) algorithm. It requires a case-sensitive password. **Note:** The password you provide for this operation is not stored anywhere so you will need to remember it in order to make it available for the decryption process when you deploy the applications.
2. If you selected the option to encrypt at least one field, during the transfer-to-server operation you will again be prompted to provide this password. The Host Publisher Studio transfer-to-server process will use this password to derive a 168-bit key and this key will be used to encrypt the user passwords.
3. When using this option, passwords are protected and are stored encrypted on the server in the XML configuration files.
4. When you deploy your Host Publisher applications, you must provide exactly the same password you entered in step 2. The password is case sensitive.

4.1.2 Encryption algorithm

Host Publisher uses triple-DES (3DES) to encrypt selected user passwords that you defined in the database and host Integration Objects using the Studio. For an explanation of the DES encryption algorithm refer to 2.1.1, “Symmetric encryption algorithms” on page 10.

4.2 Securing client sessions

The major security points that need to be addressed when a client talks to the application are:

1. User authorization for application access
2. Protection of the transmitted data

4.2.1 User authorization

The first step is to determine if you need to authenticate your users or not. Many implementations allow universal access to the application page, then build in a sign-in window to the application and use a back-end system to do the authentication. Other implementations authenticate with the Web server using the authentication mechanisms built into the Web server. One such method is basic authentication using a simple user ID and password, in conjunction with an access control list to determine if the user is known to the system, and then if the user has authorization to access the application.

Other authentication methods may be used, but it is beyond the scope of this book to examine all the authentication methods available on the market. The security administrator should select a method that is consistent with the security policy and technical ability of the customer and within the budget.

4.2.2 Data encryption

If you are planning to secure the client session, you need to be aware of the following:

1. Use the HTTPS protocol in the URL.
2. The Web server must also be enabled for SSL and digital certificates must be available.
3. If you are invoking the RIOServlet from a Java applet, the SSL support for your connection will be provided by your browser.
4. If you are running a Java stand-alone application, the XML requester supports SSL (on the default SSL port 443). In this case, you can use the

IBM sslight.jar package or the SSL support in Java applications from Sun, found at http://java.sun.com/products/jsse/.

4.3 Securing the host connection

In addition to securing the client-to-server session, you may also need to secure the server-to-host connection. Your security policy will dictate this requirement.

Host Publisher server uses the Host On-Demand Java Beans to provide the connection to the host system. In IBM WebSphere Host Publisher Version 2 the Host On-Demand Version 4 Java Beans are used, which provide the following security capability:

- SSL connections, with optional client and server authentication, are supported for TN3270E, TN5250 connections

Note: SSL support for Database On-Demand is not available.

4.3.1 Defining a secure host connection

In order to configure a Host Publisher application you have to do the following:

1. Add any unknown CA certificates to the CustomizedCAs.class file.
2. Create an application in Host Publisher Studio.
3. Transfer application to the Host Publisher server and deploy it.
4. Configure the IBM WebSphere Application Server.

4.3.1.1 Add unknown CA certificates to CustomizedCAs.class

To use the certificate signed by an unknown CA, Host Publisher must trust the CA. Therefore you have to embed the unknown CAs certificate within a Java class file and insert it into Host Publisher's CLASSPATH. Host Publisher Studio uses this class file, CustomizedCAs.class, to establish a secure connection while the Integration Object is being built. Host Publisher Server also uses this file to establish the secure connection when the Integration Object is invoked.

A utility, gencert.bat, is included with the Host Publisher Studio to convert a certificate file to a Java class file. You can also use this utility to add a certificate from an unknown CA into the list of trusted CAs by following these instructions:

1. Open gencert.bat in a text editor, and replace `--site` with `--ca` and save it as gencacert.bat.

   ```
   .\jdk\bin\java -classpath .;.\sm.zip;.\jdk\lib\classes.zip
   com.ibm.hodsslight.tools.keyrng CustomizedCAs add --ca %1%
   ```

   Note

   The `gencert.bat` command only works with binary X.509 certificates.

2. Type the following command:

   ```
   c:\HostPublisher\Studio\gencacert.bat cacertificate_file
   ```

   where `c:\HostPublisher` is the directory where you installed Host Publisher and `cacertificate_file` is the name of the certificate from the unknown CA.

3. The gencert.bat tool will request you to enter a password. To generate a proper certificate file, click **Enter** when prompted for a password.

   Note: *Do not* enter a password.

4. The result will be a class file called CustomizedCAs.class. You must have a copy of this file on the Studio and on the Server. On the Studio, copy this file into the Studio subdirectory (c:\HostPublisher\Studio, for example). On the server, copy this class file to the \HostPub\Common directory for WebSphere Application Server 2.0.3. For WebSphere Application Server 3.0.2, copy the file to the \HostPub\Server\production\beans directory. You can place it anywhere in the CLASSPATH, as long as WebSphere Application Server picks up the class file when processing a request for an Integration Object configured for SSL support. Host Publisher Server uses this file to establish the secure connection when the Integration Object is invoked.

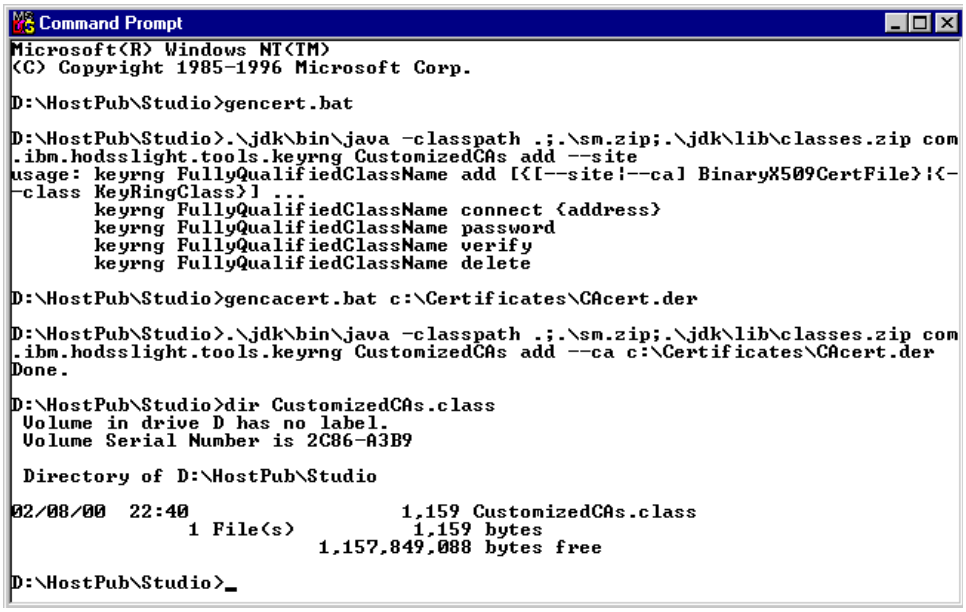

Figure 69. Gencert.bat

4.4 Sample scenarios

This section provides a scenario for a database and host Integration Objects. In both cases you select the option to protect (encrypt) the user passwords.

4.4.1 Database

A user ID and password are configured to access the database as illustrated in Figure 70.

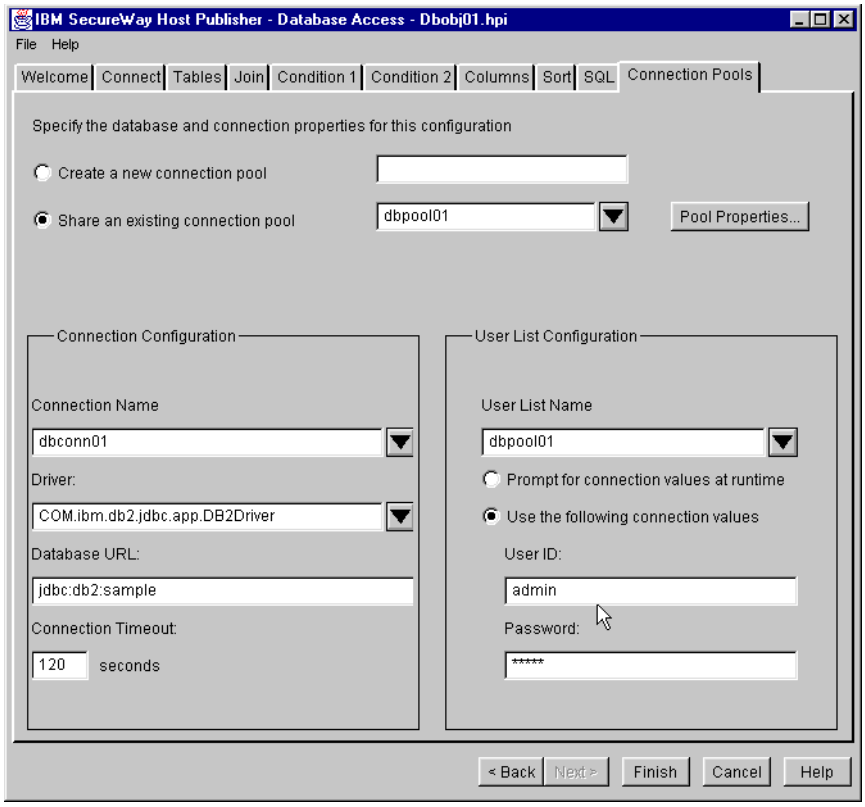

Figure 70. Database Integration Object configuration - user ID and password

During the transfer to server operation you are prompted to select the password security option for password fields as shown in Figure 71.

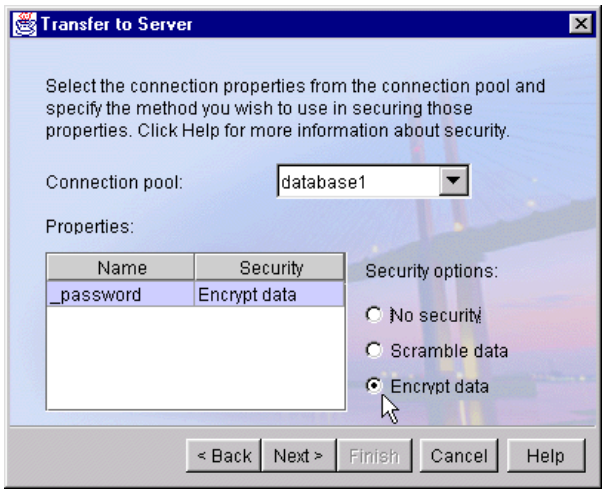

Figure 71. Selecting password encryption

If you selected encryption, you are now prompted to enter a password to derive the encryption key as illustrated in Figure 72.

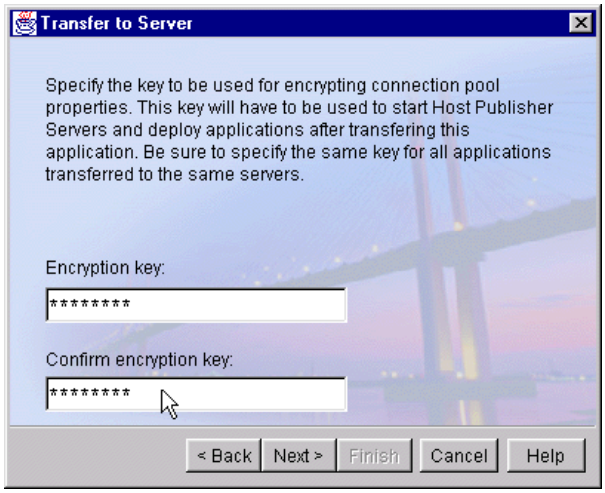

Figure 72. Providing a key for password encryption during transfer

Figure 73 shows the XML file for the userpool options and indicates the following:

- The defineproperty tag has the encrypt variable set to "2" indicating this field is encrypted.

- The value for the propertypassword has been encrypted in the userpool file for transfer to the staging subdirectory.
- In addition, since the encrypted value is a hexadecimal number, it has been translated into 64-ASCII code.

```
<schema>
  <defineproperty encrypt="0" name="_userid"/>
  <defineproperty encrypt="2" name="_password"/>
</schema>
<localuserpool name="database1" session="AAKL/ANuqXYcS/
  <entry key="admin">
    <property name="_userid" value="admin"/>
    <property name="_password" value="DXGnrOczGOA="/>
  </entry>
</localuserpool>
```

Figure 73. Encrypted Password in Userpool File

When the database application is deployed (see Figure 74), the same password is provided and the reverse process takes place to recover user passwords.

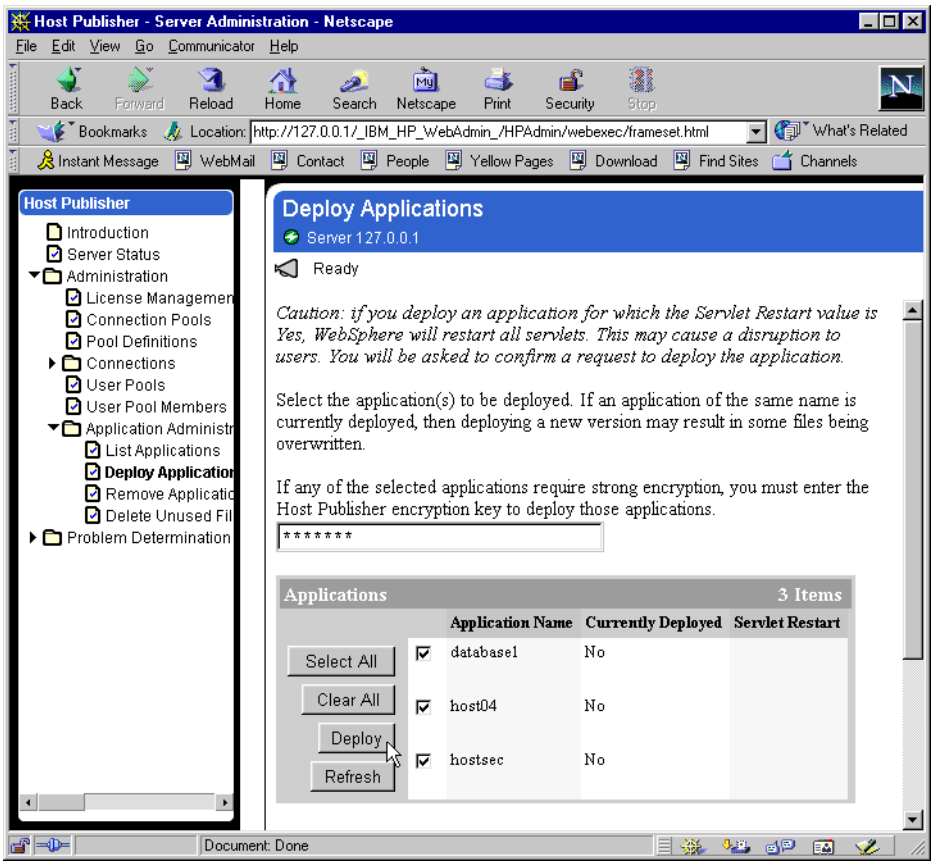

Figure 74. Entering Key for password encryption during deployment

4.4.2 Host Integration Object

The following will step you through defining the host connection and the associated security issues:.

1. To start building a host Integration Object, select **Start -> Programs -> Host Publisher Studio -> Host Access** to open the Host Access window, then click **Next**.

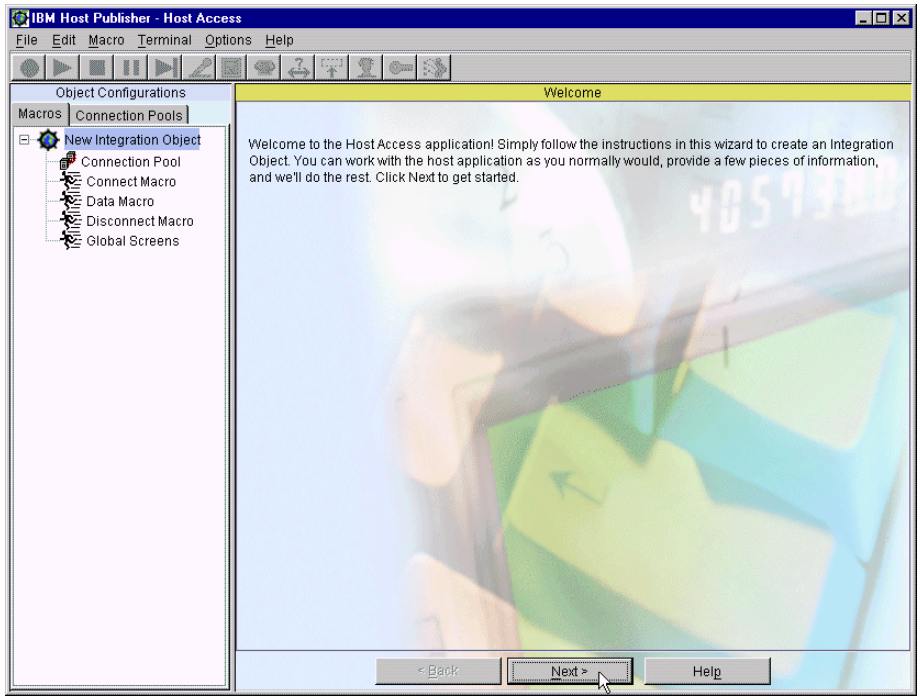

Figure 75. Host Publisher session setup

2. Select **New Integration Object** and click **Next**.

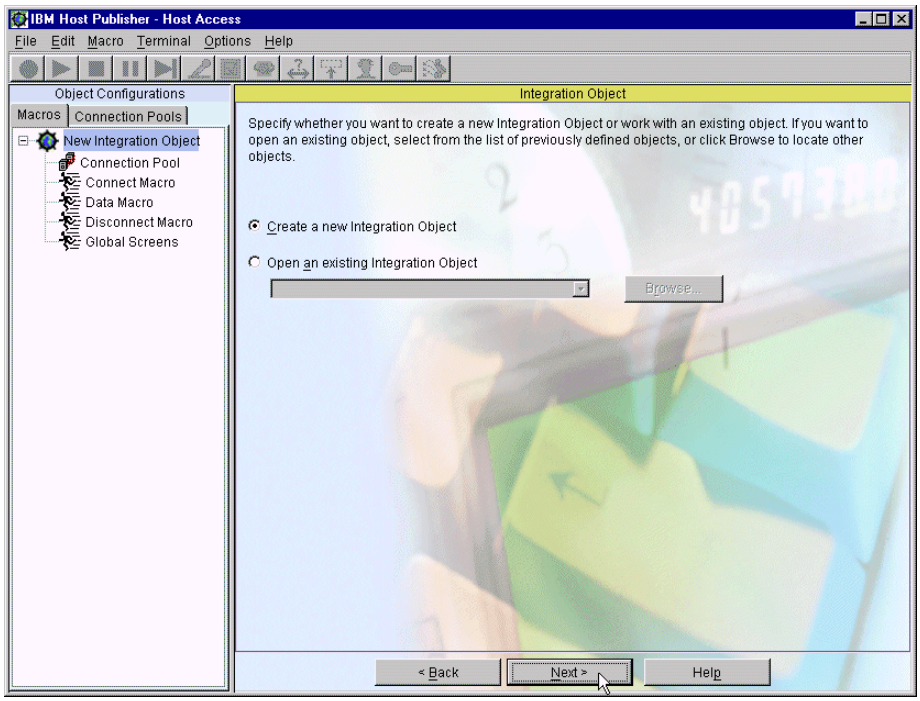

Figure 76. Create Integration Object

3. In our example, the host is an OS/390, MVS03B, and the Telnet session will connect to port 23001, which has been configured for secure connections. Therefore, enter the desired fields, as shown in Figure 77, and click **Next**.

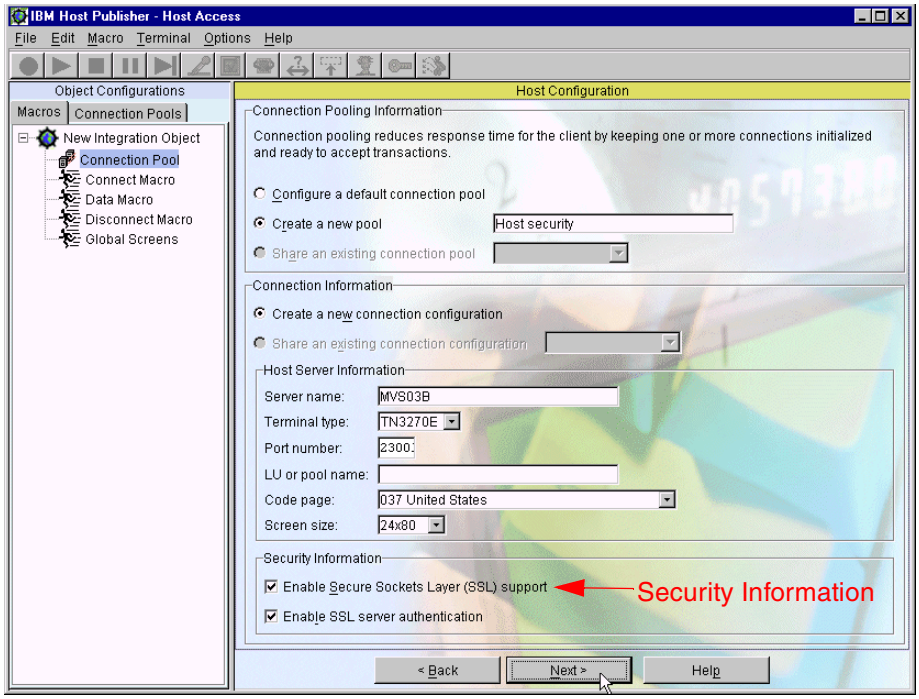

Figure 77. Connection pool security

4. Now you have the option to configure a user list.

   Figure 78 shows how you enter a user password for a host user ID. A list of users and passwords is provided in the User List Configuration pane. These user IDs and passwords can be used when you create the connect and data macros.

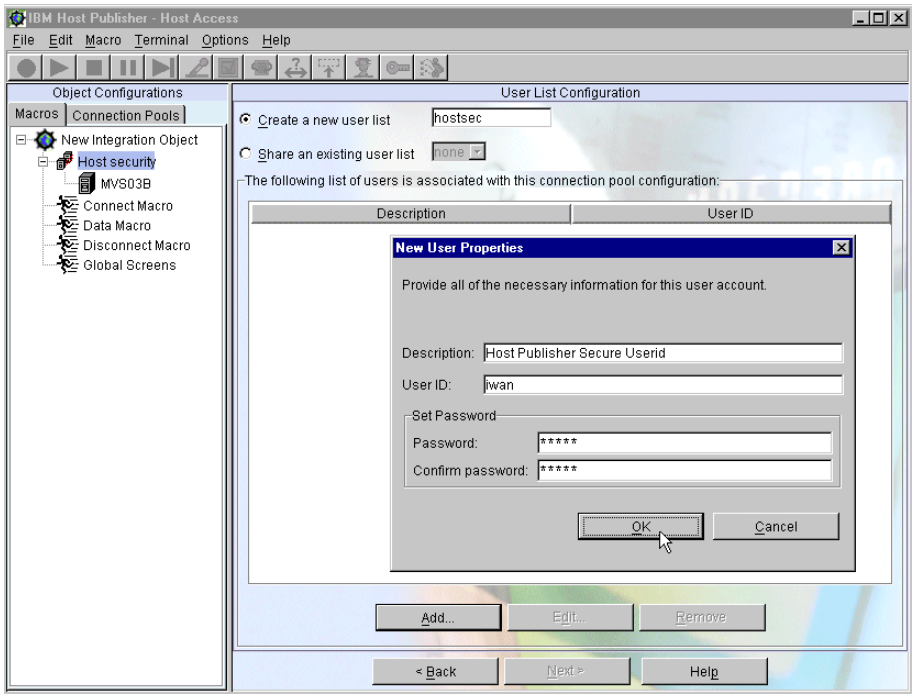

Figure 78. Host security - user ID and password

5. Click **Next** then **Save**, and save the Host Integration bean, for example Hostsec.hpi.

> **Note**
>
> If your SSL session fails to connect, you can find useful information in the status bar. The possible error messages during session setup are listed in Appendix B, “Understanding the OIA” on page 231.

After saving all of your host Integration Objects, you can create a Host Publisher application.

If you created a user list, you can select a user ID from the created list when building the connect macro. Figure 79 illustrates how this is done.

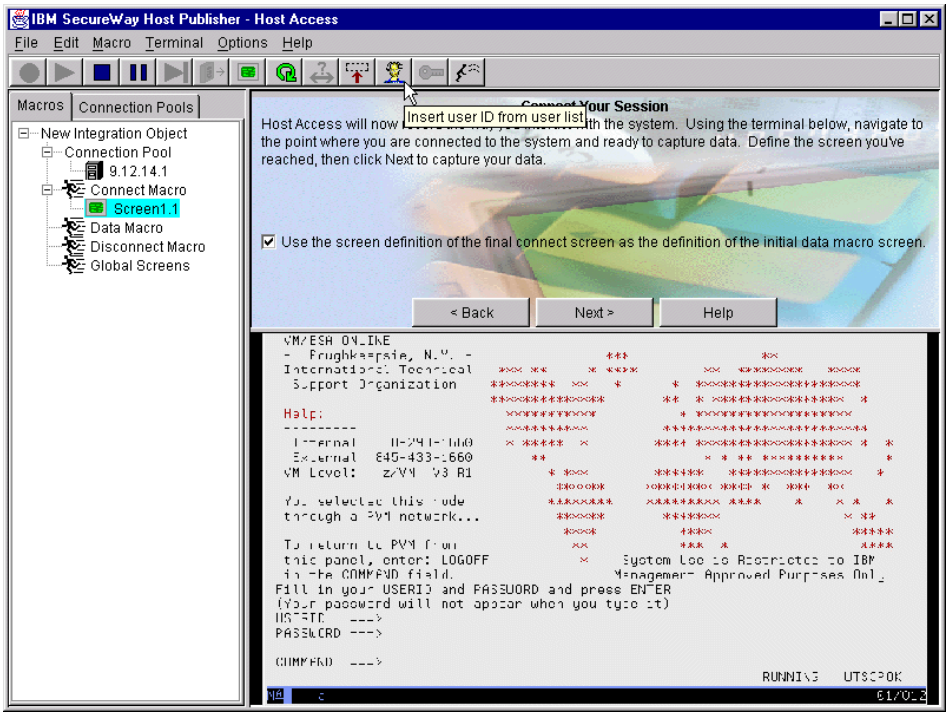

Figure 79. Inserting a user ID from the user list

Figure 80 shows how you will insert a password from the created user list.

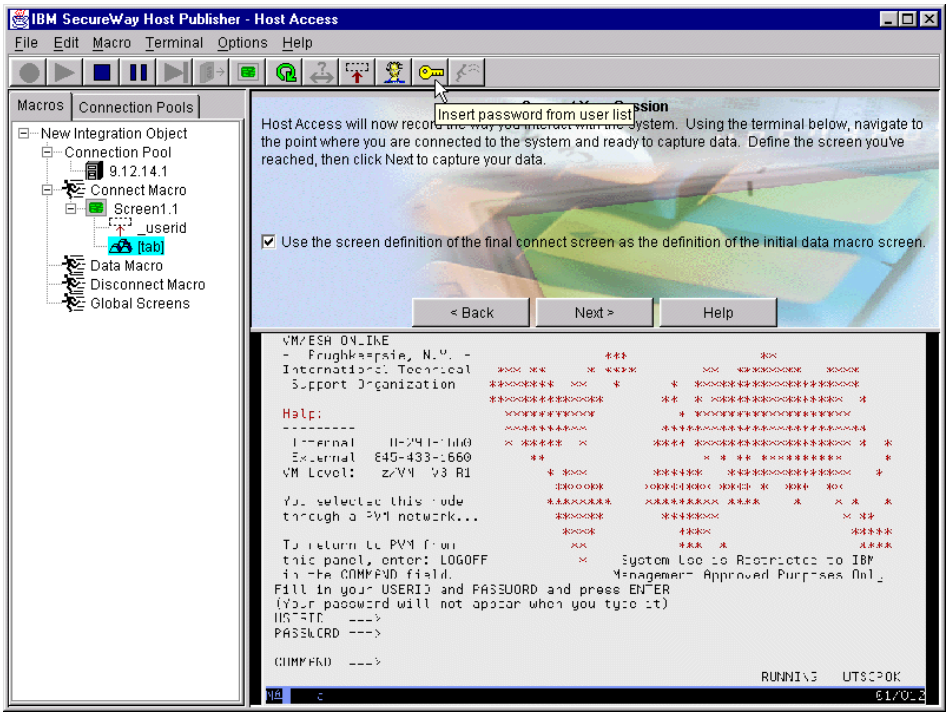

Figure 80. Inserting a password from the user list

More information on creating the Host Publisher application is found in *Building Integration Objects With IBM SecureWay Host Publisher Version 2.1*, SG24-5385.

4.4.3 Transfer application to server and deploy

After creating a Host Publisher application in the Host Publisher Studio, you must transfer it to the Host Publisher Server and deploy it.

During the transfer to the server you are prompted for the security options for this password, so select the **Encrypt** data option. Notice that you can define multiple user IDs and passwords if required.

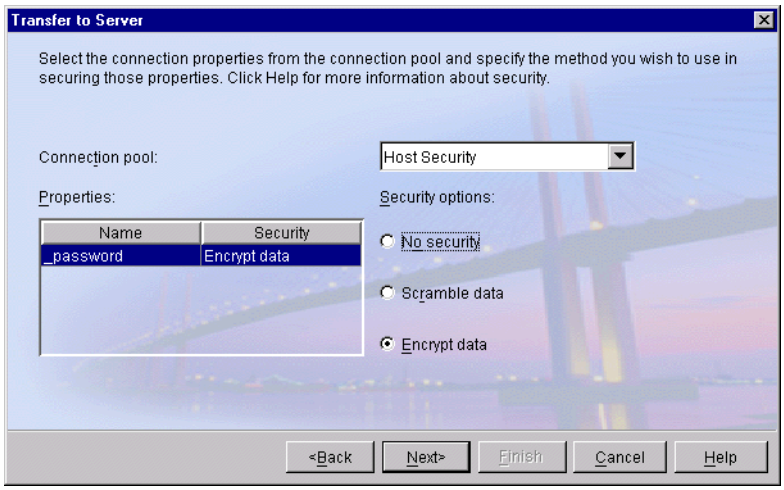

Figure 81. Transfer application to Host Publisher server

If you selected encryption, you are now prompted to enter an encryption key as illustrated in Figure 82.

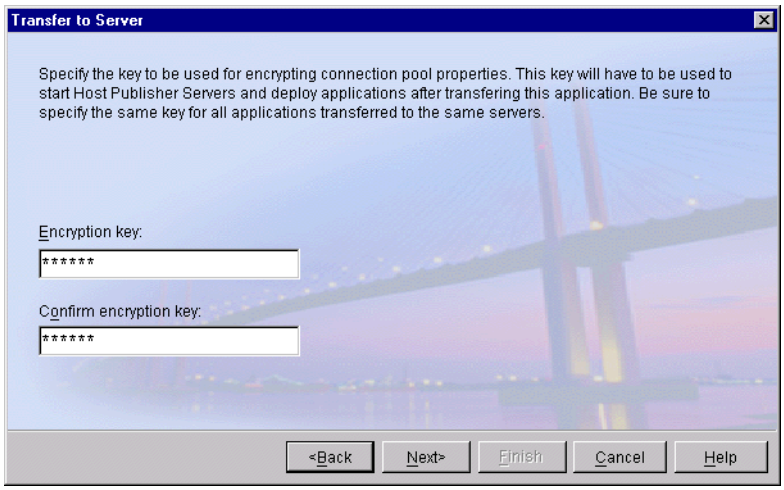

Figure 82. Providing a Key for password encryption during transfer

As a result the userpool XML file, `C:\Hostpub\Server\staging\shared\hostsec.userpool` shows the defineproperty tag with an encrypt value of "2" (encrypted) and the value for the password in now encrypted and translated into 64-ASCII code as illustrated in Figure 83.

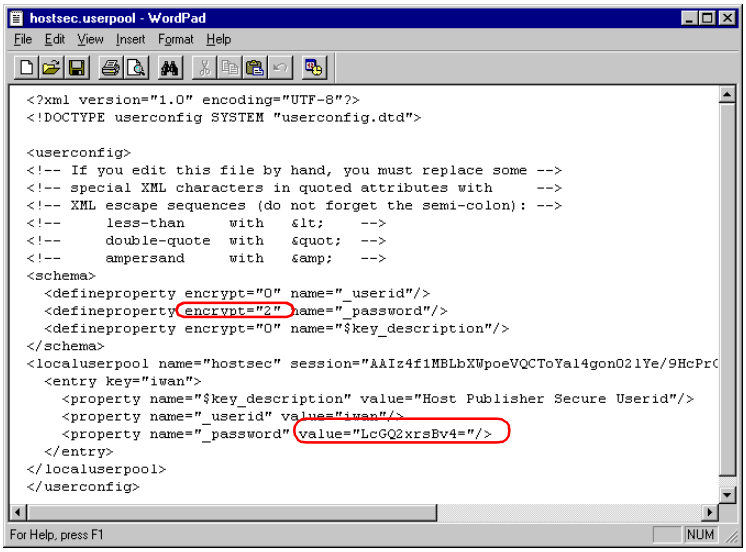

Figure 83. Encrypted password in userpool XML file

When the host application is deployed (see Figure 84), the same password must be provided and the reverse process takes place to recover the user passwords.

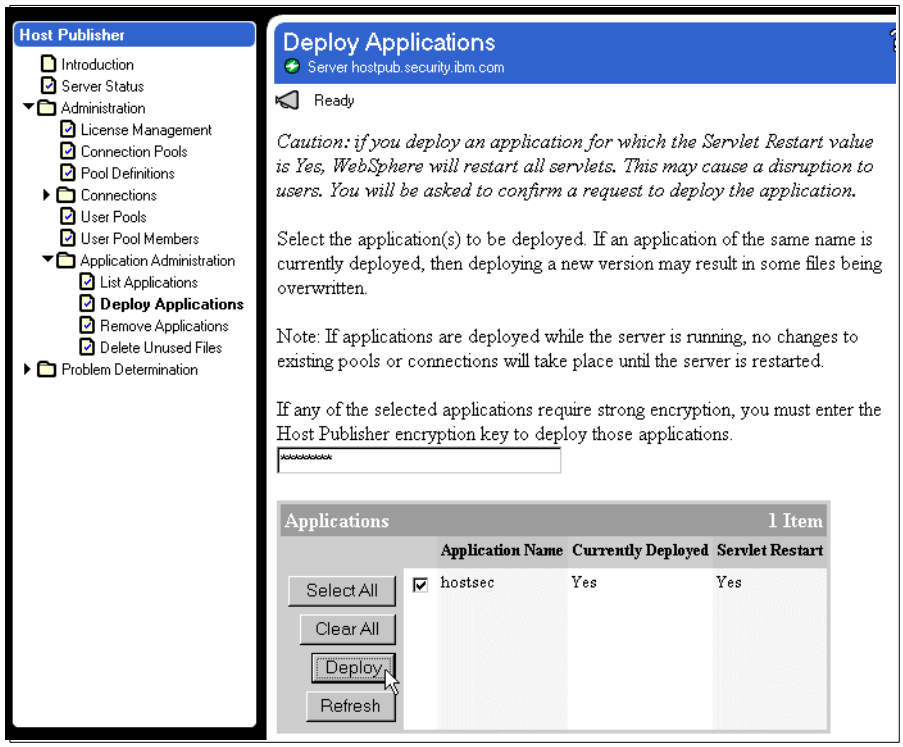

Figure 84. Entering Key for password encryption during deployment

Note

If you have to restart your Host Publisher Server, you also must enter the encryption key first. For example if you boot your PC, or when you stop and start HostPubServer in IBM WebSphere Application Server (see 4.5, “Configure IBM WebSphere Application Server” on page 138), the Host Publisher Server will not start up automatically.

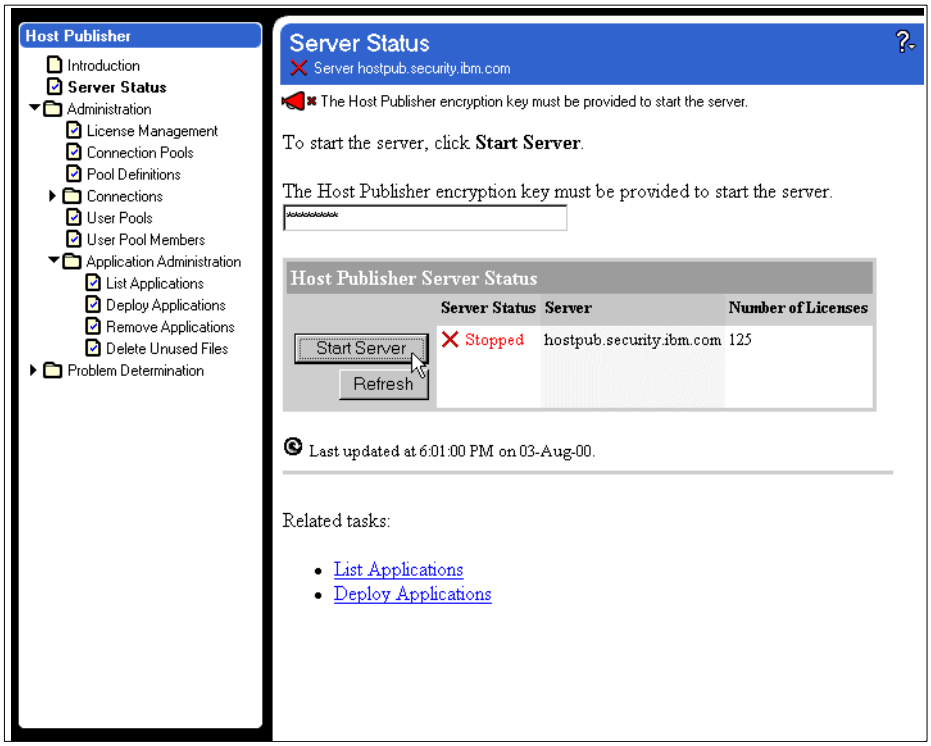

Figure 85. Start IBM WebSphere Host Publisher server

4.5 Configure IBM WebSphere Application Server

A WebSphere Application Server can provide a platform for multiple hosts. Each of these hosts is represented by a virtual host name and a list of one or more DNS aliases by which it is known. When a servlet request is made, the server name and port number component of the URL is compared to a list of all known aliases in an effort to locate the correct virtual host and serve the servlet. If no match is found, an error is returned to the browser. When no port number is specified in the URL, port 80 is assume.

There are several conditions that may not be obvious that will require you to add an alias:

- If your URL specifies a port number, then you must define an alias that includes the port number.

- If you use HTTPS to connect with your WebSphere Application Server, you must define an alias with the port number that HTTPS is using, even if you are using the default port of 443.
- If your Web server is host for multiple IP addresses, each IP address must have an alias and appropriate port number(s).

Table 5 illustrates the required alias rules.

Table 5. WebSphere alias examples

Reference URL	Required alias
http://127.0.0.1/servlet/HODConfig (usable only from the WebSphere machine)	127.0.0.1
http://localhost/servlet/HODConfig (usable from the WebSphere machine)	localhost
http://bigtex.itso.ral.ibm.com/servlet/HODConfig	bigtex.itso.ral.ibm.com
https://bigtex.itso.ral.ibm.com/servlet/HODConfig	bigtex.itso.ral.ibm.com:443
http://bigtex/servlet/HODConfig	bigtex
https://bigtex/servlet/HODConfig	bigtex:443
http://205.223.100.15/servlet/HODConfig	205.223.100.15
https://205.223.100.15/servlet/HODConfig	205.223.100.15:443

If the Web server is properly configured for all the connections and ports prior to the installation of the WebSphere Application Server, the WebSphere Application Server will add all the appropriate aliases upon installation; however, if anything changes, you must update the aliases manually.

4.5.1 Setting the WebSphere alias

To set the required aliases using the graphical interface you must first select the host that you are using, **default_host** in our example, then select the **Advanced** tab as shown in Figure 86. Next, scroll down to an empty alias field and from here enter the required alias. Repeat this process until all aliases are entered, then click **Apply**.

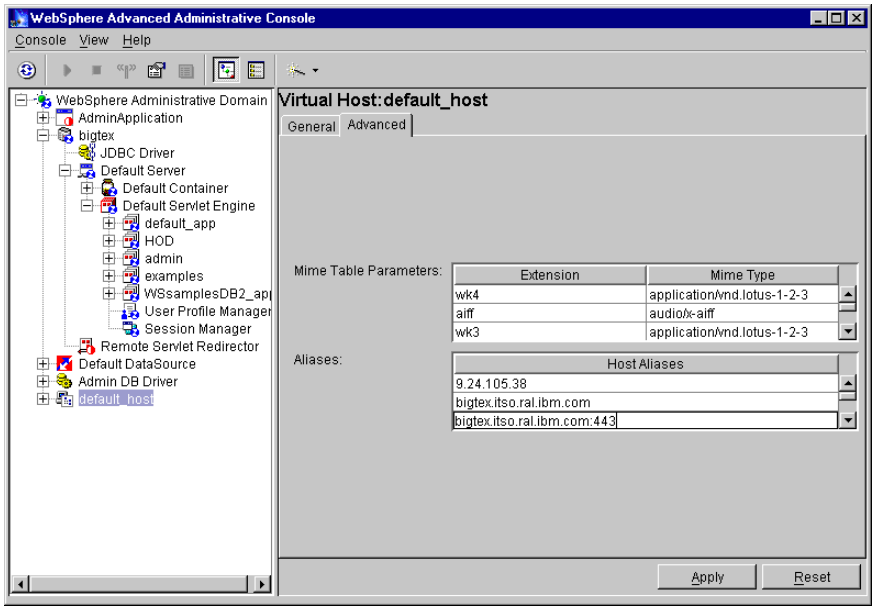

Figure 86. WebSphere default_app alias

4.5.2 Recognizing unknown CAs

Another point to remember is that the host Integration Object and WebSphere Application Server have to be able to locate the CustomizedCAs.class file when processing a request for an Integration Object configured for SSL support. Put it in the common directory, and then add the common directory to the classpath, using the same definition used to put the Host Publisher JAR files in WebSphere's classpath.

Part 2. Implementation scenarios

Chapter 5. Security using a reverse proxy

This scenario deploys applications to the Internet user using IBM WebSphere Host Publisher, and is concerned with three items:

1. Protecting the Host Publisher Server from direct access by clients on the Internet.
2. Ensuring secure sessions between the client and the Host Publisher Server across the Internet.
3. Ensuring that the Telnet session to the host is protected.

This scenario was built using the IBM HTTP Server, the IBM SecureWay Firewall, and IBM WebSphere Host Publisher running on Windows NT. Any Web server or firewall with equivalent capabilities may be used. The configuration for this scenario is shown in Figure 87.

To protect the Host Publisher Server (L) from direct access, place it behind a firewall (D) that operates as a reverse proxy. To secure the session traffic between the client and the Host Publisher Server, the firewall is configured to allow only HTTPS traffic while disabling FTP and Telnet traffic. The Telnet traffic between the Host Publisher system (L) and the host system (G) will be secured with SSL sessions.

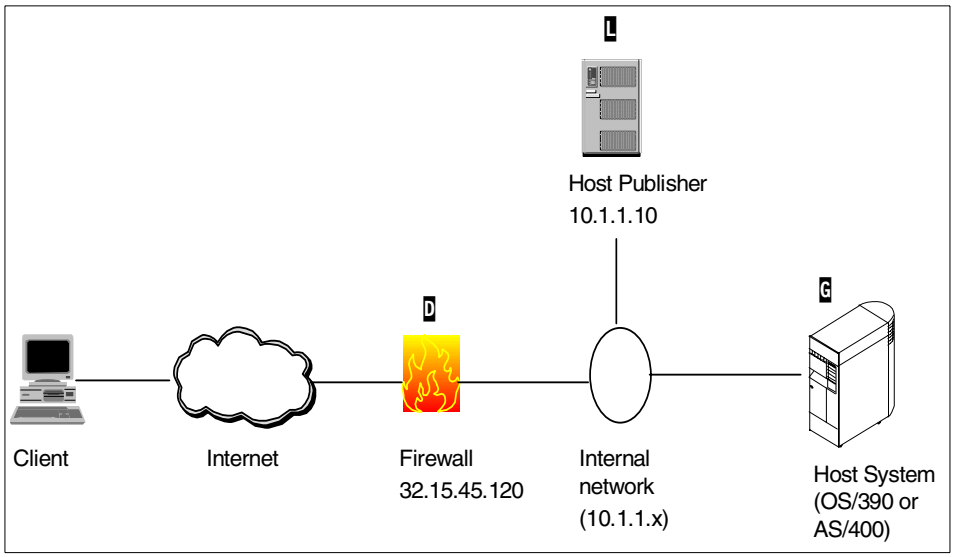

Figure 87. Simple firewall with reverse proxy

This chapter discusses how to configure the HTTP server for SSL, and explains the basic firewall settings (reverse proxy) used to secure the access to the internal network.

5.1 Host Publisher Server HTTPS configuration

In order to support HTTPS (SSL), on your Web server you require an X.509 digital certificate signed by a Certificate Authority who is designated as a trusted CA by your server.

There are three ways to obtain a certificate:

1. Buy a certificate from an external CA provider
2. Obtain a temporary test certificate from an external CA provider
3. Create and use a self-signed certificate

Information on how to create a self-signed certificate using the IBM Certificate Management utility is found in 3.10.1, “Creating a self-signed certificate” on page 109.

To set up a secure connections using a digital certificate signed by a Certificate Authority you must perform the following steps:

1. Create your server key database and certificate request.
2. Receive the public certificate for the CA if it is not already a well-known CA.
3. Receive your certificate signed by the CA, and install it.
4. Register the server key database with the server.
5. Start the server.

5.1.0.1 Create the certificate request

The IBM Key Management Utility supports creating a certificate request for a CA. To launch this utility from the Windows NT Start menu, select **Programs -> IBM HTTP Server -> Start Key Management Utility** and open the key database. Check your documentation on how to launch this utility on other supported platforms.

1. Click **Personal Certificate Requests** from the key database content frame as shown in Figure 88.

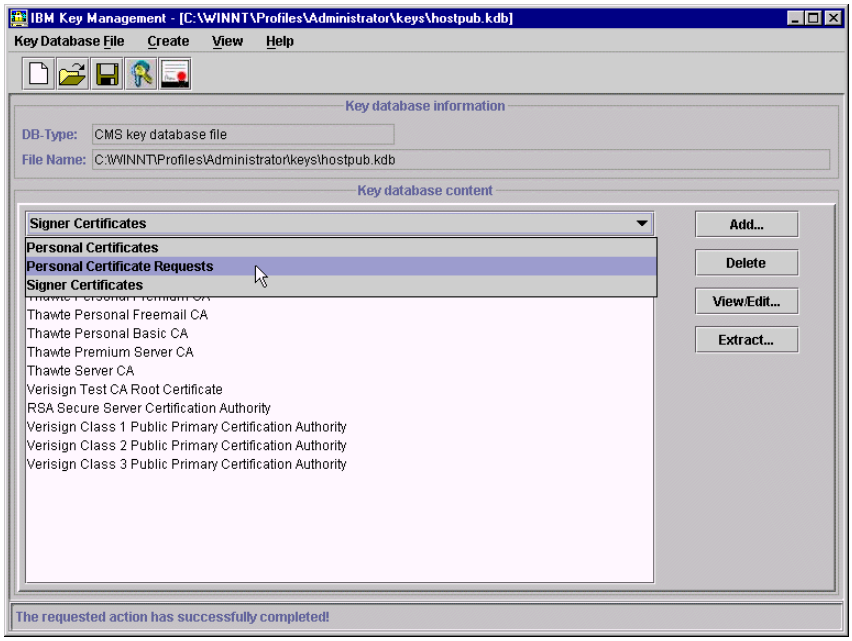

Figure 88. Personal certificate request

2. Select **New** to bring up the Create New Key and Certificate Request window as shown in Figure 89.

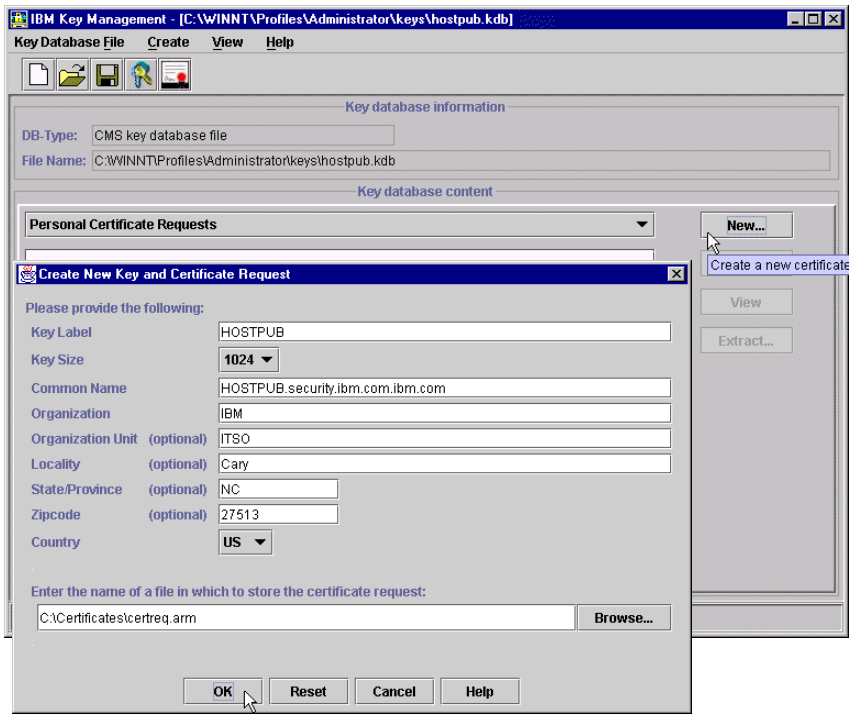

Figure 89. Personal certificate request details

3. Use the guidelines from Table 3 on page 110 to complete the certificate request, then click **OK** to continue.
4. A window will pop up to remind you to send the certificate to the CA. In the case of an external CA you would typically use the CA's secure Web site. If you had your own CA, such as Lotus Domino Certificate Authority, or Tivoli SecureWay Public Key Infrastructure, you would consult the procedures for that CA. Click **OK** to continue.

   As you see in Figure 90, you will see that your certificate request has been added in the key database.

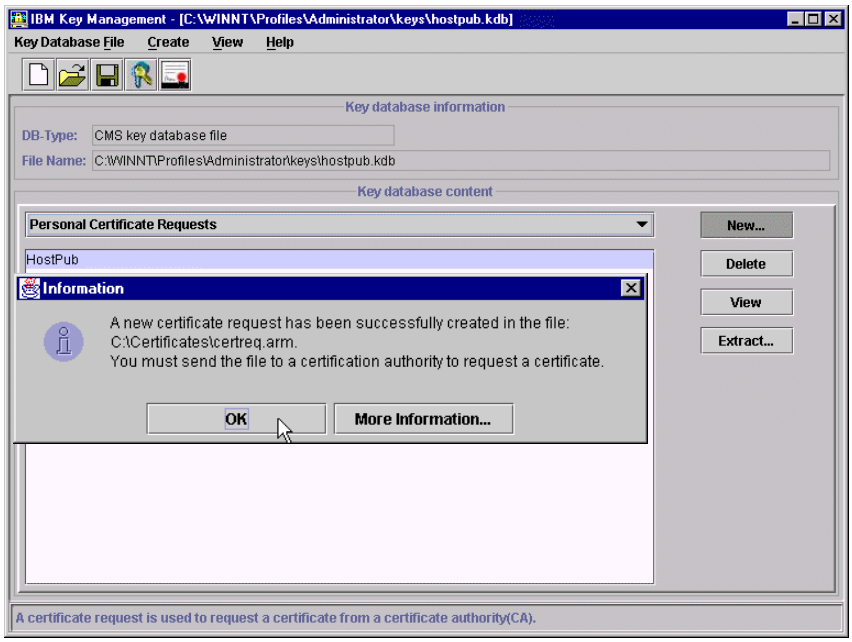

Figure 90. Personal certificate request complete

Now that you have the certificate request, it must be signed by the Certificate Authority. When the certificate has been signed and returned by the CA, you must receive it using the IKEYMAN utility. If the CA that is signing your certificate is already in your list of trusted CAs, then you can skip directly to 5.1.0.3, “Receive certificate signed by CA” on page 148.

5.1.0.2 Receive the CAs certificate

You only need to exercise this step if your CA is not already in your list of well-known CAs.

1. Select **Signer Certificates** in the Key Database content frame (see Figure 88 on page 145), then click **Add**.
2. When the Add CA’s Certificate from a File window opens (see Figure 91), select the appropriate CA file on your hard disk and click **OK**.

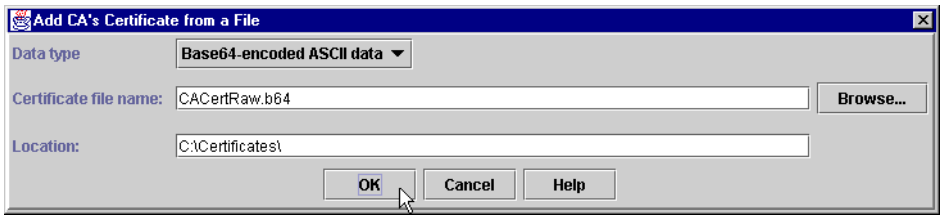

Figure 91. Add a CA certificate

3. Enter a meaningful label for the certificate, for example “Trust Authority CA”, then click **OK**.
4. The CA will be added into the list of Signer Certificates as shown in Figure 92.

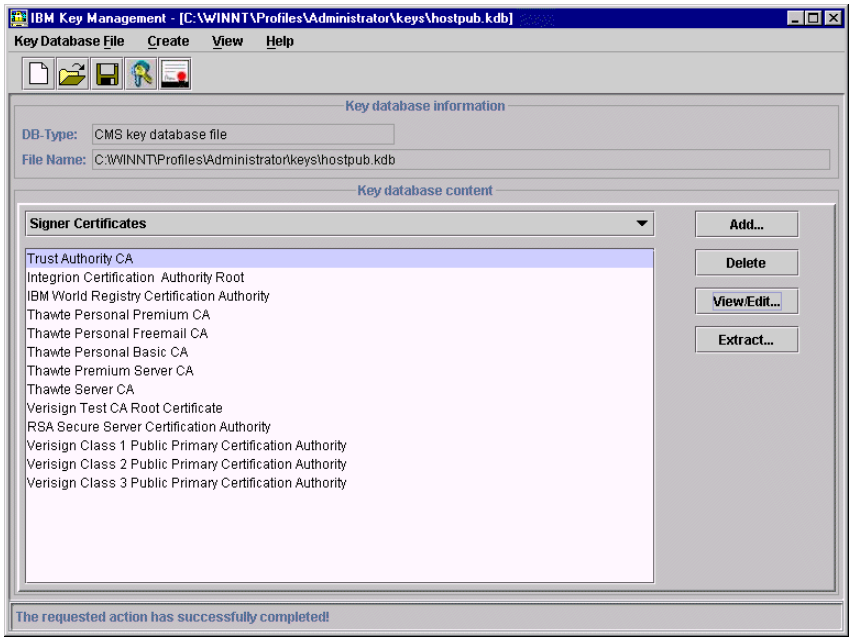

Figure 92. New CA in key database

5.1.0.3 Receive certificate signed by CA

When you receive your signed certificate from the Certificate Authority, you must update your key ring file by using the following instructions:

1. Select **Personal Certificates** in the Key Database content frame (see Figure 93), then click **Receive**.

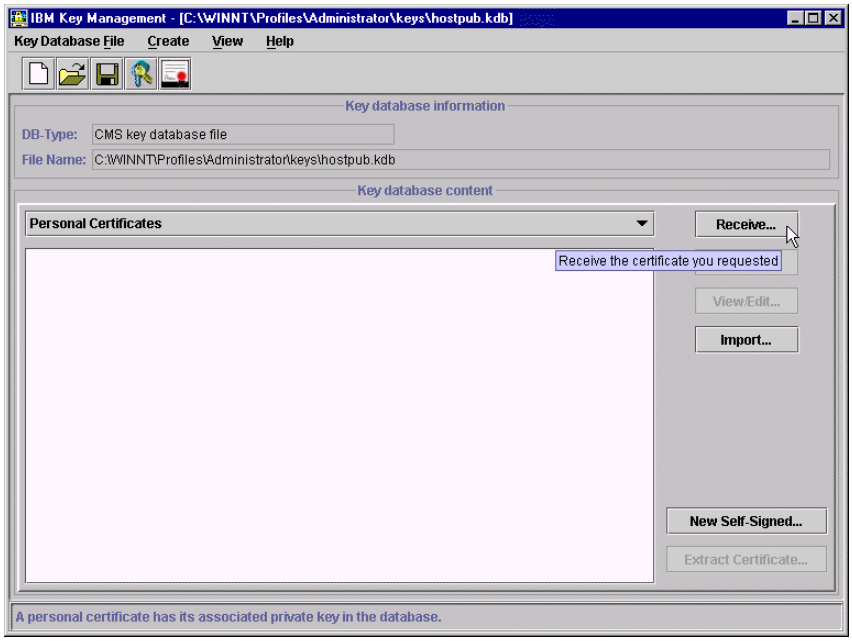

Figure 93. Receive signed certificate

2. The window shown in Figure 94 will open. Specify the location of the appropriate signed certificate from your hard disk and click **OK** to add it to the database.

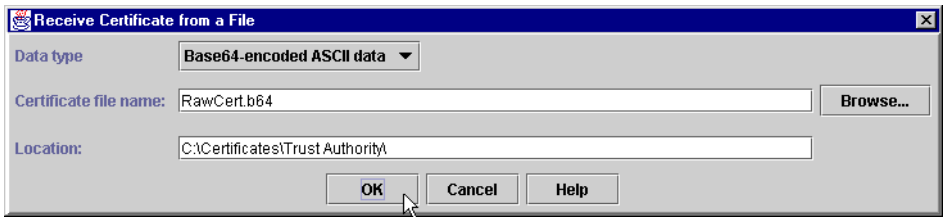

Figure 94. Locate signed certificate

3. The signed certificate will show up in the Personal Certificates (see Figure 95).

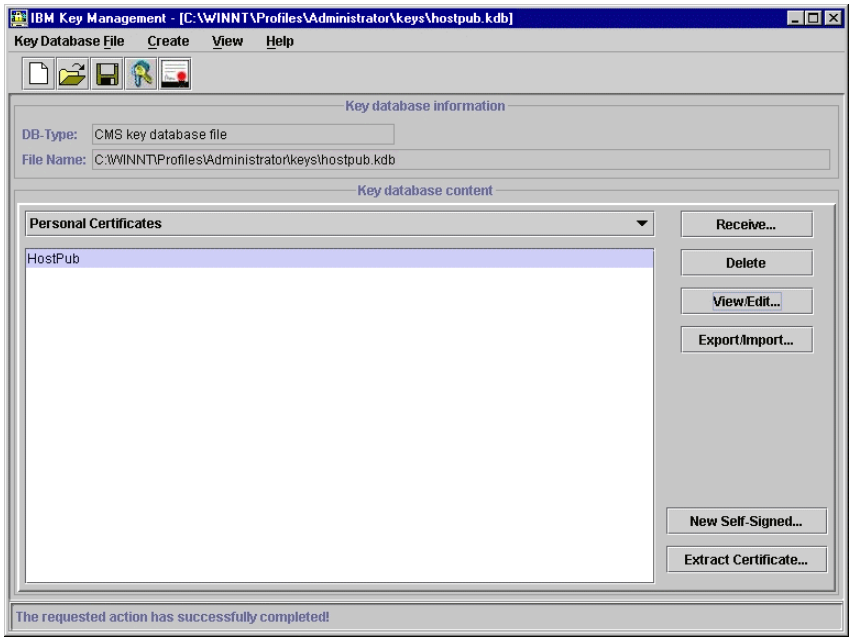

Figure 95. HTTP server personal certificate

5.1.0.4 Register the server key database with the server

To make these certificates operable, you must register the key database with the server. For the IBM HTTP Server, you must edit the configuration file httpd.conf and add the necessary SSL statements. These statements can be found in the httpd.conf.sample configuration file. Both files, httpd.conf and httpd.conf.sample, are located in the C:\Program Files\IBM HTTP Server\conf\ subdirectory.

Consult your Web server documentation for how this is done for other platforms.

Open the sample file with a text editor:

1. Locate the lines as shown in Figure 96 and copy them into httpd.conf and uncomment the one that represents the level of encryption for which your HTTP server is enabled.

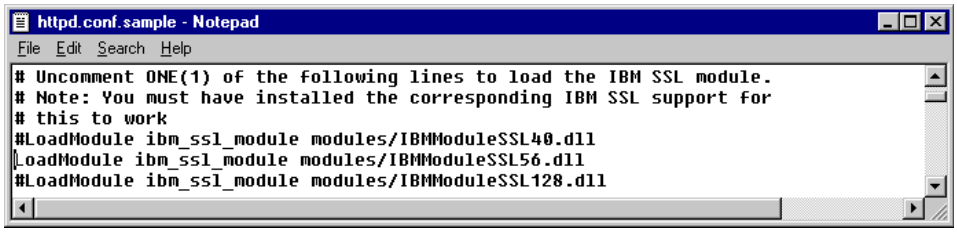

Figure 96. Configure HTTP Server SSL level

Important

You must install IBM SSL support in IBM HTTP Server 1.3.6.2 for this to work.

2. Copy the “Start sample SSL“ section from the httpd.conf.sample file into the httpd.conf file, and make sure the following steps are performed. Refer to Figure 96 and Figure 97 for the next three items.
   a. Specify the secure port to be used (default is 443) for secure connection between the Web server and the Web browser.
   b. Place the host name of the server in the virtual host stanza for secure port (443).
   c. Ensure that the SSLEnable directive is uncommented in the virtual host stanza.

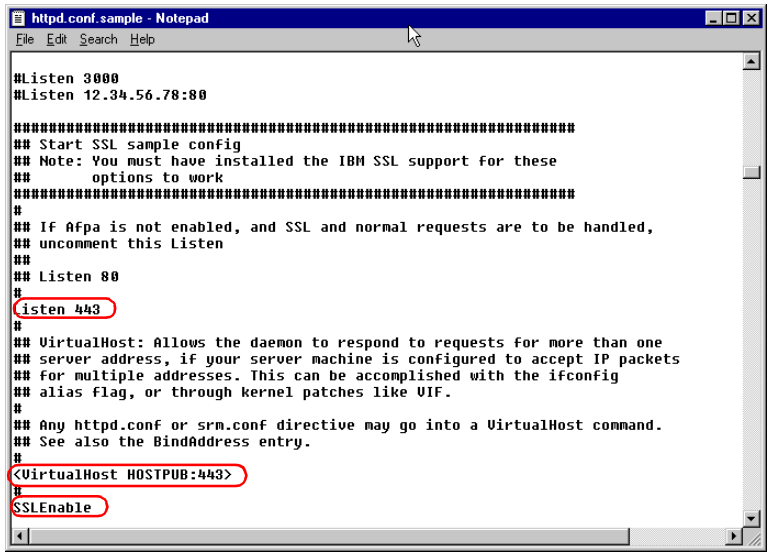

Figure 97. Enable SSL on HTTP Web Server

d. If you want to use server authentication, you must specify the key database file that will be used as shown in Figure 98.

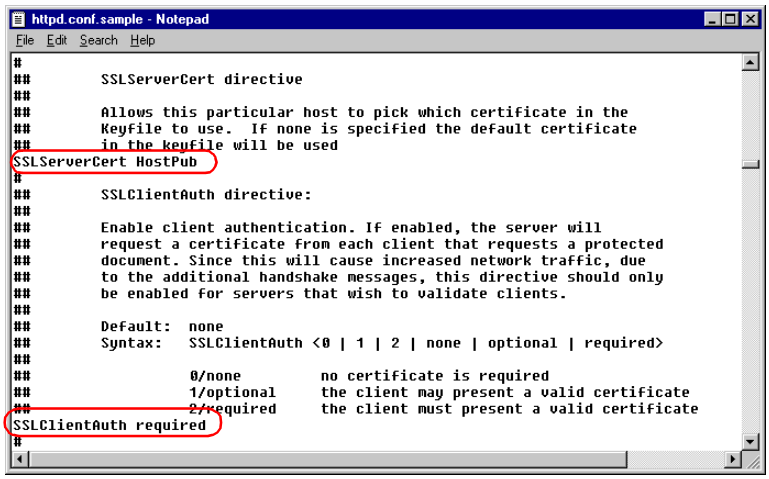

Figure 98. Require client authentication for HTTP Web server

e. Figure 99 shows how to set the Keyfile directive, which belongs outside of the virtual host stanza, to the directory where you stored the key database. This example shows the directory in c:/winnt/profiles/administrator/keys/hostpub.kdb.

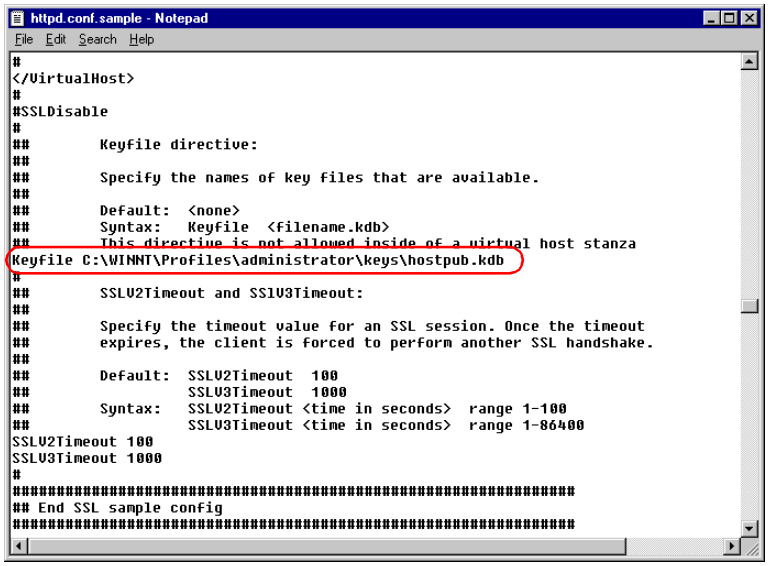

Figure 99. Keyfile directive setting

Note

The configuration file httpd.conf contains default settings. If you have installed a previous version of the Web server, your existing configuration file is preserved as httpd.conf and the default configuration file is renamed httpd.conf.default.

3. To have a completely secure environment, you should not only configure HTTPS but also disable HTTP in the Web server. To do so, add SSLEnable as the last line in the httpd.conf file. This way SSL will be enabled globally and not only for the virtual host.

5.2 Firewall configuration

A proxy server provides client access to network resources to which they do not have direct access. Clients are configured to access the proxy directly, as though the proxy were providing the service, then the proxy reissues the same request to the real server. Clients require no modification, apart from accessing the proxy IP address instead of the target server IP address.

The IBM SecureWay Firewall has a proxy for each protocol, HTTP, HTTPS, FTP, etc. This scenario used the HTTP/HTTPS proxy, which can be used in several modes:

- Outbound
- Reverse
- Chained

This scenario uses the only the reverse mode.

5.2.1 Reverse HTTP

Reverse connections are possible, allowing the HTTP proxy to relay HTTP requests from many Internet clients to a server in the secure network.

It is recommended that the best location for the secure HTTP server is in a segregated network or DMZ. It is not recommended that you allow access to HTTP servers in your secure network from the Internet.

Using the configuration shown in Figure 87 on page 143, the client will address the IBM WebSphere Host Publisher using the IP address of 32.14.45.120. When the firewall receives the request on the configured port (443 for HTTPS), it will forward it to IP address 10.1.1.10 on the same port. Thus, the firewall breaks the connection so that at no point is there any direct communication between the client and the server; therefore, the server is protected.

More explanation on how to set up the reverse proxy, proxy authentication and how to configure the clients' browser can be found in Chapter 6, "Proxy", in *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855.

Chapter 6. Security using a DMZ

The DMZ is the most commonly deployed scenario for a host integration solution. The DMZ scenario described in this chapter is illustrated in Figure 100. The operations of both IBM WebSphere Host On-Demand and Host Publisher Server are covered in this environment.

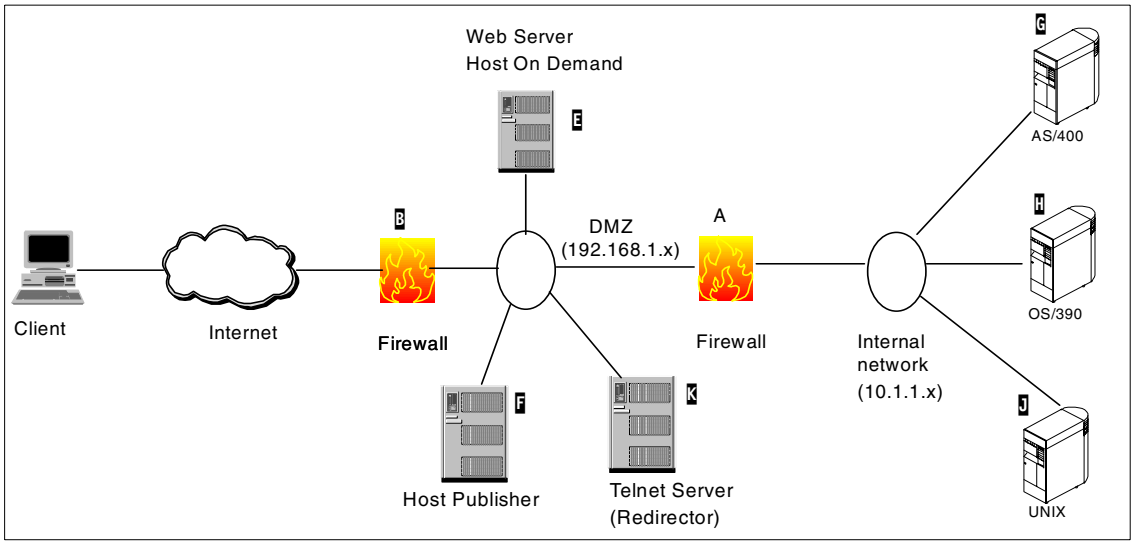

Figure 100. Classic DMZ

For the purposes of illustration, assume that the organization shown in Figure 100 is an insurance company. The IBM WebSphere Host Publisher system is set up to allow anyone who has a policy with the insurance company to access the records in his or her account. The IBM WebSphere Host On-Demand system is set up to allow employees and agents direct access to the internal OS/390, AS/400 and UNIX systems in the corporate organization.

The security policy has been set so that all data for all users accessing these systems via the Internet must travel over secure connections. Customers using the IBM WebSphere Host Publisher inquiry system will require only a browser that supports an SSL connection to the IBM WebSphere Host Publisher system. In addition to using SSL sessions for all transactions employees and agents will be required to authenticate with all servers they come in contact with. As such, the company is issuing each employee and agent authorized for Internet access an X.509 digital certificate signed by a Certificate Authority that is owned and operated by the insurance company.

This scenario used the Tivoli SecureWay Public Key Infrastructure product, formerly named IBM Trust Authority.

Any public Certificate Authority, such as VeriSign or Thawte can provide valid X.509 certificates; however, by operating the Certificate Authority themselves, the company felt that they had more control over who gets the certificates and can revoke any certificate immediately. The decision whether a company should use a pubic Certificate Authority or to deploy their own is a complex one and outside the scope of this book. There is a very good IBM Redbook on this subject, *Deploying a Public Key Infrastructure*, SG24-5512.

The company has set a security policy that all Telnet sessions from outside the secure network must be secure and use client authentication for additional protection.

6.1 External firewall

In the classic DMZ scenario, two firewalls are used, one between the non-secure Internet and the DMZ (shown as firewall B in Figure 100 on page 155), and the other between the DMZ and the secure internal network (shown as firewall A). Firewall B is usually a filtering router that screens input to allow access to only specific protocol/port/internal address combinations. All other requests would be rejected.

In this scenario, firewall B was set up with the following general filtering rules:

- HTTP(S) request are allowed to only the IBM WebSphere Host On-Demand server (E) and the IBM WebSphere Host Publisher server (F). All other HTTP(S) requests will be rejected.
- All HTTP traffic must be to the secure port (443) to the two Web servers.
- Multiple Telnet ports must be opened (one for each internal TCP/IP port/address combination); however, all Telnet traffic is restricted to the Telnet Redirector (K). All other Telnet traffic will be rejected. Refer to 3.4.2, “Configuring the Host On-Demand Redirector” on page 79 for details on configuring the IBM WebSphere Host On-Demand Redirector.
- All FTP traffic will be blocked. IBM WebSphere Host On-Demand supports FTP; however, the FTP implementation does not support SSL transport at this time.
- All other traffic and ports will be blocked. This will include the IBM WebSphere Host On-Demand configuration port; therefore, there are two options:

a. Use the configuration servlet to provide this support over a secure HTTPS connection, or

b. Use the Deployment Wizard to create customized pages that define all the sessions the users need and either forbid saving of personal session changes, or allow them to save any customized changes for their sessions to their own workstations. This is the option that was selected for this scenario.

6.2 Servers in the DMZ

All servers in the DMZ would be responsible for controlling access to their own resources. For example, the Web server on which IBM WebSphere Host On-Demand is running should restrict all connections to HTTPS.

A more rigid access control would be to implement a user ID and password challenge. Alternatively you could require the user to provide their X.509 certificate for authentication. Any user not providing the proper credentials would be denied access to the server.

6.2.1 IBM WebSphere Host Publisher server

The Web server here should be set up to accept only HTTPS connections. The IBM WebSphere Host Publisher application should be set up to prompt the user for access control information such as account number and password. These controls should be built into the Integration Object or they could originate at the host application. Optionally, the Telnet connection to the required server on the other side of firewall A could also be established via an SSL connection using basic SSL or client authentication.

6.2.2 IBM WebSphere Host On-Demand server

The Web server on this machine is configured to accept only HTTPS connections; therefore, all client downloads will be encrypted. The client is also required to present its X.509 certificate to authenticate in order to download the IBM WebSphere Host On-Demand applet.

The Deployment Wizard was used to create custom pages rather than manage additional user IDs and the configuration server. All custom pages have the license use tracking function disabled. Without license use tracking or user logins, there is no requirement to run the IBM WebSphere Host On-Demand service manager. By not running the service manager, the configuration port (default 8999) is not open, thus eliminating an additional entry point for a potential hacker.

Without the service manager running, remote administration is not possible; therefore all maintenance will be done on a shadow server on the secure side of the network, and all changes will be copied to the IBM WebSphere Host On-Demand server in the DMZ.

6.2.3 DMZ Telnet server

This Telnet server is here to provide an additional layer of security. There are two possible Telnet server configurations:

1. A Telnet gateway:

   A Telnet gateway will accept TN3270 of TN5250 connections inbound and convert them to an SNA link to the respective host. The SNA traffic does not travel through the internal firewall (A). This configuration does not support a VT connection.

2. A Telnet Redirector.

   There are two IBM Telnet Redirectors. They both support client-side and pass-through SSL support.

   a. The IBM WebSphere Host On-Demand Redirector

   b. The Communications Server for AIX V6 Telnet Redirector

6.2.3.1 Telnet Redirector

The IBM WebSphere Host On-Demand Redirector is not recommended when either large volumes of connections or high volumes of transactions. However, in the initial implementation, the number of connections and volume of traffic will be within the range supported by the IBM WebSphere Host On-Demand Redirector. When the number of connections or the volume of traffic through the IBM WebSphere Host On-Demand Redirector increases significantly, it will be replaced by the Communications Server for AIX Telnet Redirector. Both redirectors support client-side SSL, pass-through security modes, and client authentication.

This scenario deployed the IBM WebSphere Host On-Demand Redirector initially because of the low volume of internal users. Plans are to migrate to the Communications Server for AIX V6 Telnet Redirector when the number of sessions increases.

For information on how to configure Communications Server for AIX, refer to the product documentation or to *IBM Communications Server for AIX, V6 New Features and Implementation Scenarios*, SG24-5947.

Refer to 3.3, “Defining a secure Telnet session” on page 63 for details on how to configure the client sessions, and to 3.4, “The Host On-Demand

Redirector" on page 77 for details on how to configure the Redirector. The Redirector configuration that was used in this scenario is shown in Figure 101.

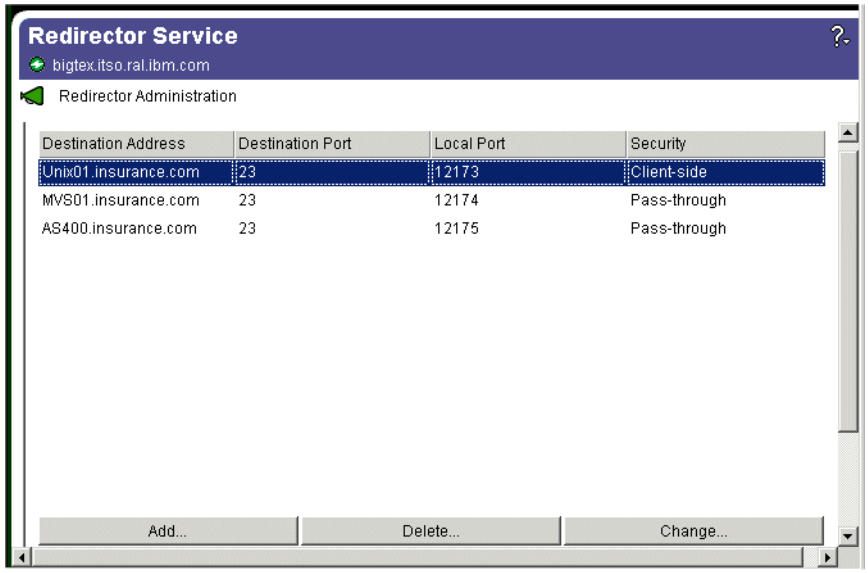

Figure 101. Redirector configuration

6.3 Internal firewall

The inner firewall (A) must have rules defined to restrict traffic based upon the company's security policy. For example, only Telnet requests coming from the Telnet Redirector (K) and the Host Publisher Server (F) are allowed through the inner firewall (A) to the internal network. Since IBM WebSphere Host On-Demand does not support secure FTP it is wise to disable FTP requests completely on the external firewall (A). Database On-Demand does not support secure client connections; therefore, Internet-based clients were not allowed to use Database On-Demand, and Database On-Demand sessions were restricted to the Host Publisher Server only. IBM Host Publisher Studio uses FTP to transfer applications to the IBM Host Publisher Server. Make sure that you enable FTP through the internal firewall (B) between these machines when updating or deploying changes to the IBM Host Publisher Server.

6.4 Secure servers

The UNIX servers do not support SSL connections; therefore, they must rely on the Telnet Redirector to provide SSL support and client authentication. In the early stages of the deployment, the IBM WebSphere Host On-Demand Redirector is used, with plans to migrate to Communications Server for AIX with its Telnet Redirector when the volume of clients increases.

The OS/390 and AS/400 systems must be configured to support secure Telnet sessions with client authentication. If the OS/390 system is at V2R10 or higher, Communications Server for OS/390 may be configured to use Telnet-negotiated session, see 3.3.2, “Telnet-negotiated session” on page 64, to minimize the number open ports. For details on how to configure OS/390 to support Telnet-negotiated sessions refer to *IBM Communications Server for OS/390 TCP/IP 2000 Update Technical Presentation Guide*, SG24-6162, or the product documentation.

6.5 Clients

All IBM WebSphere Host Publisher clients use HTTPS to access the application. The Integration Object or the host application is responsible for further access control.

All IBM WebSphere Host On-Demand clients communicate to the Web server and IBM WebSphere Host On-Demand server using HTTPS protocols. The VT client communicates with the UNIX servers via the Telnet Redirector, which is configured for client-side SSL. The Telnet clients will communicate with the Telnet server using SSL and optionally client authentication. The 3270 and 5250 clients communicate with their host systems via the IBM WebSphere Host On-Demand Redirector configured in pass-through mode. The AS/400 and Communications Server for OS/390 systems are configured for SSL and client authentication.

Chapter 7. Security using a virtual private network

Several significant changes have increased the interest in virtual private networks (VPN):

- The use of Internet Service Providers (ISP) has increased and companies are now less likely to own their own network. In some cases, you might be able to replace a dedicated line with a VPN. If you have mobile users, you might be able to use a worldwide dial-up ISP through which clients can connect.
- With the increased use of DSL and cable modems, users prefer to use their DSL/cable connection rather than dial-up for their host connection.
- Working from home has become popular. The Small Office/Home Office (SOHO) worker needs access to many applications, not just terminal emulation and file transfer. Traditionally, a firewall would be used to provide access to many services, but to do this many adjustments to firewall rules may be required. A VPN offers the convenience of a standard set of firewall rules.
- Windows 2000 includes VPN client support.
  - Recent improvements in VPN allow for tunnels to be constructed for dynamic IP addresses (such as a DHCP-assigned IP address from an Internet Service Provider).
  - Some VPN vendors are improving the throughput of their products using hardware-encryption accelerator cards.

7.1 Security considerations

The use of a VPN is not without its security concerns:

- Once the datagram exits the VPN gateway into the secure environment, it will normally be unencrypted (similar to any other equipment on the secure network). If you require end-to-end encryption from client-to-host, you will need to double encrypt, which is very slow. With IBM WebSphere Host On-Demand, this would entail a standard IBM WebSphere Host On-Demand SSL session be established between the client and the Telnet server, in addition to the use of the VPN.
- Some VPN client software packages do not show whether a connection is encrypted or not, such as with the security symbol (padlock). Since SSL is not typically used with VPN tunnels, there is no way for client applications such as your browser or Host On-Demand to detect that the connection is secure; therefore, the security symbol (padlock) that is shown in the

browser window and the IBM WebSphere Host On-Demand Telnet window will not appear locked.

- Since it is expected that most VPN clients will connect via the Internet the remote client will likely be the subject of a security attack, such as the planting of a Trojan horse (refer to 1.3, “Typical threats to security” on page 4).

7.2 Using IBM WebSphere Host On-Demand with a VPN

You may select the VPN server and client of your choice. The scenario built while writing this book used the following configuration.

- IBM Firewall 4.2 running on IBM AIX 4.3.3 with maintenance pack 5 (J in Figure 102).
- Ashley Laurent Client for Firewall 4.2 for NT, which is included with IBM Firewall 4.2.

The strategy was to use an X.509 digital certificate as identification to access the secure network gateway (J).

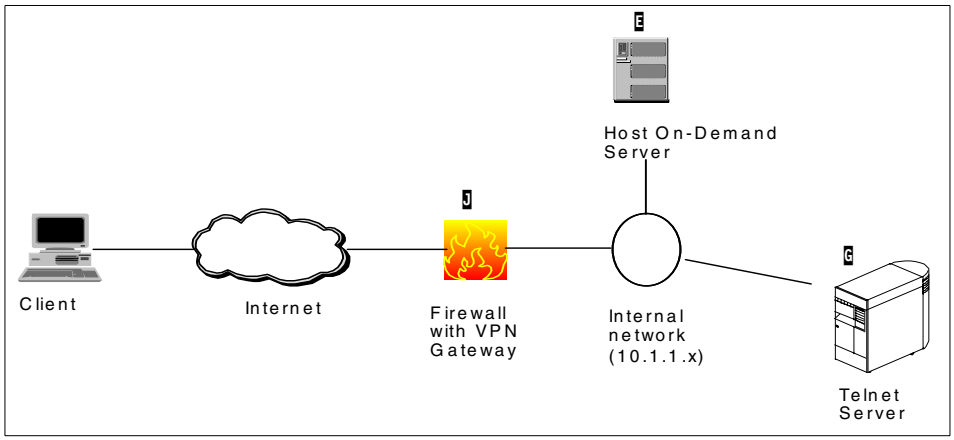

Figure 102. Sample VPN network

Figure 103 illustrates the data flows for an IBM WebSphere Host On-Demand connection to the Telnet server. It is assumed that the digital certificates have been issued to the clients and that the VPN gateway has been properly configured to use the digital certificate as the authentication mechanism.

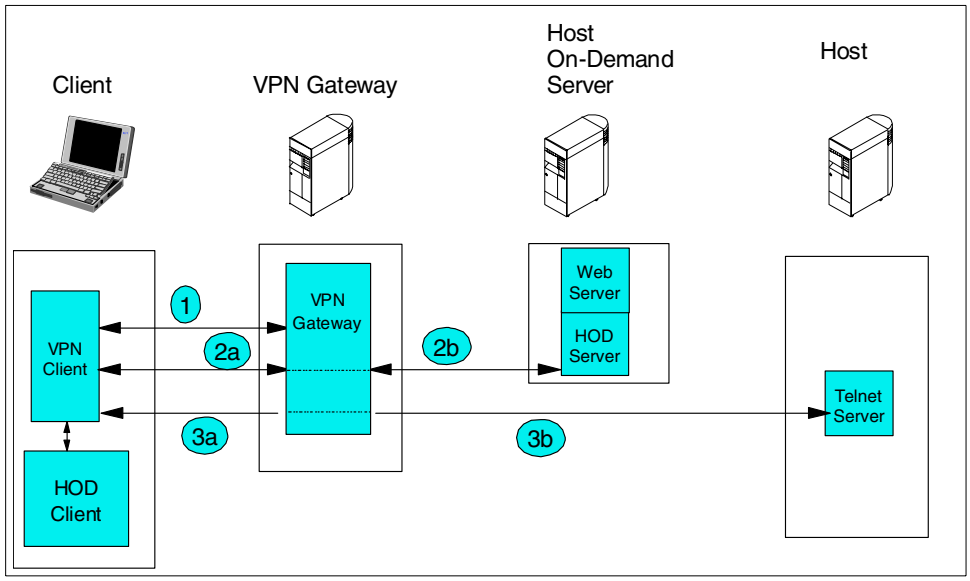

Figure 103. Simplified connection process

1. The user must connect and use his X.509 certificate to validate with the VPN gateway to establish the VPN tunnel.
2. The client requests access to the IBM WebSphere Host On-Demand client page via the browser.
   a. The request goes from the browser through the VPN client (encrypted) to the VPN gateway.
   b. The VPN gateway forwards the request to the IBM WebSphere Host On-Demand server using the originating protocol (HTTP or HTTPS).
3. All IBM WebSphere Host On-Demand clients communicate via the VPN client through the VPN gateway.
   a. All requests from the client to the VPN gateway will be encrypted.
   b. The data flowing from the VPN gateway to the destination will flow using the security negotiated between the IBM WebSphere Host On-Demand client and its destination. Refer to Chapter 3, “IBM WebSphere Host On-Demand security” on page 47 for details on the security capabilities.

The key message to understand with a VPN solution is that the VPN client and VPN gateway provide encrypted sessions between the VPN client and the VPN gateway regardless of any other security methodology employed by

IBM WebSphere Host Publisher or IBM WebSphere Host On-Demand. The encryption stops at the VPN gateway, so if additional security is desired between the VPN gateway and the destination host system, it will be the responsibility of the IBM WebSphere Host Publisher or IBM WebSphere Host On-Demand systems to enable and enforce these security policies.

The following redbook and Redpaper are excellent resources for understanding more details on deploying VPN solutions.

- *A Complete Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, SG24-5309.
- *Remote Access to AS/400 with Windows 2000 VPN Clients*, `http://www.redbooks.ibm.com/redpapers/pdfs/redp0036.pdf`

Chapter 8. Security using Lotus Domino

One of the challenges faced by using Host On-Demand across the Internet is authentication. The client needs to verify the identity of the server (server authentication) and the server needs to verify the identity of the client (client authentication). One of the most popular tools for doing this authentication is the X.509 digital certificate. Digital certificates are also used in the negotiation phase of establishing an encrypted SSL connection between the client and the server. Refer to 2.1.6, “Public Key Infrastructure” on page 20 for more information.

Digital certificates are administered by a Certificate Authority (CA) such as VeriSign, Thawte or Equifax. These companies are in the business of authenticating users and issuing digital certificates. If a client or server later becomes untrusted, the CA would revoke the certificate. Most browsers have an extensive list of trusted CAs.

One of the advantages of using a private CA is that it gives the administrator the capability to issue and revoke certificates to only those individuals to whom they wish to provide access. If an employee leaves the company, the administrator has complete control to revoke the employee’s certificate immediately.

Lotus Domino also acts as a Certificate Authority. Because companies who have deployed Domino also have an administrator in place to validate users and manage Lotus Notes IDs, the administrator can assume the additional responsibility of managing X.509 digital certificates, thus potentially saving the company significant amounts of money. The distribution of X.509 certificates can be minimized by sending them out attached to an existing Notes user ID file.

The Lotus Domino administrator has the ability to integrate e-mail, groupware, Web server and certificate management functions into a single administrator. This integration is a very powerful capability.

Let’s now examine how a company could use their Lotus Domino environment as a platform to extend Host On-Demand to Internet-based clients in a secure environment.

8.1 Scenario configuration

The configuration for this IBM WebSphere Host On-Demand scenario is shown in Figure 104.

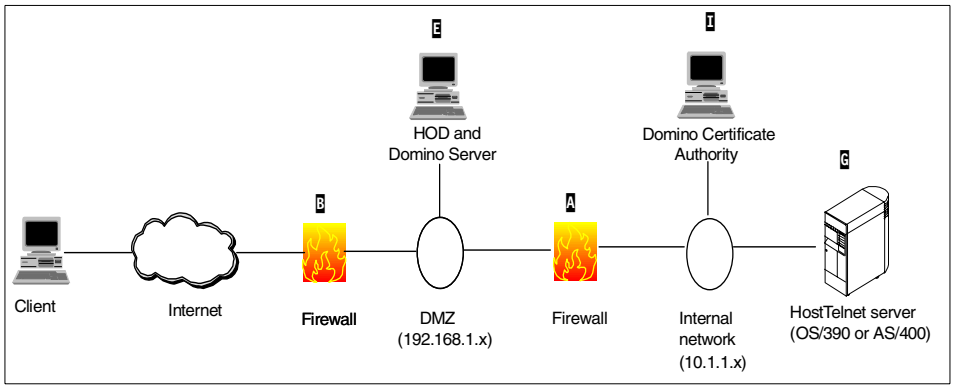

Figure 104. Host On-Demand with Lotus Domino network setup

In this scenario a Lotus Domino Web server (E) is set up in the DMZ to host the Host On-Demand server. The Lotus Domino CA (I) is on the secure internal network, and is the signer of the certificates for all clients and servers, E and G.

The following steps outline how the environment shown in Figure 104 is set up and used:

1. The Domino Certificate Authority and a master key are established on server I.
2. The CA administrator will then extract its public key and send it securely to the other servers (E and G) and to all the clients. Since the Domino Certificate Authority is an unknown authority, all the servers and clients must import the public certificate as a trusted root.
3. The Certificate Authority server (I) and the Host On-Demand and Domino Server (E) will each make a public key from their respective private keys. These servers will give the keys to the CA administrator, who must sign the certificates.
4. A Domino administrator will put a CA signed certificate in each Notes user's ID file. The administrator should send this ID file to the user in a secure manner.
5. The user will extract the certificate and store it in the Web browser as their personal certificate.
6. The user will connect to the Host On-Demand server via HTTPS and load the Host On-Demand applet. The user will recognize and trust the server and the server will authenticate the client.
7. The Host On-Demand applet will connect to the host via Telnet SSL.

8.2 Setting up the Domino server

The Notes Domino server serves two capacities in the scenario presented: as the Web server on which Host On-Demand is installed, and as the Certificate Authority that will issue and manage all X.509 digital certificates. This chapter reviews the setup of these components for the scenario presented in this book. Additional information may be found in the IBM Redpaper titled *Domino Certification Authority and SSL Certificates*, found at http://www.redbooks.ibm.com/redpapers/pdfs/redp0046.pdf.

Below are the steps required to set up the Domino Certificate Authority and the Web server to use the CA:

The initial steps must occur on the Domino CA server.

1. **Create the Certificate Authority database**

   This is the database where certificate requests are signed. The database will contain the public certificates for all of the clients and servers; and the “master” certificate.

2. **Create the Certificate Authority and key ring and certificate**

   This step creates the master key that will be used to sign certificates for clients and other servers.

3. **Configure the Certificate Authority profile** (optional)

   This is used to automate some of the CA administration process and set some default settings.

4. **Create the key ring and certificate for the CA server**

   This step will make the CA server a trusted server.

5. **Set up the Domino CA server to be a Web server**

   This step is required for distribution of the server certificates. Client certificates could be distributed in this manner as well, but in this scenario the certificates are distributed with the Notes ID file.

Next, the administrator must prepare the Domino Web server.

6. **Create the key ring and certificate**

   A key must be generated for this server that will need to be signed later by the Domino CA.

7. **Create Certificate Request**

   Send the key to the Domino CA to sign our certificate.

The Domino CA processes the certificate request.

8. **Approve and sign the certificate**

   The Domino CA administrator will approve the certificate request of the Domino Web Server by signing the certificate and sending notification to the Domino Web server via e-mail.

The Domino Web Server receives notification that the certificate has been signed.

9. **Receive notification of the approval**

   Receive the approval via e-mail with the location and reference number to pick up the signed certificate.

10. **Pick up the certificate from the Domino CA server**

    The Domino Web server retrieves the public certificate of the Domino CA server.

11. **Pick up the signed certificate**

    The Domino Web server retrieves the private certificate and stores it on the server.

12. **Install Host On-Demand and configure the Domino Web server**

    This server needs to be set up to run Host On-Demand.

The following is a list of considerations to keep in mind before continuing:

- The installation and general configuration of Lotus Domino, Host On-Demand, and firewalls will not be covered here. Consult the software documentation, Redbooks, and online help for information on these topics.
- Port 1352 needs to be open on the firewall between the Domino Certificate Authority and the Domino Web server. This will be used by the Notes Remote Procedure Call. The firewall should be set up to allow this port to be open only between these two IP addresses. The HTTPS port (usually 443) and/or the HTTP port (usually 80) need to be opened on firewall A and firewall B (see Figure 104 on page 166). The Host On-Demand ports must also be opened also.
- The Domino CA needs to be created on a Domino Designer client and administered on a Domino Administrator client. A single computer can be used for both. The computer with the Domino Administrator client needs to have a Web browser.
- These steps should be completed by someone with administrator privileges on the servers.

- The Host On-Demand and Domino Server will be in the DMZ. This machine should be hardened. Some ideas are:
  - Require a password to start the Domino Server process. The password is stored in the server ID file.
  - Domino automatically defines some communications ports and enables some of them. In most circumstances, TCPIP is the only port that needs to be defined and enabled. All others can be removed.
  - Unnecessary databases should be moved to different servers. Required databases, such as the Names and Addresses book, should be encrypted, views and documents should be restricted, and access control lists checked. Use caution when giving out anonymous access.
  - The Lotus Notes Agents on the necessary databases should be reviewed and access checked.
  - Each Domino domain should have a Certification Log database. This database is usually created during installation. It can be recreated with the certlog.ntf template. The database file name should be certlog.nsf.
  - Internet users can be added to the Domino Directory and passwords can be assigned. Existing users can also be assigned Internet passwords. The Domino administrator has the ability to upgrade to a more secure Internet password format. This is a good idea as long as all the users will only be accessing Domino servers at level 4.6 and above (including R5). This can be done in the Administrator's Console as shown in Figure 105 on page 170.

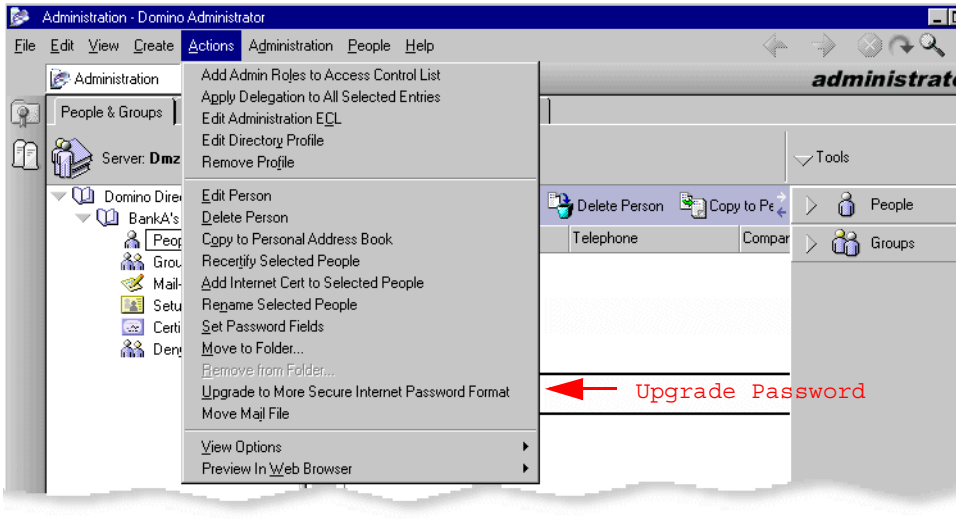

Figure 105. Upgrading the security on Internet passwords

- The key length of server certificates should be determined by encryption laws and the encryption level of software on the server. A Web server that only supports 40-bit encryption has problems with a 1024-bit key.
- The key length of users certificates should be determined by encryption laws and the encryption level of the software. A 1024-bit key cannot be imported into a browser that only supports 40-bit encryption.
- If a change is made to the HTTP configuration on the Domino Web server, the HTTP server task needs to be restarted to take effect. If security is changed, the Domino server needs to be restarted. Restarting the server also forces updates on some of the changes.
- Sometimes changes to the Names and Address book take a while to proliferate. Patience is always advised. Forcing replication and restarting the servers can speed things up.

8.3 Creating the Certificate Authority Database

To establish the server as a Certificate Authority, you need to create the Certificate Authority database. Much of the certificate administration is done in this database, such as:

- Create and administer a “master” Certificate Authority Key Ring. This ring holds the CA Certificate.
- Create the server certificate and key ring file for the CA server.

- Approve and sign (or deny) server certificate requests.
- Approve and sign (or deny) client certificate requests.
- Add client certificates issued by external CAs to the Domino Directory.

To create the database, perform the following steps on a Domino Designer client using an ID with server administrator permissions:

1. Open the Notes workstation.
2. From the menu bar, select **File -> Database -> New**.

   A window titled New Database will appear (Figure 106).
3. Use the information in Table 6 to complete the form.

Table 6. Creating the CA database form values

Field	Value
Server	The name of the Domino Server that will store the CA database - Ours is DominoCA/BankA
Title	Domino R5 CA
File Name	certca.nsf
Template Server	The name of the Domino Server - Ours is DominoCA/BankA
Show advanced templates	Select this box to be able to find the correct template
Domino R5 Certificate Authority	Highlight this template

When complete, the window should look similar to Figure 106.

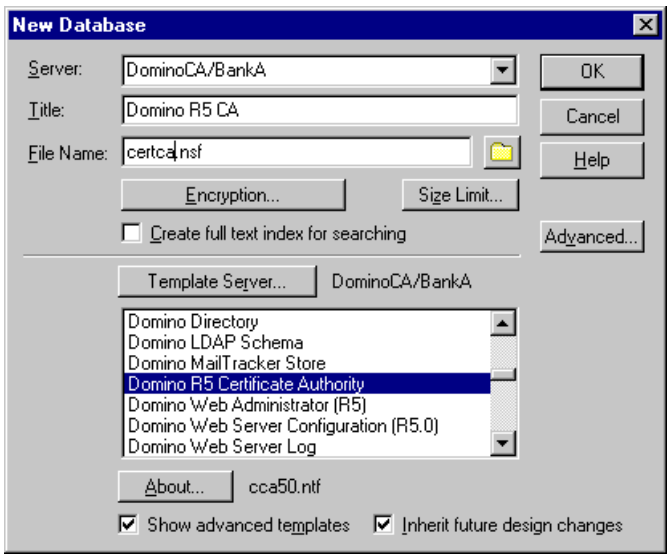

Figure 106. Creating a new CA database

4. Click **OK** when finished.

This creates the Domino R5 Certificate Authority database. The About Certificate Authority document appears. Select **File -> Close** two times, to return to your Notes client workspace.

Note

If you experience one or more error message boxes with the messages `View or Navigator 'defView' does not exist` or `You have insufficient access to perform this operation`, it is best to exit completely out of the database and open it again before performing operations in it.

When the database is created the database icon in Figure 107 appears on your Notes client workspace.

Figure 107. The Domino R5 CA database icon

8.3.1 Create the Certificate Authority key ring and certificate

You are about to create the “master” CA certificate. Use this certificate to sign server and client certificates by adding the CA's digital signature to the server and client certificates. The CA certificate is stored in a binary password protected key ring file.

This key ring file is *very* important. If the file is lost, the password is forgotten, or the file is hacked, a new key ring file must be created and everything must be recertified.

1. Open the Domino R5 CA database by double-clicking the icon shown in Figure 107. This brings you into the Certificate Authority setup menu shown in Figure 108.

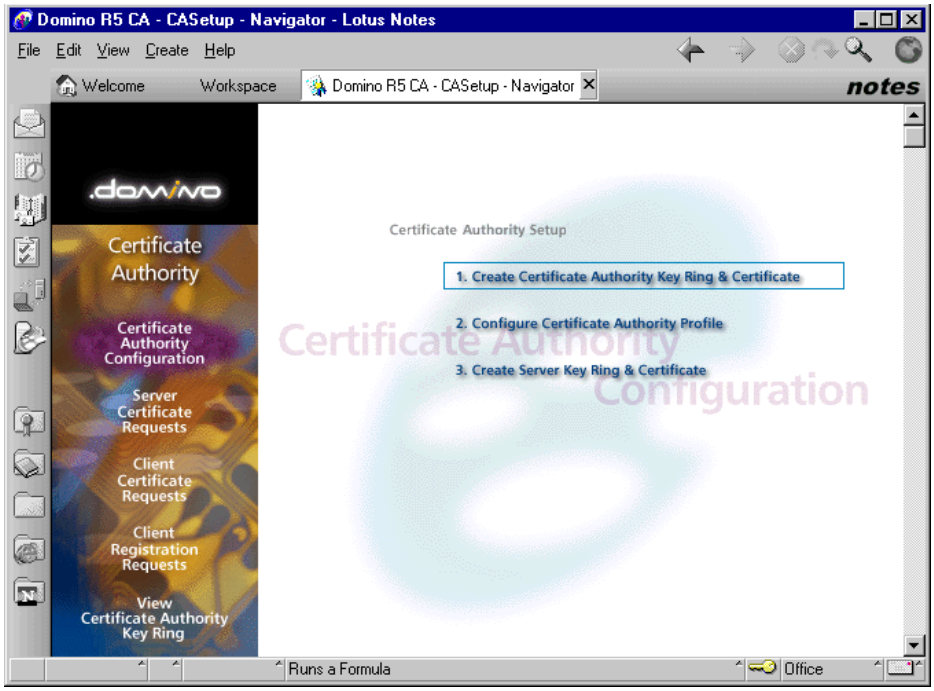

Figure 108. The Certificate Authority setup menu

2. Select **Certificate Authority Configuration** in the left pane and select **1. Create Certificate Authority Key Ring**. Fill out the form, using the information in Table 7.

Table 7. Create CA key ring form values

Field	Value
Key ring file name	Specify the file name and location for the CA key ring. The default directory is the data folder for the Notes client (usually `x:\lotus\notes\data`). The default file name is CAKey.kyr. Some use the Domino domain name in the key ring file name.
Key ring password	Specify and confirm the password for the CA key ring.
Key size	Specify the key size (in bits) of the public/private key for the CA key ring. The choices are 512 and 1024 bits (the default). The key size does not need to be the same size for the SSL and CA key rings. This key size is not limited to the size of the key supported by the Web server or browsers. At this time, the export regulations of the United States on key sizes for international use do not apply to the key sizes of the CA certificates. Since this is the "master key", many companies will choose 1024 bits for stronger protection.
Distinguished name	The distinguished name provides your unique identity as a Certificate Authority. This is the information that displays as the "issuer" in certificates that you sign being the CA. Using all distinguished name fields decreases the chance of two servers having a name collision.

When completed, the window will look similar to that shown in Figure 109.

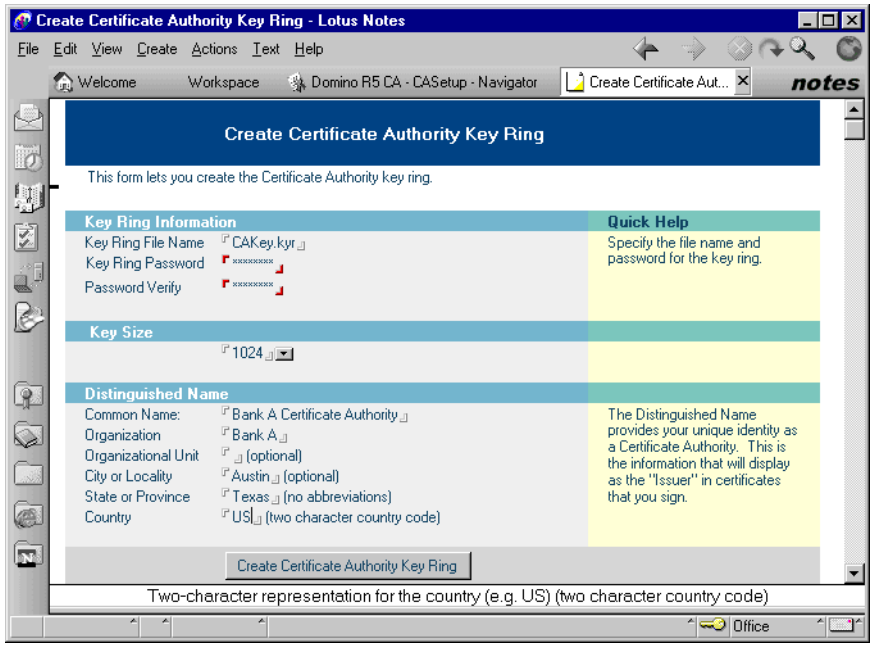

Figure 109. Creating the Certificate Authority Key Ring

3. After entering all the information, click **Create Certificate Authority Ring**, and the confirmation window similar to the one shown in Figure 110 will appear. Click **OK** to complete the process.

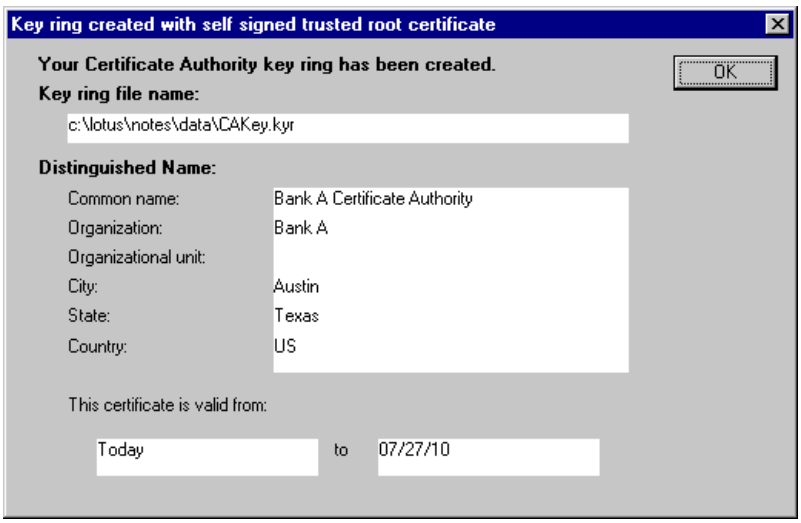

Figure 110. The confirmation message for creating the CA key

8.3.2 Configure the Certificate Authority profile (optional)

This procedure sets up some defaults and automates some certificate administrative tasks. This profile was configured to automatically send the certificate requester an e-mail after the certificate signing request is approved or rejected. Most of the settings are used for the e-mail. Follow the steps below to configure the profile.

1. In the Domino R5 CA database as shown in Figure 108 on page 173, select **Certificate Authority Configuration** in the left pane and select **2. Configure Certificate Authority Key Ring**.
2. Use the information from Table 8 to complete the form.

Table 8. Certificate authority profile form values

Field	Value
CA Key file	Specify the filename and location of the CA key ring created in 8.3.1, “Create the Certificate Authority key ring and certificate” on page 173. The default is the name and location you specified when you created the CA key ring. This needs to be updated if the location of the key file changes.
Certificate server DNS name (optional)	Specify the DNS host name (also referred to as the URL) of the certificate server. The DNS name is used in an e-mail sent to the certificate requester. The e-mail gives the URL location to pick up the certificate.
Use SSL for certificate transactions	This is also used in the e-mail sent to the certificate requester. If selected, it will point the requester to the secure port to pick up the certificate. It is recommended that you keep the default value of yes.
Certificate server port number	This specifies the TCPIP port required on the certificate server to pick up the certificate. This is used in the automatically generated e-mail sent to the certificate requester. Using HTTPS is recommended; therefore, specify your secure port (default is 443).
Mail confirmation of signed certificate to requester	If this is selected, an automatic e-mail is sent to the certificate requester with the URL (including HTTP or HTTPS), and port number for the requester to pick up his certificate. The default is Yes. If it is set to No, the administrator must set up his own mechanism to notify the certificate requester.

Field	Value
Submit signed certificates to AdminP for addition to the directory	By selecting this, the signed certificate automatically is submitted to the Administration Process, which then puts them into the queue of certificates that must be added to the Domino directory. If it is set to No, the administrator must set up his own mechanism to perform these steps manually.
Default validity period	This puts in a default value in the Validity Period field. The Validity Period field appears when the administrator is approving certificate requests. The number is in years.

When completed, your form will resemble the one shown in Figure 111.

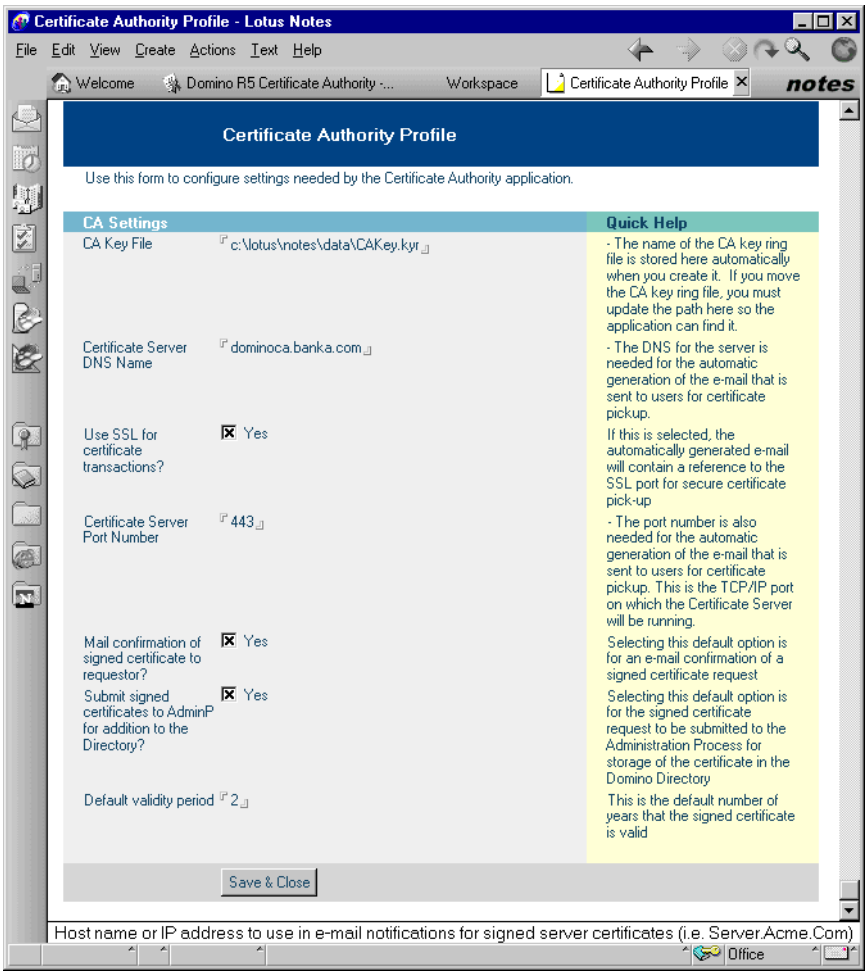

Figure 111. Creating a Certificate Authority profile

3. Click **Save and Close** to complete this phase.

8.3.3 Create the key ring and certificate for the CA server

Now you can move on to the last step of becoming a CA, which is to create the SSL server key ring and certificate for the CA server.

All SSL-enabled servers must have their own SSL key ring to establish a secure connection. The CA is no exception. SSL communication from the CA is initiated by an SSL key ring that contains the signed certificate from the CA.

1. In the Domino R5 CA database, select **3. Create Server Key Ring & Certificate** in the **Certificate Authority Configuration** view as shown in Figure 108 on page 173.
2. Using the information in Table 9, complete the form shown in Figure 113.

Table 9. Create Server Key Ring form values

Field	Value
Key ring file name	Specify a name and location for the SSL key ring file. The default directory is the data folder for the Notes client (usually x:\lotus\notes\data). The default file name is CAKeyfile.kyr. It might be helpful to name the file after the server name.
Key ring password	Specify and confirm the server key ring password.
Key size	Specify the size in bits of the public/private key for the SSL key ring on the server. The choices are 512 bits or 1024 bits. The default is 512 bits.
CA certificate label	This label identifies the CA certificate that is installed in the key ring server.
Distinguished name	The distinguished name identifies the CA server, where the users will request and pick up their certificates. The common name field will be populated with the DNS host name for your CA Web site. For some browsers, the common name in the certificate will be compared to the host name in the URL being requested. If there is a mismatch, a warning is shown. The other entries are to further distinguish this server from others.

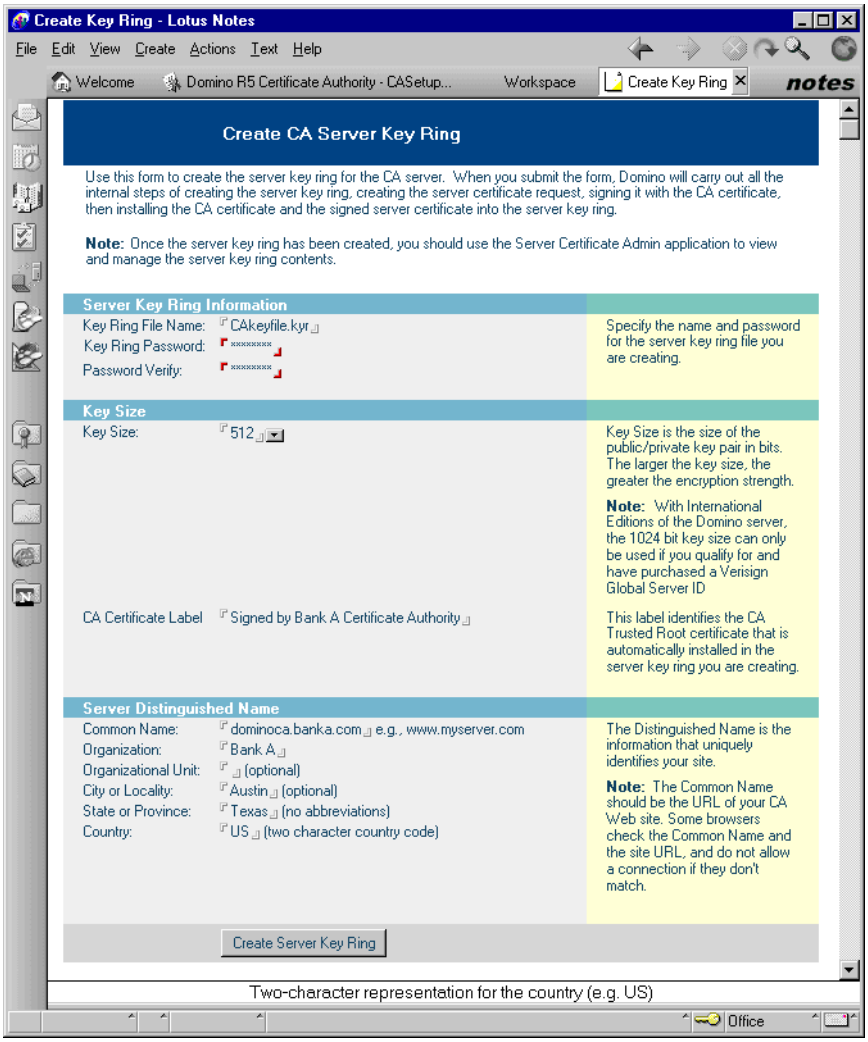

Figure 112. Creating a CA server key ring

3. Click **Create Server Key Ring** when the form is complete.
4. A password box will appear. Enter the password for the CA key ring (master key).
5. A window as shown in Figure 113 will now appear. Read it well before closing it.

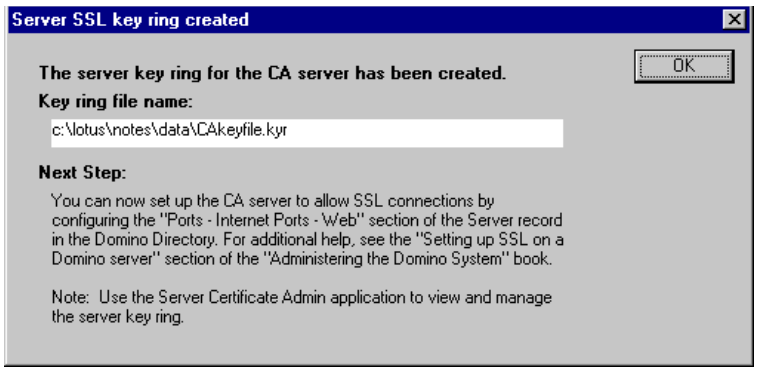

Figure 113. CA Server SSL key ring window

6. Specify the location for the key ring file in a directory accessible to the CA Server, then click **OK**.

8.4 Setting up the Web server

The server with the Domino CA can be set up as a Domino Web server to receive and post certificate requests.

1. Go into the Administrator's Console and edit the server information for the Domino CA server. Select the **Ports -> Internet Ports** tabs. The relevant portions of this tab are shown in Figure 114.

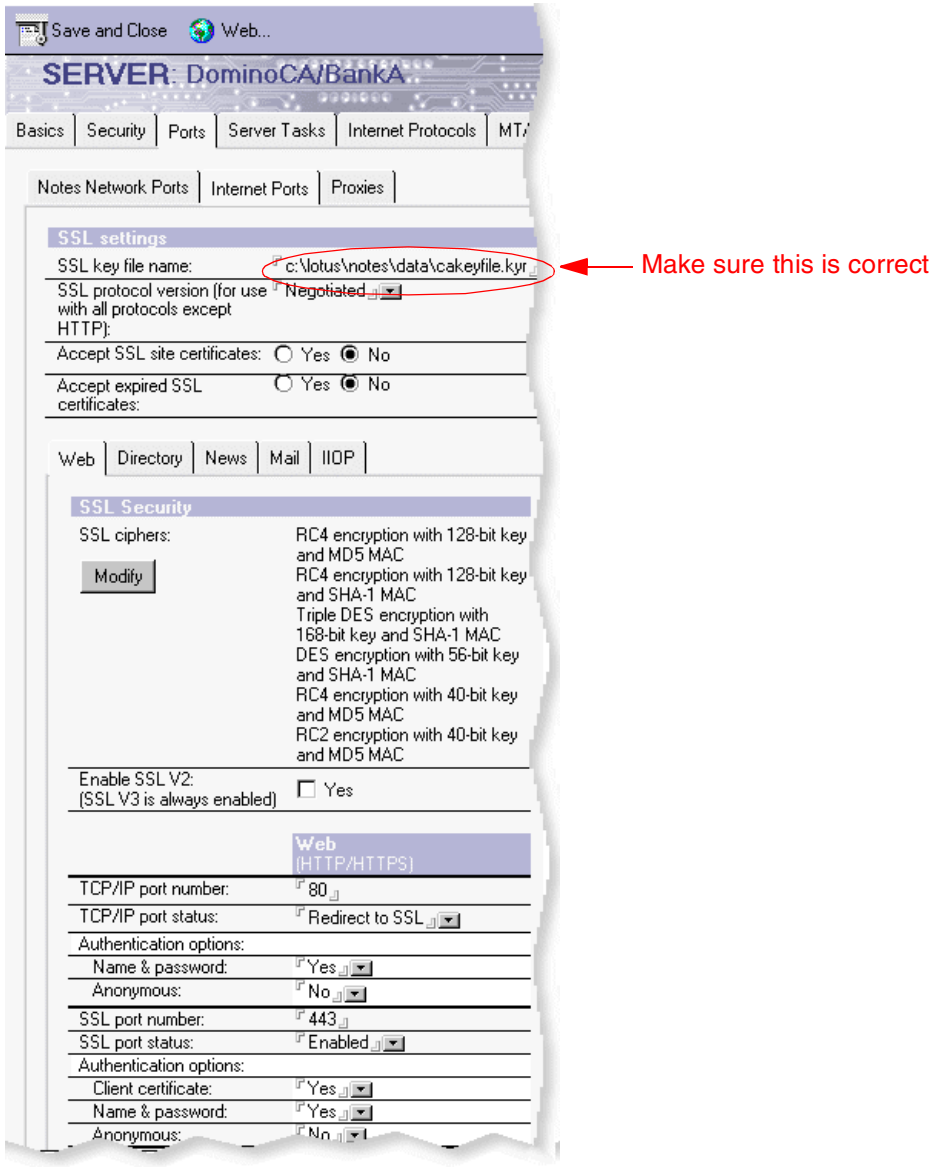

Figure 114. Internet port setting for Domino CA server

2. Fill out the form shown in Figure 114 using the information shown in Table 10.

Table 10. Web server security setup form values

Field	Value
SSL key file name	This is the key file used for the CA server. It was created in 8.3.3, “Create the key ring and certificate for the CA server” on page 178. It is *not* the Key Ring file for the Certificate Authority “master key”. Include the directory relative to the CA server where the file is located.
Accept expired SSL certificates	Set it to No to deny clients with expired certificates the ability to access the server.
Enable SL V2	Yes allows users with old browsers the ability to connect using SSL V2. Many browsers written in 1995 and earlier do not support SSL V3. **Note:** SSL V2 does not have the ability to support client authentication.
TCP/IP port number	Enter the port that Domino uses to listen for normal unencrypted HTTP traffic. The default is 80.
TCP/IP port status	The choices are Enabled, Disabled, or Redirect to SSL. Choosing Redirect to SSL will send all requests for the TCP/IP port to the SSL port. This setting is recommended.
Name & Password	The choices are Yes and No, and applies only if the TCP/IP port status is set to Enabled. Setting it to Yes will prompt the user for his user name and Internet password.
Anonymous	The choices are Yes and No, and applies only if the TCP/IP port status is set to Enabled. Choosing Yes will allow users who are not in the Names and Address book to access the Web site. If Yes is selected for both Anonymous and Name and Password, Domino will first try to authenticate the user with the name and password. If this fails, Domino will try to connect the user anonymously.
SSL port number	This is the port that Domino uses to listen for SSL requests. The default is 443.
SSL port status	The choices are Enabled, Disabled. Enabled is recommended.
Client certificate	The choices are Yes and No, and applies if the SSL port status is set to Enabled. **Note:** This scenario used client certificates for authentication; therefore, Yes was selected.

Field	Value
Name and Password	The choices are Yes and No, and applies if the SSL port status is set to Enabled. If both Client certificate and Name and Password are set to Yes, Domino will attempt to validate the user by the user ID and Internet password only if a client certificate is not available. Choosing No will allow users not listed in the Names and Address Book to request a certificate.
Anonymous	The choices are Yes and No, and applies if the SSL port status is set to Enabled. Choosing Yes will allow users who are not in the Names and Address book to access the Web site. If Yes is picked for Client certificate, Name and Password, and Anonymous, Domino will attempt to validate the user by the Client certificate and then the user ID and Internet password. If this fails, Domino will try to connect the user anonymously. **Note:** The recommendation is to set anonymous to No.

3. Save this information.
4. Verify the HTTP settings under the **Internet Protocols - HTTP** tab (Figure 115).

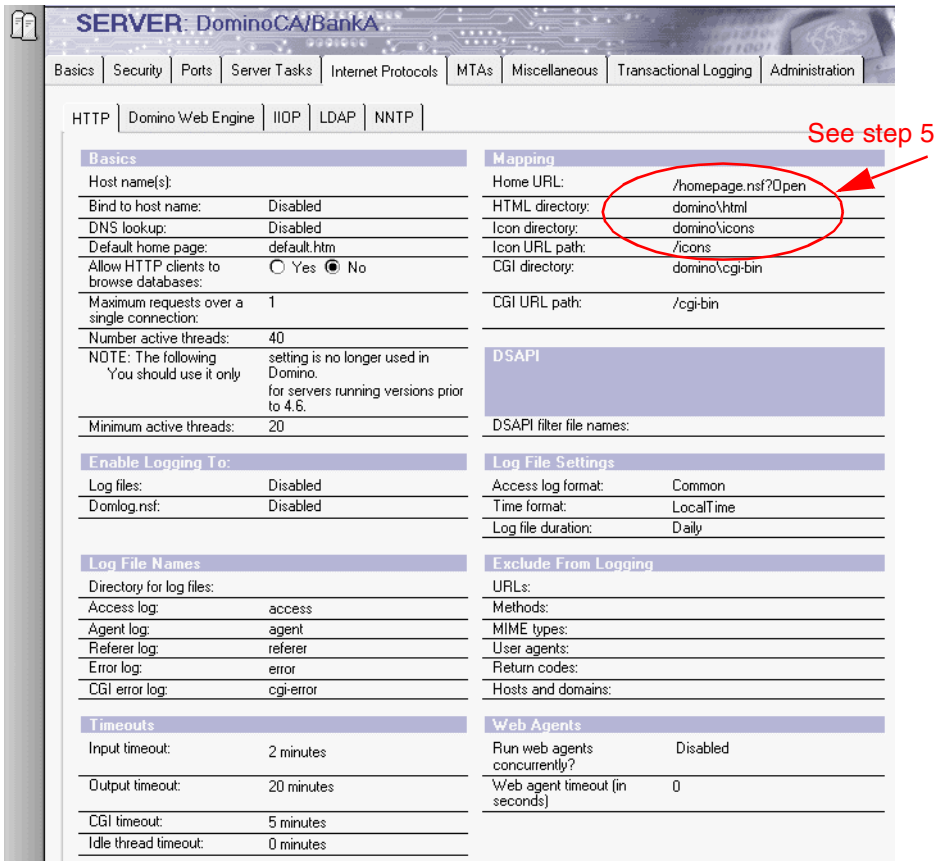

Figure 115. Domino - HTTP mapping

5. The certca.nsf database needs to be accessible from the Web. Check the HTML directory settings to make sure this is possible.
6. Start or restart the HTTP Web service task. This can be done under the server tab in the Administrator client. If any of the security settings are changed, restart the Domino server instead. The HTTP Web server task can be started automatically, when the Domino Server starts up. This setting is in the notes.ini file under Server Tasks as shown in Figure 116.

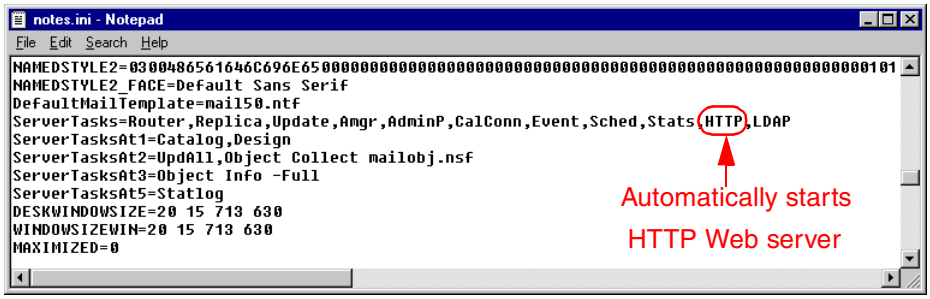

Figure 116. Notes.ini file that autostarts HTTP Web server

8.4.1 Create the key ring and certificate for the DMZ server

Next, you must create a key ring and certificate to be used by the Host On-Demand and Domino Web server. This is done by the Server Certificate Admin database, which is created during server setup.

The Server Certificate Admin database is used to perform the following:

- Create and manage the server certificate and key ring file, which holds the server certificate.
- Send a request to either an internal or external CA to sign our server certificate.
- View requests that are submitted to the CAs.
- Add the CA's certificate as a trusted root to the server certificate.
- View information about certificates in the key ring file.
- Control client access to the server by adding or removing trusted root certificates from the key ring file.

The Server Certificate Admin database should be on the Host On-Demand and Domino Web server. Find the icon on your workspace similar to the one shown in Figure 117.

Figure 117. The Server Certificate Admin icon

Each Domino Web server has its own unique Server Certificate Admin database. Make sure the one on the correct server is used.

If the database is already created, skip to 8.4.1.2, “Creating the key ring and certificate” on page 188.

8.4.1.1 Creating the Server Certificate Admin database

If the Server Certificate Admin database is not on the Domino Web server, it must be created. The following steps must occur on a Domino Designer client with an ID with server administrator permissions:

1. From the menu bar, select **File -> Database -> New**.
2. Use the information found in Table 11 to fill out the form shown in Figure 119.

Table 11. Server Certificate Admin database form values

Field	Value
Server	The name of the Domino Server - Ours is DmzDomino/BankA
Title	Server Certificate Admin
File name	certsrv.nsf
Template server	The name of the Domino Server - Ours is DmzDomino/BankA
Show advanced templates	Select this box to be able to find the correct template
Server Certificate Admin	Highlight this template

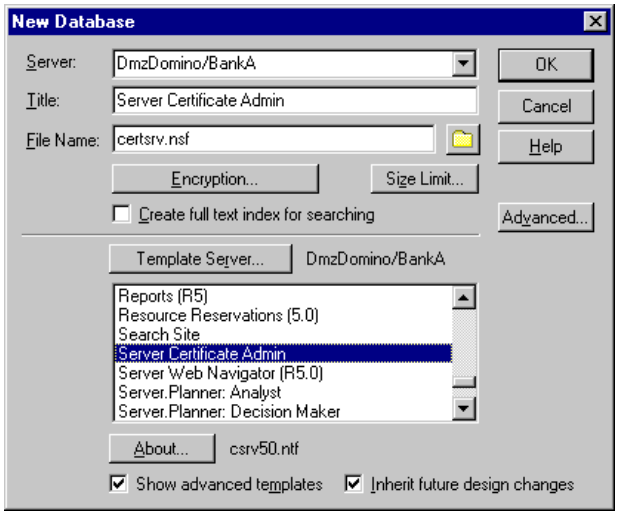

Figure 118. Creating the Server Certificate Admin database

3. Click **OK** to create the Server Certificate Admin database.

8.4.1.2 Creating the key ring and certificate

Now you are ready to create the key and certificate for the Host On-Demand and Domino Web server.

1. Open up the Server Certificate Admin database and the window shown in Figure 119 will be displayed.

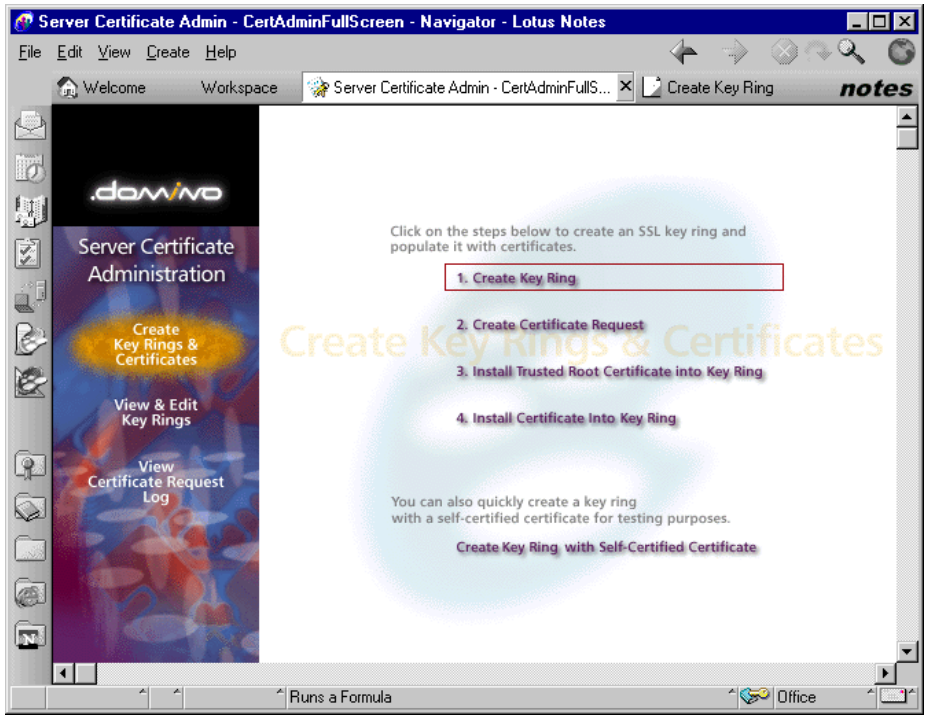

Figure 119. Server Certificate Admin database

2. Select **Create Key Rings & Certificates** in the left pane and choose **1. Create Key Ring**.
3. Use the information from Table 12 to complete the form shown in Figure 121.

Table 12. Create key ring form values

Field	Value
Key ring file name	Specify the file name and location for the CA key ring. The default directory is the data folder for the Notes client (usually x:\lotus\notes\data). The default file name is Keyfile.kyr. It is suggested that you use the server name in the key ring file name.
Key ring password	Specify and confirm the password for the CA key ring.
Key size	Specify the size (in bits) of the public/private key for the SSL key ring on the server. The choices are 512 or 1024. The default is 512.

Field	Value
Distinguished name	The distinguished name identifies the certificate for the Host On-Demand and Domino server. The common name field should be populated with the DNS host name of the Web server. For some browsers, the common name in the certificate will be compared to the host name in the URL being requested. If there is a mismatch, a warning is shown. The other entries are to further distinguish this server from others.

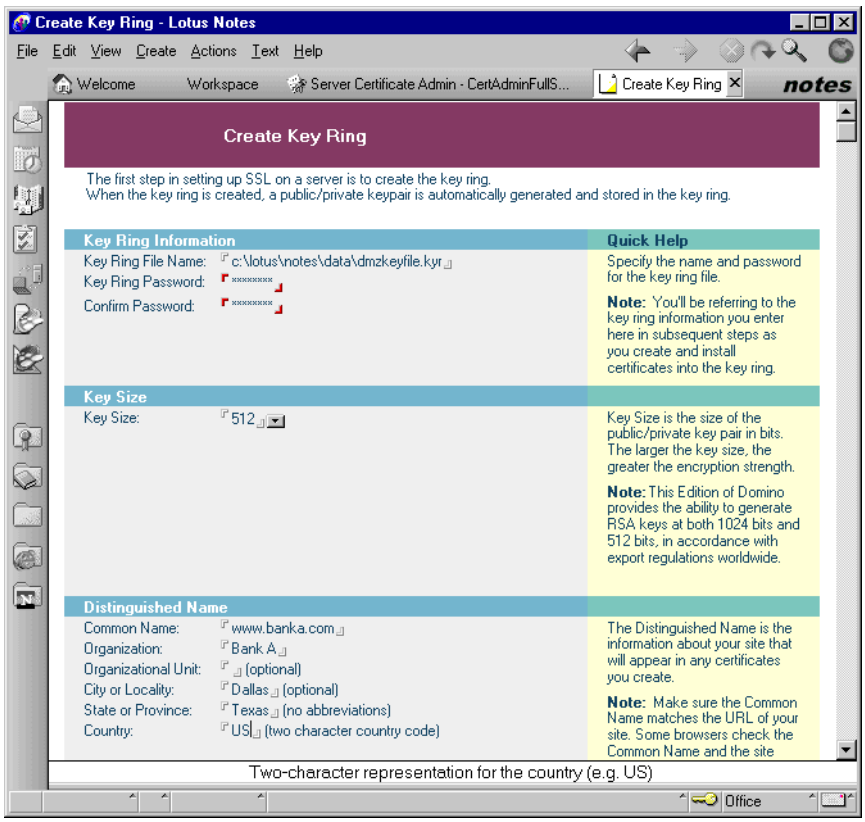

Figure 120. Creating the Host On-Demand/Domino server key

4. Click **Create Key Ring w**hen the form is completed and the key ring will be created. The confirmation window as shown in Figure 121 will be displayed.

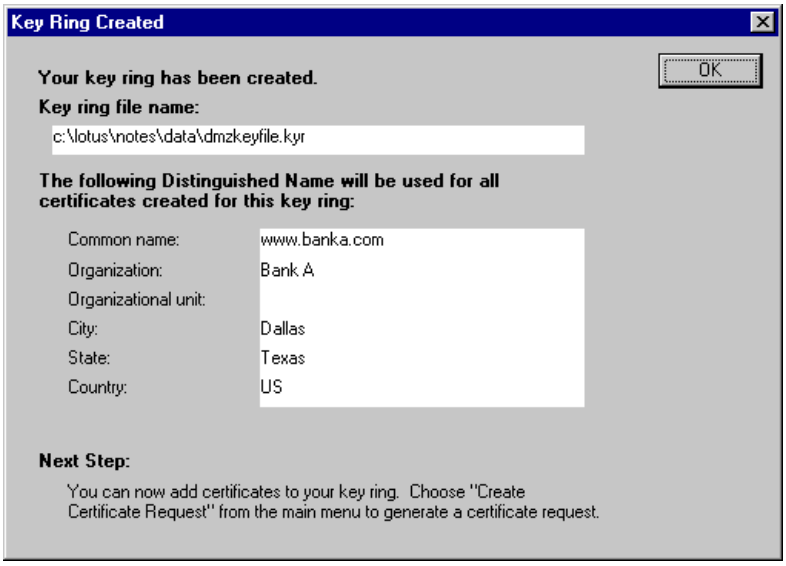

Figure 121. Key ring created confirmation

5. Click **OK**.

You now have an SSL key ring with an unsigned certificate. You need to get your certificate signed by a trusted Certificate Authority before you can do SSL.

8.4.2 Create Certificate Request

You will now get your certificate signed by the Certificate Authority, your Domino Certificate Authority.

1. First select **Create Key Rings & Certificates** in the left pane and then select **2. Create Certificate Request** as shown in Figure 119 on page 189.
2. Complete the Create Server Certificate Request document using the information given in Table 13.

Table 13. Create server certificate request form values

Field	Value
Key Ring File Name	Specify the SSL key ring file and location to be signed.
Log Certificate Request	There is an option to log the request. The default is Yes.
Method	The key can either be pasted into a form or sent to the CA by e-mail.

When completed, the document will look similar to that shown in Figure 122.

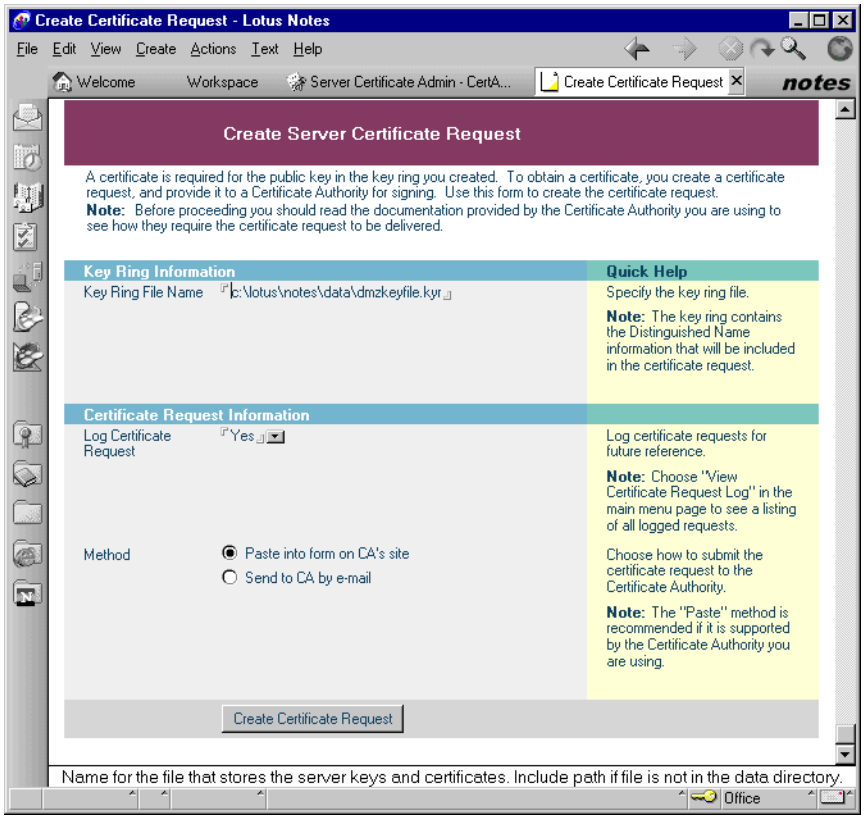

Figure 122. Create server certificate request

3. Click **Create Certificate Request**.
4. Enter the password for the key file when requested and click **OK**.
5. A window titled Certificate Request Created will be displayed as shown in Figure 123. Your distinguished name as it will appear in the certificate is displayed in the top pane for confirmation. You must now select all of the text in the bottom pane and copy it to the clipboard. You *must* include the `-----BEGIN NEW CERTIFICATE REQUEST-----` and `-----END NEW CERTIFICATE REQUEST-----` lines for the request to be valid. Finally, click **OK** to be returned to the server certificate administration main menu.

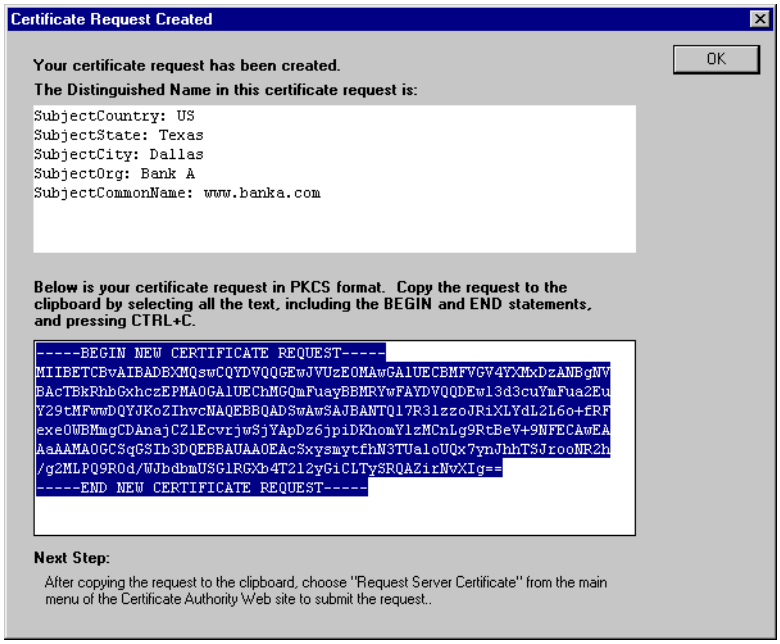

Figure 123. Certificate request key

6. Bring up a Web browser and select the URL of the Domino CA in the directory certca.nsf. (For example `https://dominoca.banka.com/certca.nsf`)
7. There might be some warnings about your browser not recognizing the authority that signed the certificate. For information on these windows, see Appendix C, “Browser operations” on page 237. Click **Next, OK,** or **YES** to get through the windows. The options and number of windows are dependent on the browser being used.
8. A login window will appear. Log in with a user ID and Internet password for the ID. The box will look similar to Figure 124.

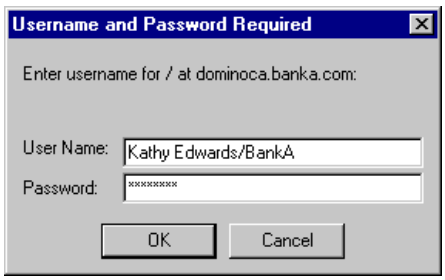

Figure 124. User name and password prompt

9. A browser window will appear. Select **Request Server Certificate** in the left pane. Fill in the information requested. The e-mail address you specify here will receive a note containing the location to pick up the signed certificate. Paste the key previously copied into the clipboard into the **Certificate Request** box. The resulting window will look that shown in Figure 125.

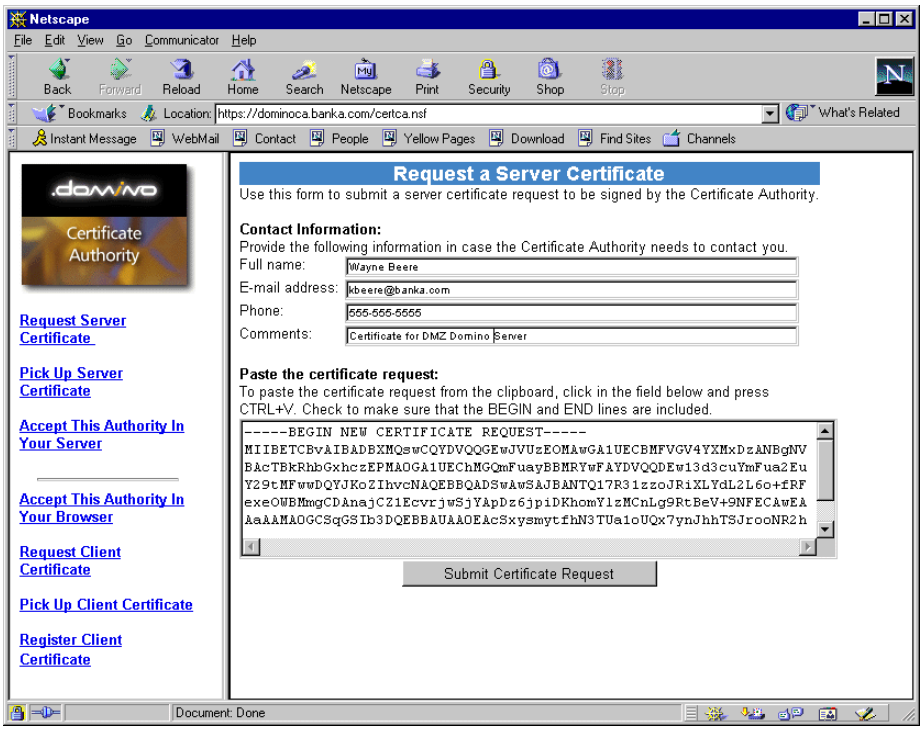

Figure 125. Request a Server Certificate window

10. Click **Submit Certificate Request**.

11. A window like the one shown in Figure 126 will be presented.

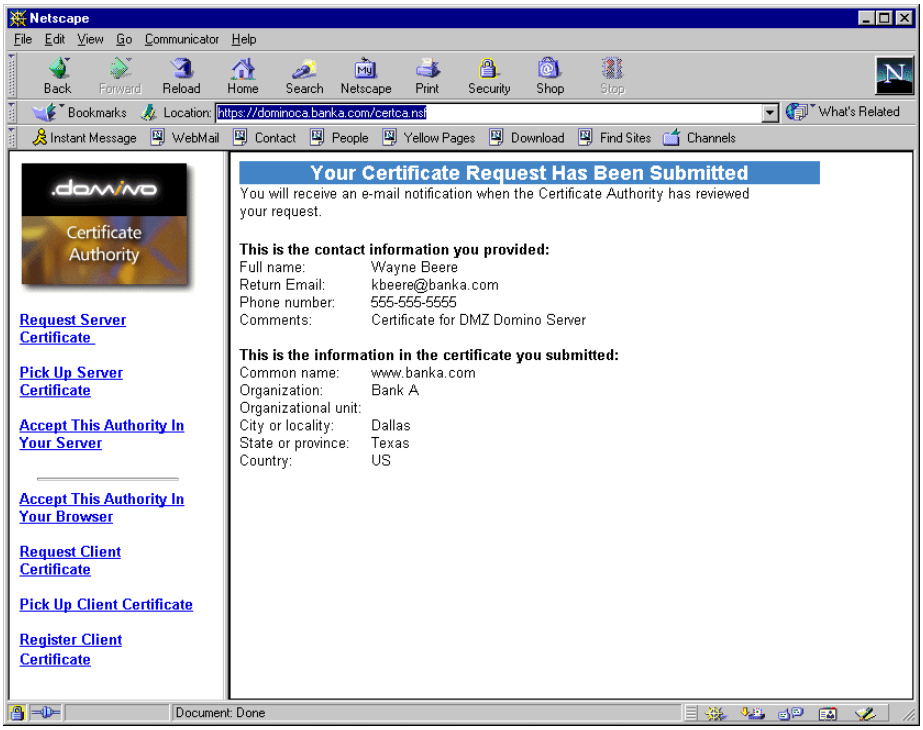

Figure 126. Submitted certificate request

8.4.3 Approval and signing the certificate

The administrator of the Certificate Authority must determine whether to trust our certificate, and if so, to sign our certificate and return it.

1. The administrator must enter the Domino R5 Certificate Authority database (the icon looks like Figure 107 on page 172) and select **Server Certificate Requests**. The request will be in the window.
2. The administrator must review the certificate request and if approved, click **Approve** (see Figure 127), then enter the CA “master key” password.

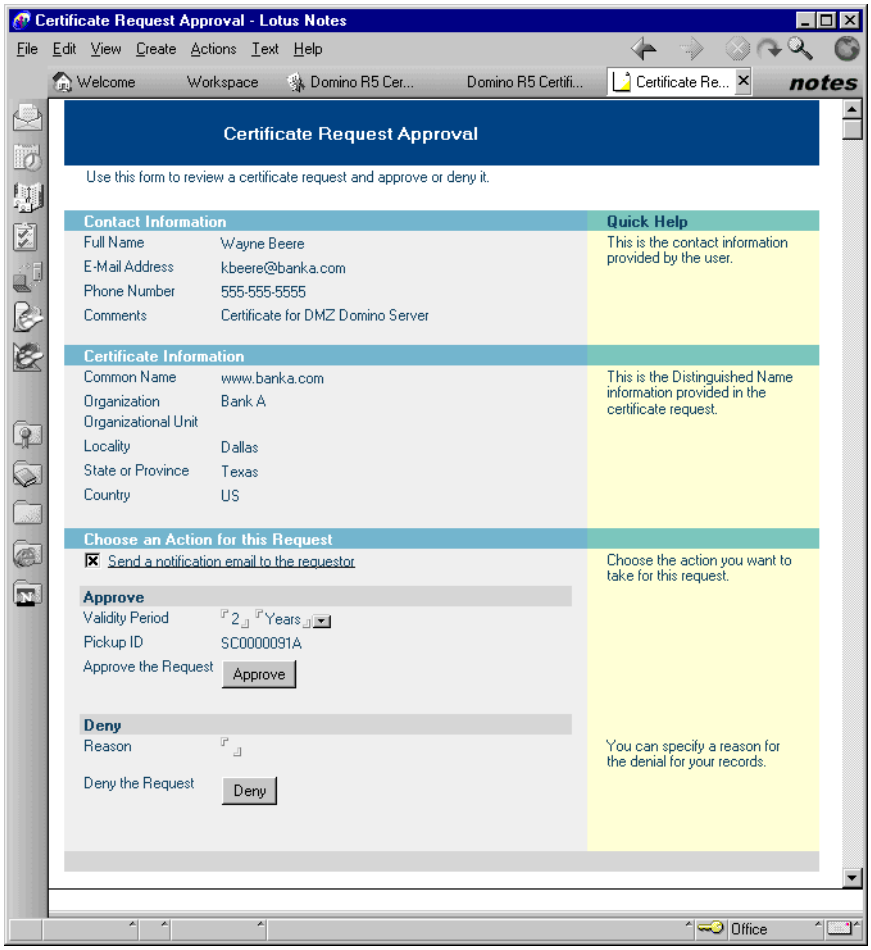

Figure 127. Certificate request approval

8.4.4 Pick up the certificate for the Domino CA server

Now the Web server administrator must receive the signed public key.

1. On the Domino Web server, bring up a Web browser and select the URL of the Domino CA in the directory certca.nsf (for example `https://domihnoca.banka.com/certca.nsf`), and a window like the one shown in Figure 128 will be shown.

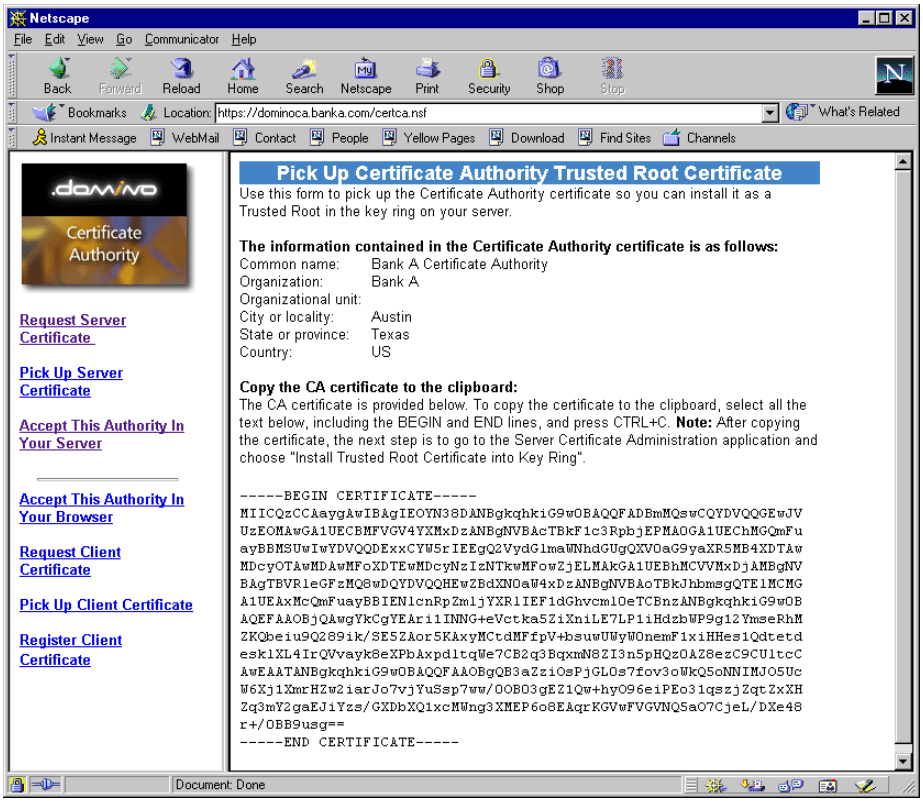

Figure 128. Pick up Certificate Authority trusted certificate

2. Follow the instructions in the window to copy the certificate to the clipboard, then open the Server Certificate Admin database on the Host On-Demand Domino Web server and select
   Create Key Rings and Certificates ->
   3. Install Trusted Root Certificate into Key Ring.
3. A document called Install Trusted Root Certificate will appear. The instructions displayed are self explanatory (see Figure 129).

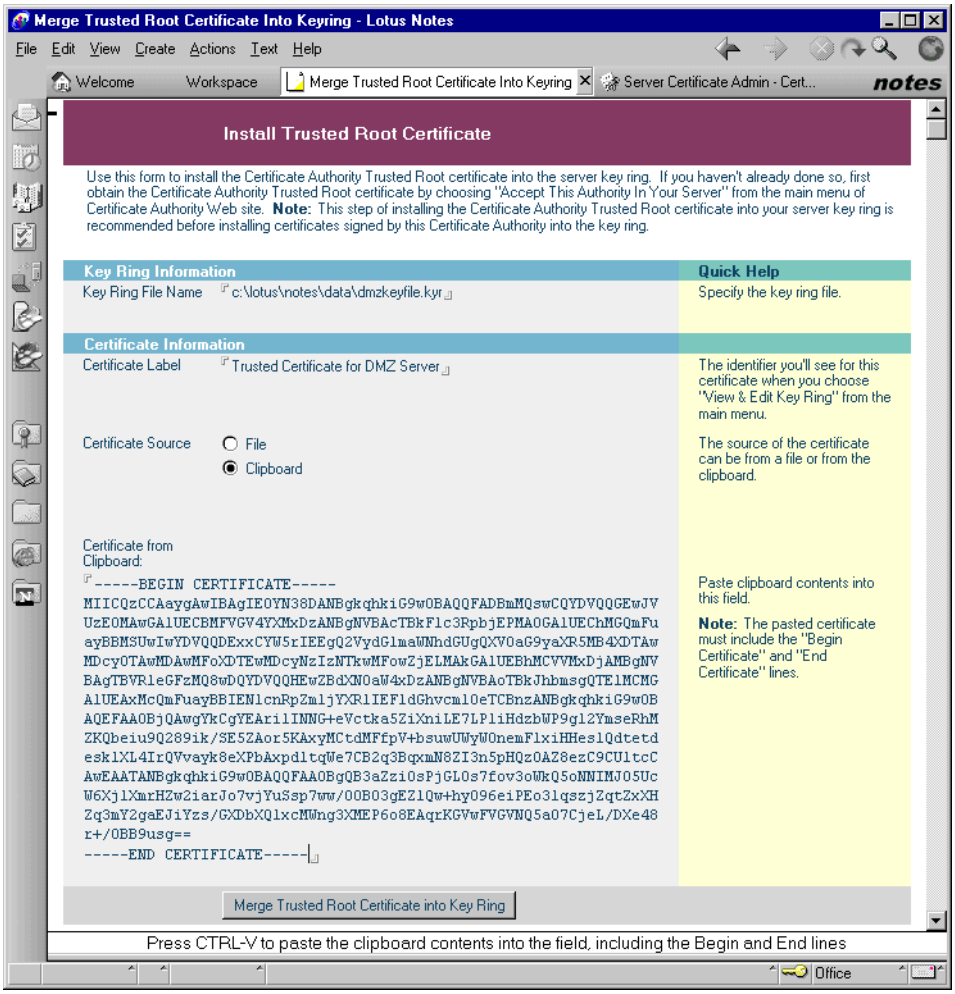

Figure 129. Install trusted root certificate

4. Click **Merge Trusted Root Certificate into Key Ring**, and a confirmation window (see Figure 130) will appear. Read it and click **OK**.

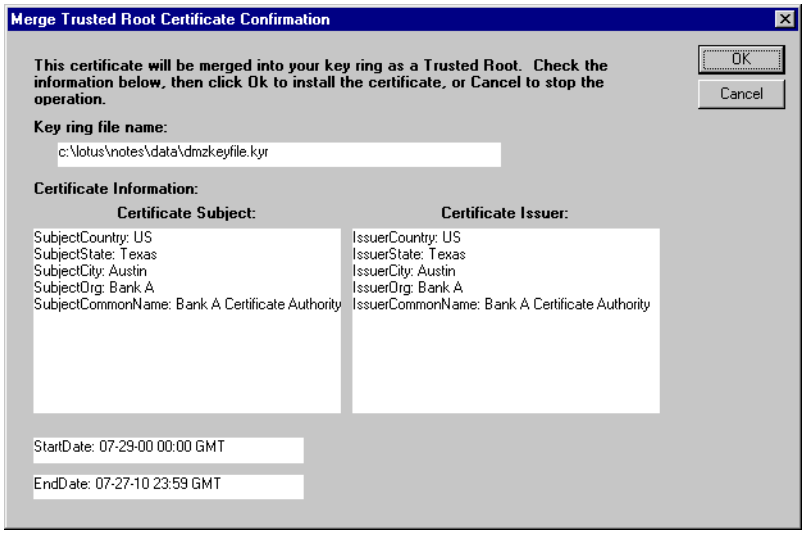

Figure 130. Merge trusted root confirmation

5. Another window will appear. Read it and click **OK**.

8.4.5 Pick up the signed certificate

1. Go back into your mail. Bring up the e-mail approving the certificate request. It will look similar to the one shown in Figure 131.

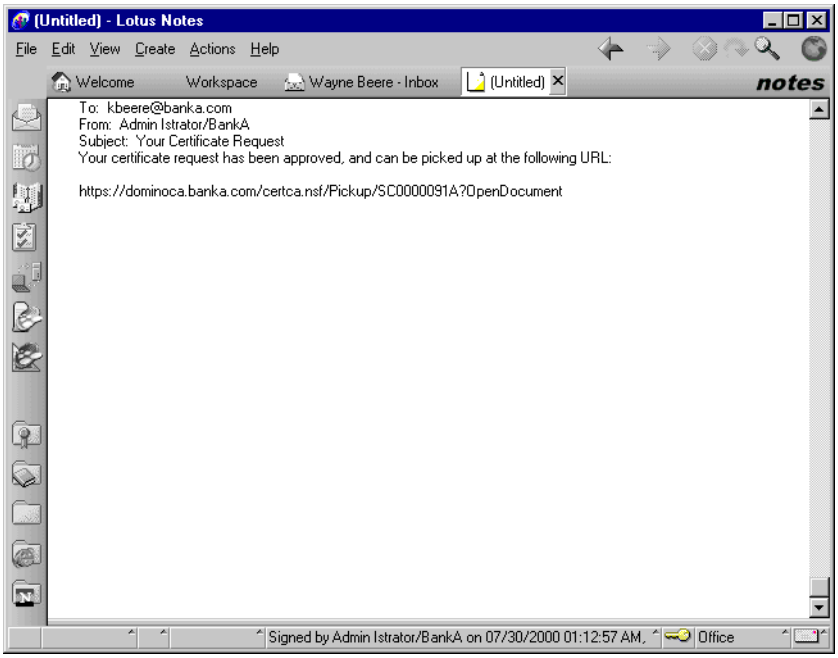

Figure 131. Signed certificate notification

2. Select the Web page listed. The administrator needs to log in over an SSL connection. Once logged in, the administrator can access any window for which they know the number (SC0000091A in our example).
3. Follow the instructions in the window (see Figure 132) and copy the certificate to the clipboard.

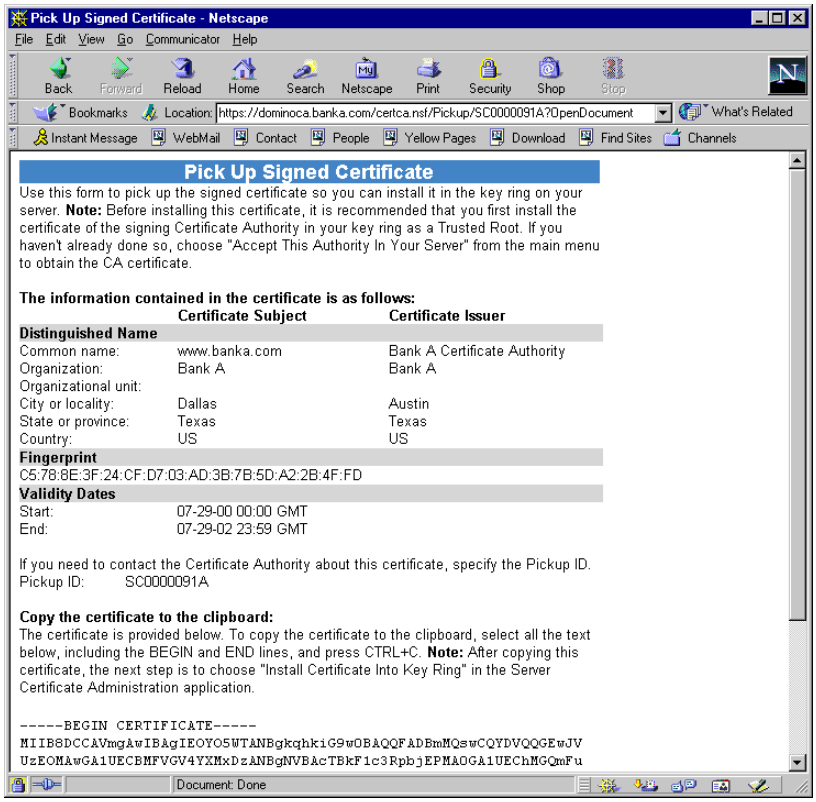

Figure 132. Pick up signed certificate

4. Open the **Serve Certificate Admin** database on the Domino and Host On-Demand server, and select **Install Certificate Into Key Ring** in the Create Key Rings & Certificates view.
5. Verify the Key Ring File Name is correct, then paste the certificate from the clipboard into the correct field as shown in Figure 133.

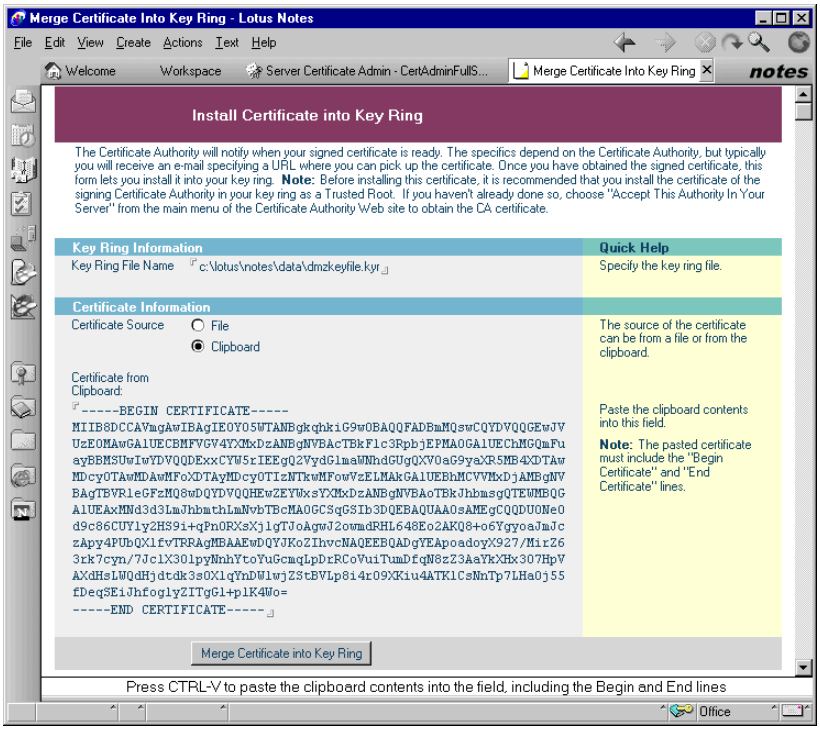

Figure 133. Install certificate

6. Click **Merge Certificate Into Key Ring**.
7. Enter the password.
8. Review the information in the window that appears and click **OK**. The resulting window is shown in Figure 134.

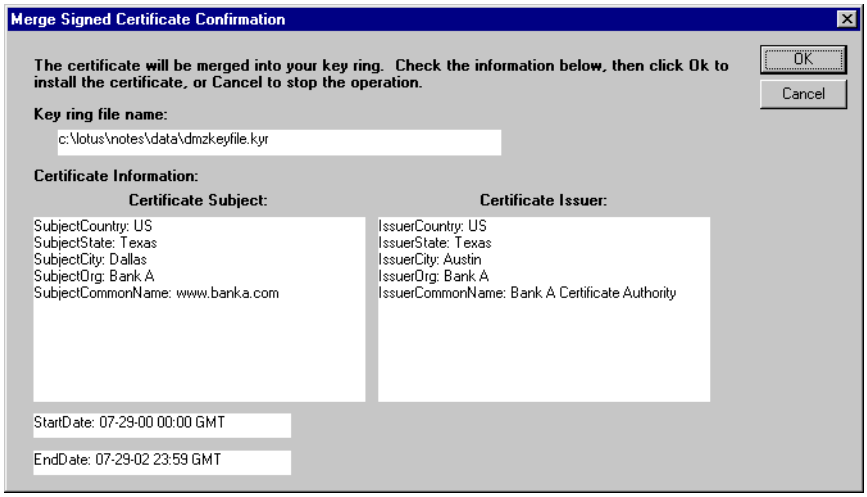

Figure 134. Merge signed certificate

9. Read the information in the window that appears and click **OK**. It suggests enabling SSL on the server. This can be done later.

8.5 Install Host On-Demand and configure the Domino Web server

The installation of IBM WebSphere Host On-Demand on the Domino Web server does not automatically configure the Web server. This section will cover how to enable Host On-Demand and the HTTPS setup on the Domino Web server.

You must configure the Domino Web server for Host On-Demand over an SSL connection. Refer to 8.4, “Setting up the Web server” on page 181 for complete instructions on setting up the Web server. There are a few specifics for this implementation that you must be aware of.

8.5.1 SSL security settings

This scenario uses the X.509 client certificate for authentication; therefore, make sure that the SSL Authentication options are set as follows:

- Client Certificate: **Yes**
- Name & password: **No**
- Anonymous: **No**

These specifications insure that only a user with a recognized client certificate will be allowed access to the Web server.

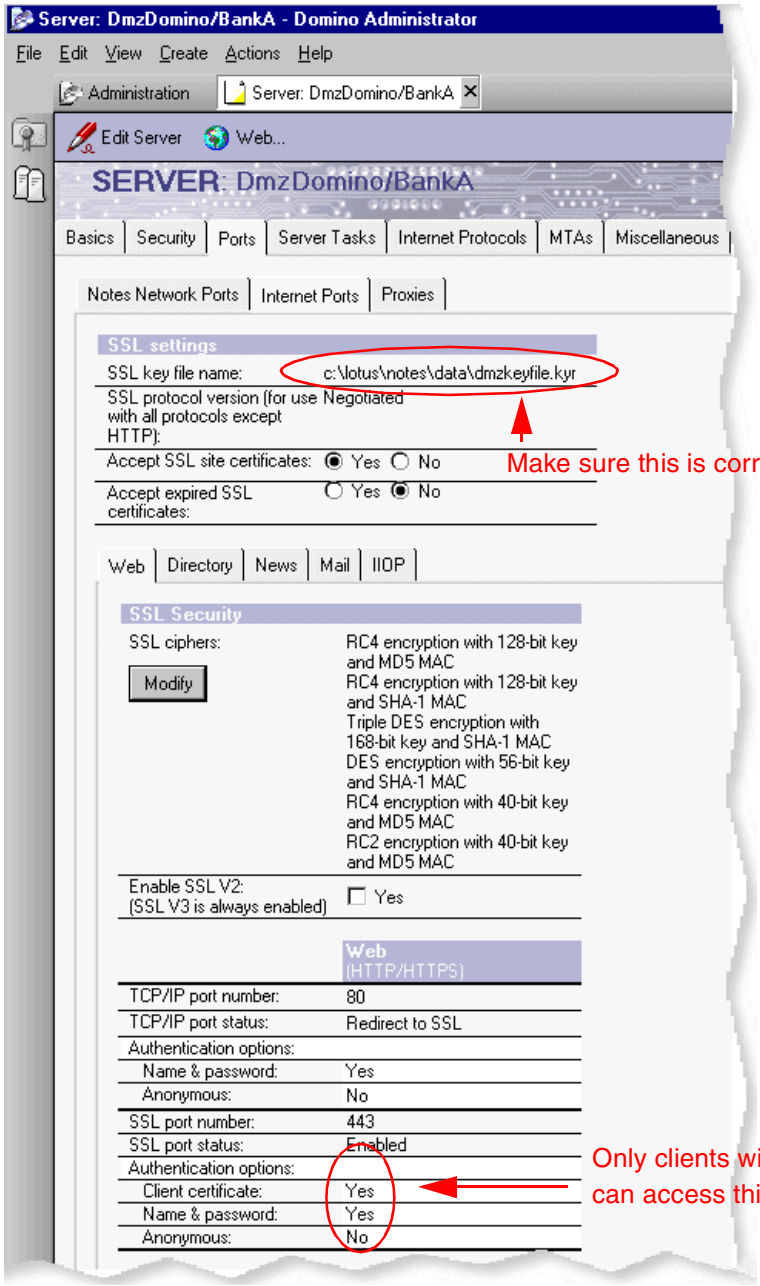

Figure 135. Domino Web server security

8.5.2 HTTP setup for Host On-Demand

You must manually configure the HTTP settings under the Internet Protocols - HTTP tab. Some of the settings that need to be altered for Host On-Demand are:

- **Host name**

  Enter the DNS host name of the Web server. It is very important that it be the same as the Common Name field specified in the server's certificate. Refer to Figure 120 on page 190.

- **Default home page**

  Enter in the location and file name of the default home page. This will be the page that comes up if no page is specified in the URL. You could assume the basic Host On-Demand download client as shown in Figure 136. Another choice could be c:\hostondemand\hod\HODMain.html.

- **HTML directory**

  The location of the directory to publish on the Internet. For Host On-Demand, the default is c:\hostondemand\hod.

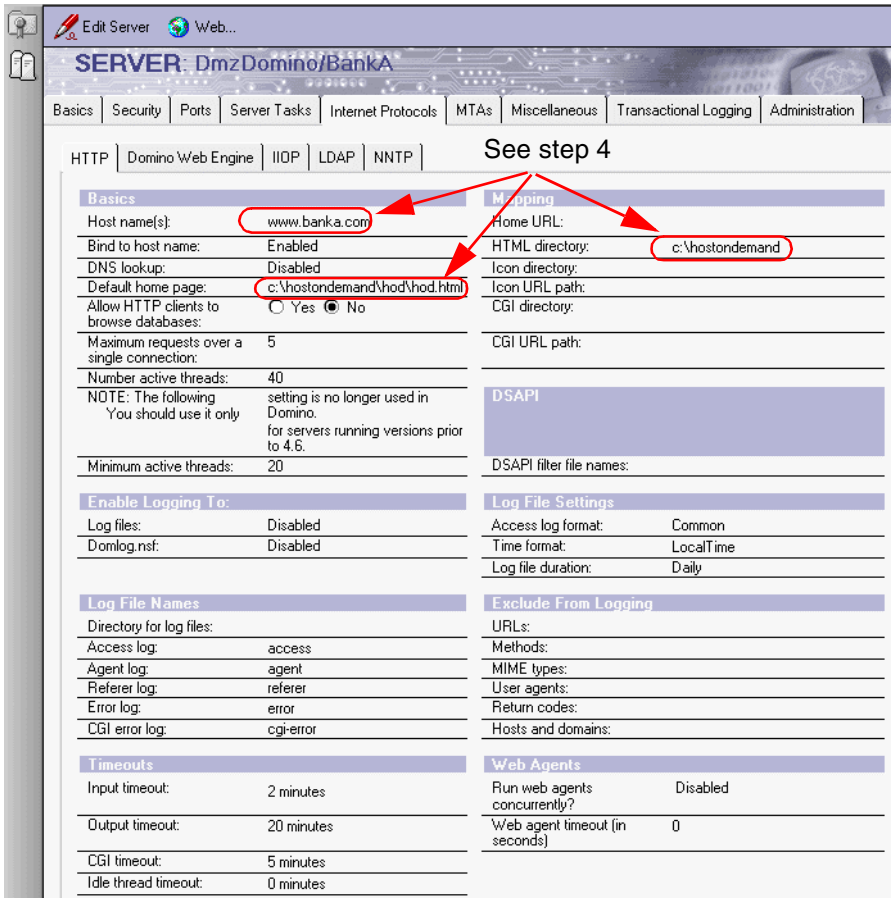

Figure 136. Domino Web server - HTTP setup

When these changes have been made, save the information, then start or restart the HTTP Web service task. This can be done under the Server tab in the Administrator client.

8.6 Setting up the clients

The users also need a signed certificate to connect to the Host On-Demand server and to establish a Telnet session. This signed certificate will be used to prove the user's identity to the server. Since our Domino CA is not a well-known CA, the users must also accept our CA's authority. There are a few ways this can be done, two of which will be covered.

Signed certificates can be distributed using Lotus Notes. A user would receive the certificate, merge the certificate into their browser, and then connect to a Web site to accept the CA's authority. The user can then use the certificate with Host On-Demand to Telnet to the host. This method works for Notes users with a user ID.

Certificates can be distributed through a Web site. Users can go to the Web site to request a signed certificate and accept the CA's authority. After an administrator signs the certificate, the user would merge the certificate into their browser. If the administrator has not already registered the client, a client can register their client certificate. This method works for Notes users and non-Notes users.

For illustration purposes, one method is shown using Netscape and the other method with Internet Explorer; however, both methods can be used with either browser.

8.6.1 Distributing certificates through the Notes user ID

The process for receiving a signed certificate through Notes is the same for Netscape and Internet Explorer. The process for merging the private certificate and the CA public key into the browsers is different.

8.6.1.1 Register the User and Send the Certificate

In order to receive a signed certificate through Lotus Notes, a user needs to have a user ID, a Notes mail database, and an Internet password linked to his user ID. The Internet password can be assigned in the Domino Directory.

The following procedure needs to take place on a machine with the Domino Administrator client.

1. If necessary, register a person. It works best if the person is on the same Notes domain as the CA database server. If not, each domain needs to be defined in the other's database and mail routing needs to be set up between the two domains. A single administrator needs to have administrator privileges on both domains and be able to access the domains from a single workstation. Select the check box for **Set Internet Password** shown in Figure 137.

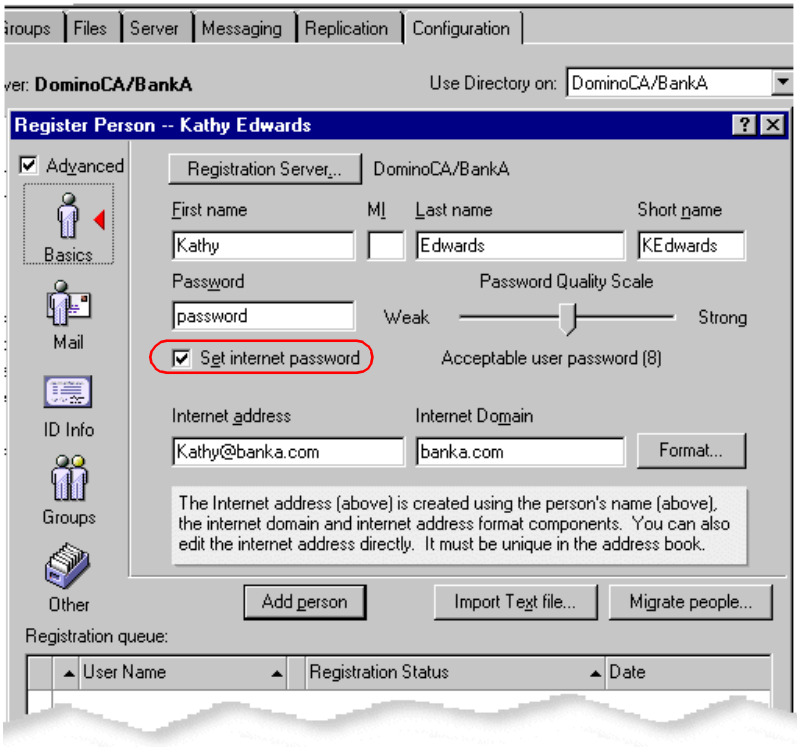

Figure 137. Domino - set Internet password

2. In the Domino Server Administrator, view all the people in a domain in the Domino Directory, and select all the names who will receive a signed certificate.
3. Select **Actions - Add Internet Cert to Selected People**. The result is shown in Figure 138.

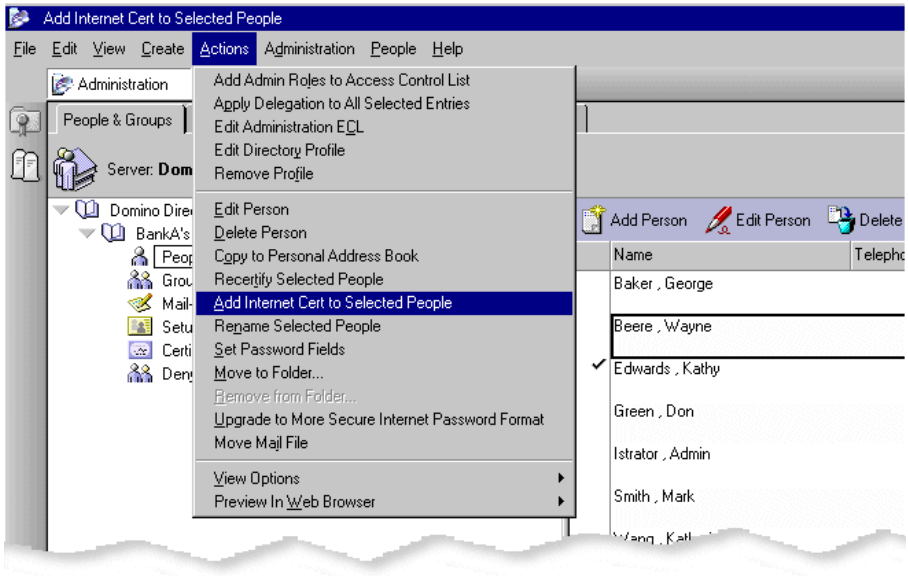

Figure 138. Domino - add Internet certificate

4. A window will appear requesting a key ring file. Select the Certificate Authority key ring file created in 8.3.1, “Create the Certificate Authority key ring and certificate” on page 173. Enter the password for the key ring file when prompted.

5. A window similar to the one shown in Figure 139 will pop up. Enter the certificate expiration date and time. Click **Certify** to proceed. A window should appear stating the request has been processed. Click **OK.**

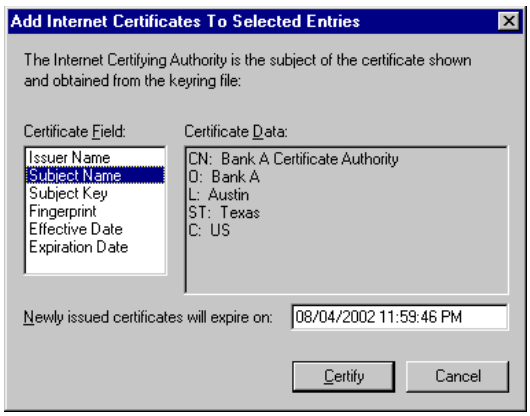

Figure 139. Domino - Internet certificate expiration

8.6.1.2 Receive and export the Certificate

The user needs to receive the CA certificate into his or her ID file and then export it to disk where it will be used later by the browser and the Host On-Demand applet. The user will perform the following steps on his workstation:

1. Open the user's Notes mail database. This will import the CA certificate into the user's ID file.
2. Select **File - Tools - User ID...**. Enter in the user ID password when prompted.
3. A window will appear as shown in Figure 140. The importing of the CA certificates can be verified by choosing **Certificates** and scrolling down in the area titled Certificate Issued By:.

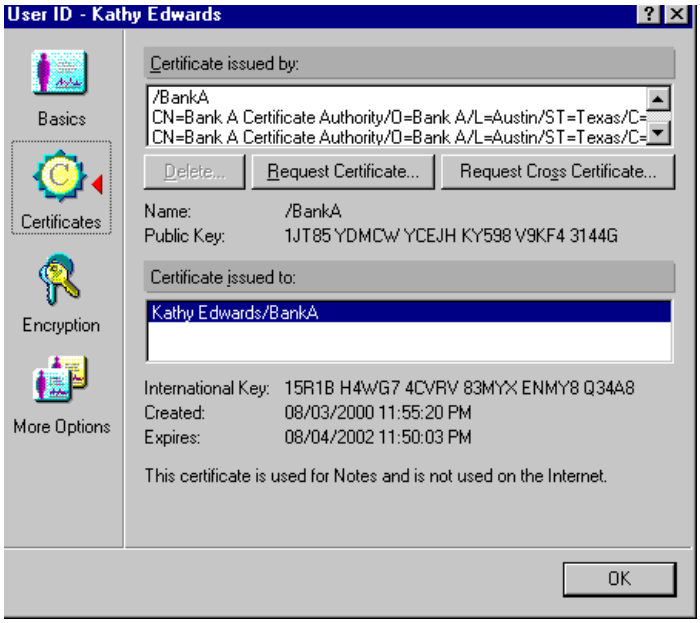

Figure 140. Notes user certificate

4. Select **More Options - Export Internet Certificates...** A window will come up showing all the available certificates to export. It will look similar to the window shown in Figure 141.

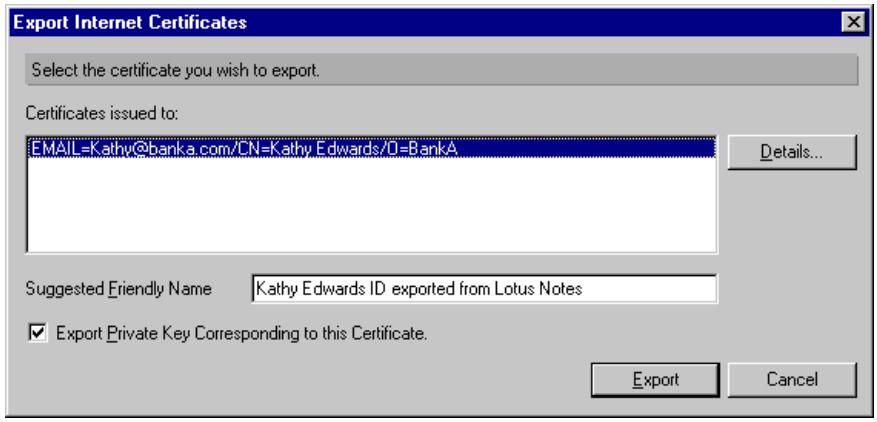

Figure 141. Notes - export Internet certificate

5. Select the certificate, and enter in a name to identify it. Select the check box next to **Export Private Key Corresponding to this Certificate** as shown in Figure 141, then click **Export**..
6. A window will prompt the user to create a password for his or her certificate. The window suggests the password be at least 8 characters long.
7. A save file window will appear requesting a file name for the exported certificate. Save it as type PKCS12 Files (*.p12). This file will be needed to import into the browser and to establish a Telnet session.
8. After the certificate is saved, a window will appear. Click **OK**, then click **OK** again to close the User ID window.

8.6.1.3 Importing certificates into the browser

1. Launch a Web browser and go to the URL of the Domino CA in the directory certca.nsf, for example `https://dominoca.banka.com/certca.nsf`.
2. There might be some warnings about your browser not recognizing the authority who signed the certificate. For information on these windows, see Appendix C, “Browser operations” on page 237. Click **Next**, **OK**, or **Yes** while going through the windows.
3. A login window similar to the one in Figure 124 on page 193 will appear. Log in using the user ID and Internet password.

The next steps are dependent on the browser being used.

Netscape browser

Use the following instructions when importing certificates into the Netscape browser.

1. When the Web page appears (Figure 142), select **Accept this Authority in Your Browser** in the left pane and then select **Accept this Authority in Your Browser** in the right pane.

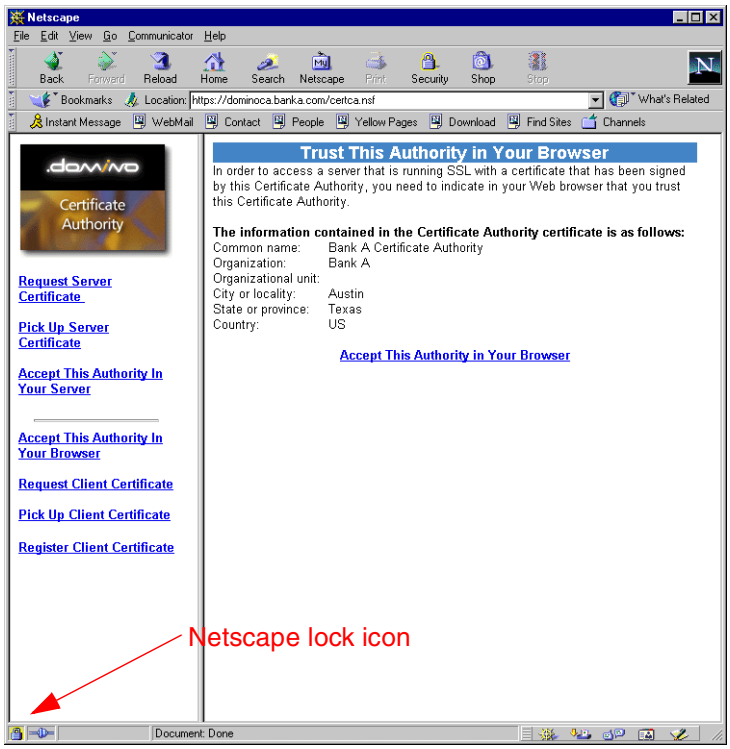

Figure 142. Netscape - trust the CA

2. Several windows will appear, and on each window you should click **Next**. The last window will ask the user to name the CA. The name entered here should be meaningful so that it will help the user identify the CA. Click **Finish.** For detailed information see Appendix C, “Browser operations” on page 237.
3. After accepting the authority, bring up the Security Info tool. This can be accomplished by selecting **Communicator -> Tools -> Security Info** from the menu, or by clicking the lock icon in the bottom left of the browser. A window like that shown in Figure 143 will appear.

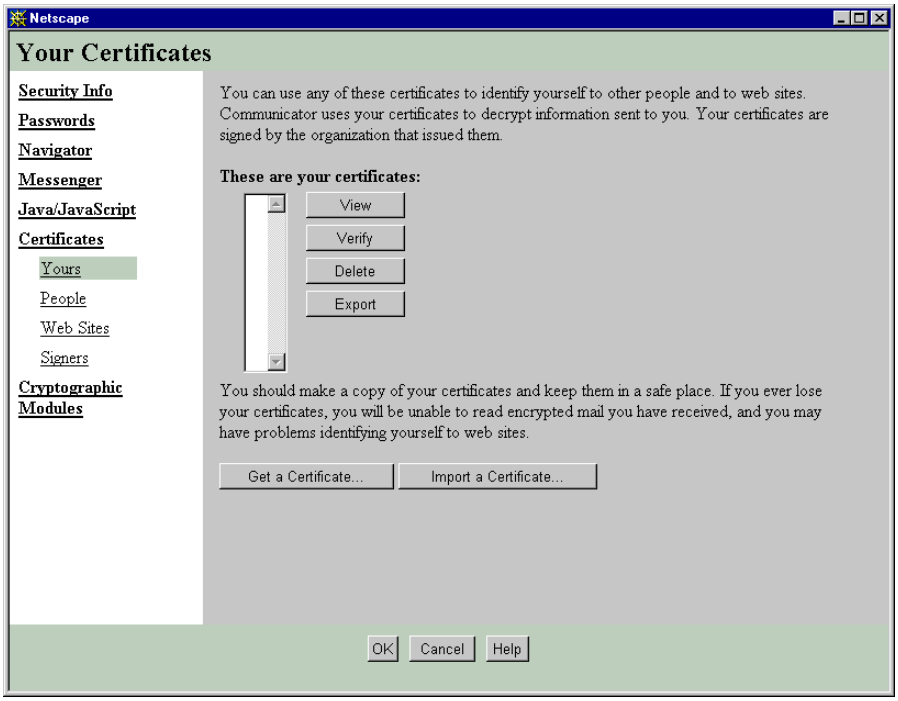

Figure 143. Netscape - security

4. Select **Certificates - Yours** in the left pane, then click **Import a Certificate**. If a password for Netscape's certificate database has not been chosen, a window will appear with the option of creating one.

   If created, the password will be needed every time a certificate is accessed in Netscape (including during client authentication, importing certificates, and exporting certificates). This password will not be used during Client Authentication in Host On-Demand. The password can be administered in the Security Info Tool by choosing **Passwords** in the left pane.

5. An open file window will appear. Select the certificate file that was exported in step 7 on page 211. The file will be of type PKCS12 Files (*.p12).

6. If requested, enter the password chosen in step 6 on page 211. A window will confirm the certificate has been imported.

7. To verify that the certificate was properly imported, launch the Security Info tool by clicking the lock icon. The CA's public certificate that was just

accepted can be seen under **Certificates - Signers**. The name will be the one given in step 2 on page 212 (see Figure 144).

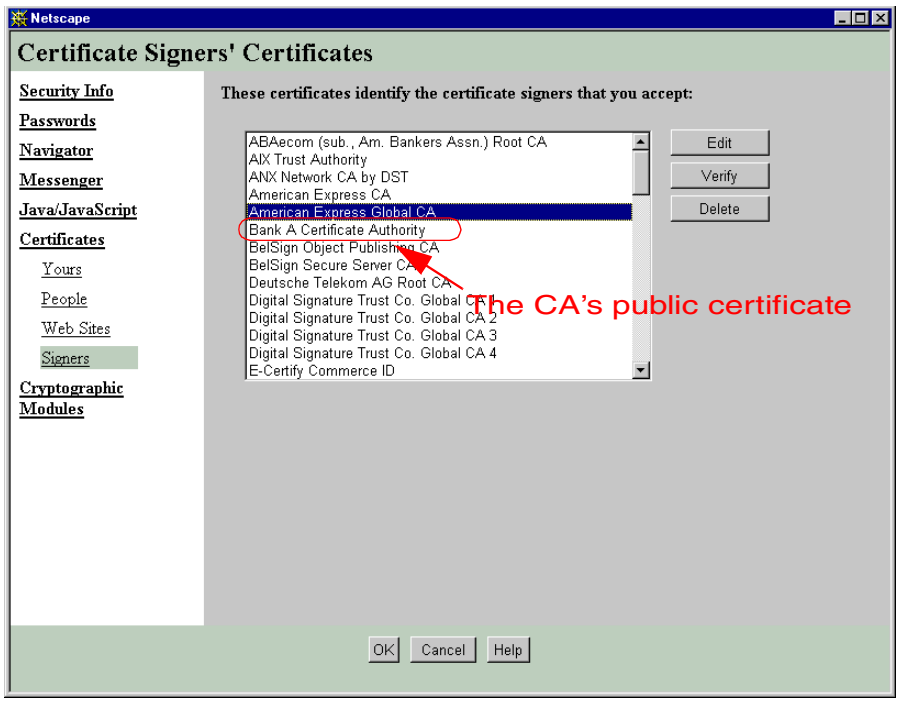

Figure 144. Netscape - signer certificates

8. To view the user's private certificate, choose **Certificates - Yours** as shown in Figure 145. The resulting name will be the one given in step 5 on page 211.

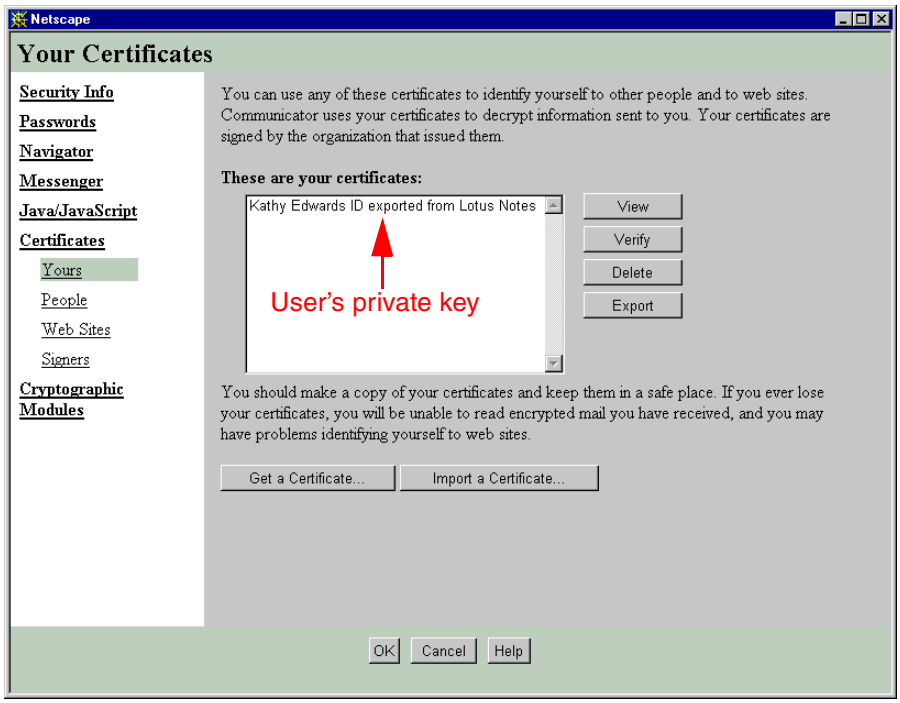

Figure 145. Netscape - personal certificate

9. Netscape has settings that help you control how Netscape will react to various security situations. See Figure 146. One option of particular interest is selection list in the box labeled Certificate to identify you to a Web site. To avoid Netscape prompting the user for the certificate each time he or she connects to the Web site, select **Select Automatically** or one of the user's private certificates from the list.

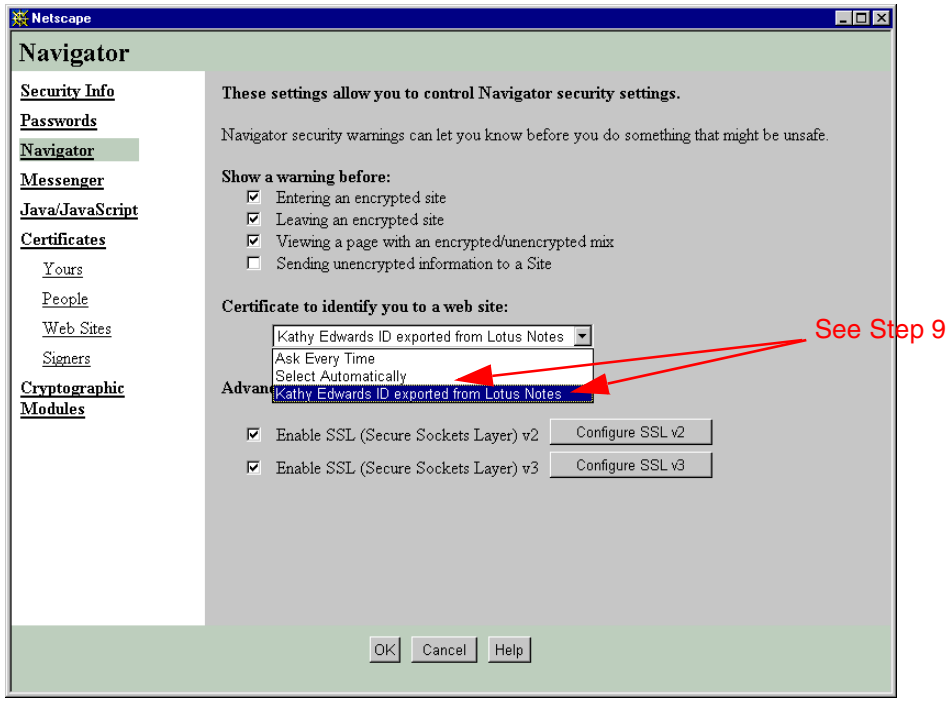

Figure 146. Netscape - private key

For Internet Explorer

Use the following instructions when importing certificates into Internet Explorer.

1. Internet Explorer will bring up a file download window with two options. Choose **Save this file to disk** and save the file.
2. Now you must install the certificate. To do so, select **Tools - Internet Options...** from the browser menu. An Internet Options window will appear.
3. Click the **Advanced** tab and scroll down to view the Security section (see Figure 147).

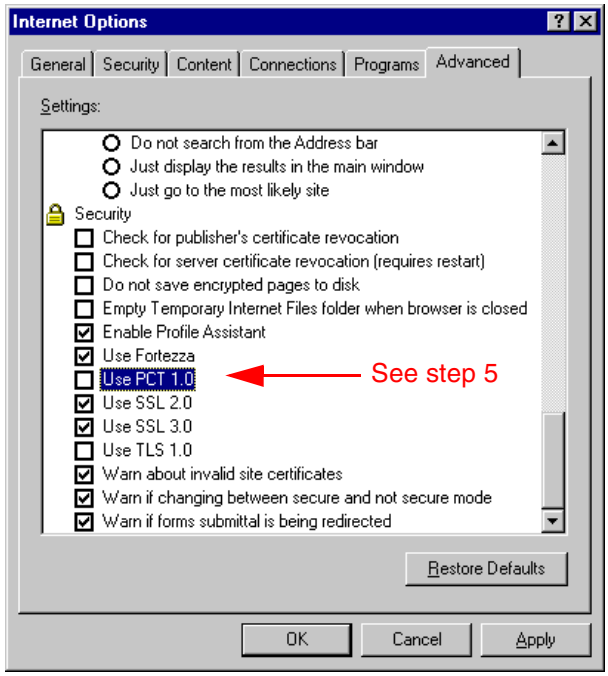

Figure 147. Internet Explorer - advanced Internet options

4. Select the **Use PCT 1.0** check box and click **Apply**. You may now deselected the box if desired.

> **Note**
>
> PCT (Private Communication Technology) is a Microsoft technology similar to SSL. The version of Internet Explorer used when writing this book had problems with client authentication if PCT 1.0 were left alone. It appeared as if some SSL flags were not enabled until PCT was turned on. Once the SSL flags are turned on, you may clear the PCT check box affecting SSL.

5. Next click the **Content** tab and click **Certificates.** A window will appear containing a list of certificates (see Figure 148).

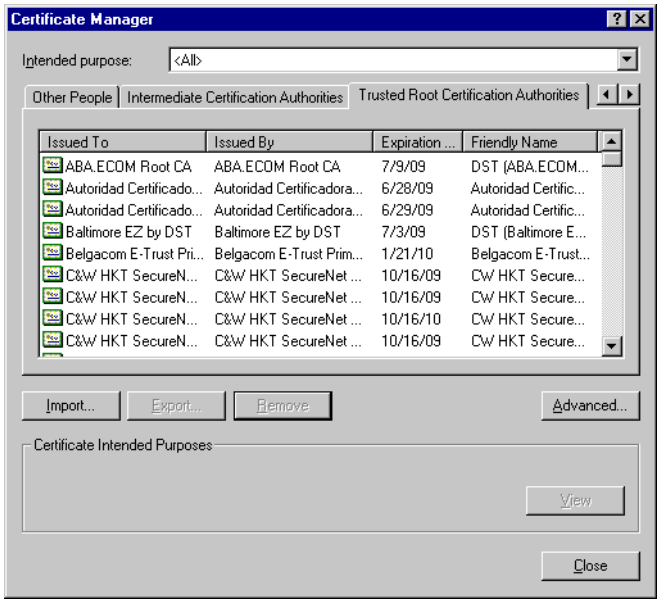

Figure 148. Internet Explorer - trusted root certificate authorities

6. Select the **Trusted Root Certification Authorities** tab. All the CAs that Internet Explorer trusts will be displayed. Now add another CA to that list.
7. Click the **Import** key. The Certificate Manager Import Wizard window will appear. Click **Browse**, then choose the directory and file downloaded in step 1 on page 216. The files should be of type **All Files (*.*)**.
8. Click **Next** in the Certificate Manager Import Wizard window to go to the second window. Select **Automatically select the certificate store based on the type of certificate** and click **Next.**
9. The window should now display all the setting for this certificate. Click **Finish**.
10. A window will appear to confirm the request to import the certificate. The window will contain the subject, issuer, validity period, and other relevant information. Click **Yes.** A window should indicate that the import was successful.
11. The new certificate should appear under the Trusted Root Certification Authorities tab in the Certificate Manager window.

Now you are ready to import the private certificate exported from Lotus Notes. The process will be similar to the way the CA's certificate was imported into Internet Explorer.

12. In the Certificate Manager**,** open the **Personal** tab and click **Import**.
13. In the first window of the Certificate Manager Import Wizard, browse for the certificate file exported in step 7 on page 211. Click **Next.**
14. The second window requires the user to enter the password for the certificate. There are also two check boxes:
    - **Enable strong private key protection**: If checked, a security window will appear after step 16 to configure the security settings.
    - **Mark the private key as exportable:** If unchecked, the private key can not be exported from Internet Explorer into a file.
15. In the third window, choose **Automatically select the certificate store based on the type of certificate.**
16. The last window contains the setting for the certificate. Click **Finish**.

    If the **Enable strong private key protection** box is checked, a window called Importing a New Private Exchange Key will appear. The window will give configuration instructions depending on the level of security desired. After configuring the window, click **OK.**
17. A window will appear to confirm the request to import the certificate. The window will contain the subject, issuer, validity period, and other relevant information. Click **Yes.** A new window will appear stating that the import was successful.
18. The new certificate will appear under the Personal tab in the Certificate Manager window. Select the **Certificate Manager** window and then click **OK** to close the Internet Options window.

Part 3. Appendixes

Appendix A. General security policies and procedures

Assessment and planning may include some of the following activities to define policy and procedures for an enterprise:

- Security workshops:

  The activities of a workshop include:

  - Understanding the business activities that are using critical internal and external connections
  - Understanding the key threats/perils related to these activities
  - Understanding the nature and priority of the organization's specific security requirements
  - Understanding and analyzing the security implications of the network topology
  - Identifying and analyzing the key security components in the network design
  - Identifying and analyzing the security characteristics of key
  - Discussing the security implications of future business plans and any impacts they might have on the current network topology and components

- Information asset profile

  The first stage is to identify the following

  - The organization's critical assets
  - Assets owners and custodians
  - Determine who depends on these assets and how they are used across the organization
  - Classify the security requirements for protecting these assets consistent with the business needs
  - Relate these security requirements to the organization's key business issues

  Interviews with key business and IT managers in the organization are conducted to understand what the critical business assets are and the nature and severity of any security risks and exposures to those assets. The risks examined are:

  - Impact to the organization if critical information gets into the wrong hands (confidentiality)
  - Impact to the organization if the wrong information is used (integrity)
  - Impact to the organization if critical information is not available for use when needed (availability)

- Security health checks

  This method identifies both the strengths and weaknesses in the organization's IT security controls. Typically, consultants conduct interviews with key managers and staff members in the organization to understand what security controls are in place in the following management areas:

  - Policy
  - Organization
  - Personnel
  - Physical controls
  - Asset classification and control
  - System access control
  - Network and computer management
  - Business continuity
  - Application development and maintenance
  - Compliance

- Security process assessment

  Compared to a security health check, this activity goes into sufficient depth to verify that each control selected has the right processes in place to implement the control. The review will be conducted against an agreed-upon, predefined information security standard or code of practice using interviews with appropriate staff within the organization. Security consultants will also verify the accuracy of the answers given by reviewing the actual processes and related documentation. This is done by inspecting examples of the process deliverables or outputs provided by the individuals who are responsible for executing the processes being reviewed.

- Application security assessment

  During this process, an in-depth, end-to-end review of the business application is conducted looking at the application's architecture, design and function, its development and maintenance processes, its operational processes and technology components, including the platform it runs on, the networking services used, and any database or operating platforms services used.

- Network security assessment

  This typically includes a technology and management analysis. The technology review component consists of intrusion tests and configuration analysis that give the involved personnel a thorough understanding of the strengths and weaknesses of the internal network components. The management review component consists of interviews with administrators

and management and reviews of documented security policies, standards, and processes.

- System security assessment

  This process typically assesses each component's (hardware and middleware) mechanisms for identification and authentication, access control, confidentiality, integrity, non-repudiation, audit, and alert in the context of the organization's documented policies, standards, and processes.

- Site security assessment

  Site assessments determine if procedures have been implemented to ensure a secure business environment for all employees and other persons working in the facilities. In addition, emergency plans covering anticipated emergencies and catastrophes have been established, and plans that adequately address the protection of people and assets should be available. Analysis of procedures that have been implemented to report and analyze security incidents, bring them to closure, and prevent reoccurrence are considered. There should be effective management processes to protect proprietary information and assets from unauthorized disclosure, modification or misappropriation. Finally, there is a process in place to provide management with a validation that the security controls within the scope of this engagement are operating effectively.

- Internet security assessment

  This is designed to help an organization to minimize the risk of a hacker causing damage to the network. This assessment provides a comprehensive review of the Internet solution on both a technical level and a management level. The technology review, consisting of multiple intrusion tests and configuration analysis, gives a thorough understanding of the strengths and weaknesses of an Internet solution. The management review consists of interviews with administrators and management and of reviews of security documentation. This provides an organization with insight into how the organization is prepared to handle the security of the solution over time.

- Ethical hacking

  Ethical hackers can simulate a real intruder's attacks but in a controlled, safe way. Consultants can tell what they find and what can be done to fix any problems and issues.

A.1 Security checklist

The following questions outline where to look for security exposures. This is not a comprehensive list and should not take the place of a professional security audit. They are presented for you to highlight the many different exposures there are to security in your environment and to help you understand how to combat them as you implement IBM Host Integration solutions.

A.1.1 Policy and guidelines

- Do you have a remote access security policy?
- Has your remote access security policy been updated recently?

  Equipment changes frequently. In some cases, procedures are specifically created for equipment that may no longer exist. New equipment may have been added.
- Does the remote access policy conform to all existing corporate communications guidelines?
- Does the remote access policy address the physical protection of the communications media, devices, computers and data storage at the remote site?
- Now that you are allowing remote users to access your system, are appropriate controls placed on access?
- Does the security policy require the classification of the functions, applications and data to determine the levels of security needed to protect the asset?
- Does a policy exist to obtain access to important proprietary information at remote sites?
- Does a policy exist that defines who is responsible in case of theft of hardware, software, or data at remote sites?
- Does a policy exist for reporting unauthorized activity?
- Does a policy exist for appropriate personal use of company equipment?
- Do remote access users have to sign a form stating they know and understand the remote access policies?
- Is there a formal, complete, and tested disaster recovery plan in place for the remote sites?

A.1.2 Identification and authorization

- Who is allowed to have access? Are there any restrictions for contractors, vendors or customers?
- Do the remote access security controls require that users be identified before the requested actions are initiated?
- Does each user have a unique identifier (user ID)?
- Does the corporate site maintain and use authentication data for verifying the identity of a user?
- Can the security controls uniquely identify each remote access user, device and port?
- Are there automatic time-out or lock-screen capabilities on the remote site equipment to control access during periods of non-use?

A.1.3 Access control

- Do the remote access security controls limit the unauthorized sharing of users?
- Does the access control mechanism support the customizing of privileges for each user ID at remote sites?
- Do the remote access security controls protect audit records from unauthorized access? Are users provided with last login session information?
- Are banners displayed regarding unauthorized usage?
- Are banners displayed regarding the usage of monitoring policy?
- Does the remote site have the capability to encrypt sensitive information, including authentication information?
- Are users allowed only one remote connection to the corporate network (per user ID or address)?

A.1.4 Auditing

- Does the remote access security mechanism record alarms and authentication violations as a default?
- Does the audit record for each recorded event identify:
  - Date and time of the event?
  - User or entity?

  - Origin of the event (for example, network address, originating phone number)?
  - Type of event?
  - Success or failure of the event?
- Is the audit trail information retained long enough to support reviews and analysis by security personnel and to meet corporate policy?
- If dial-up access to the remote site is possible, does the audit mechanism record the details associated with each user access?
- Can the security controls uniquely identify each remote access user, device and port?

A.1.5 Integrity

- Are there virus-scanning capabilities required on remote sites?
- How often are they updated?
- Are there capabilities to perform network and server congestion management in terms of monitoring, detection, and enforcement functions?
- Are measures in place to ensure the proper disposal of confidential data (paper, fax, digital, etc.) at remote sites?

A.1.6 Physical security

- Are the remote sites in physically secure locations?
- If equipment is stolen, can the perpetrator access proprietary information?
- Is a full physical inventory of remote site equipment and user systems maintained and periodically verified?
- Are backup tapes and media available and secured on-site for remote site equipment?
- Does a policy exist addressing fire, smoke, water and hazardous material contamination damage at a remote site?
- Is all paper data (proprietary, confidential, etc.) physically secure at the remote site?
- Is all computer data (floppies, hard drives, etc.) physically secure at the remote site?
- Is all media destruction (proprietary, confidential, etc.) at the remote site consistent with corporate security policies?

- Is there a process for return of equipment and proprietary data upon termination of employment or necessary company access?
- Does a policy exist for repair of equipment that contains proprietary information?
- Is there insurance for liability and personal injury at the remote site?

A.1.7 Security administration

- Are organizational responsibilities for remote access security defined?
- Is there a remote access security administrator?
- Is security a part of the defined responsibilities for the personnel who monitor, maintain and control various remote site equipment?
- Is there a process for authorizing new remote users, authorizing and updating remote user access capabilities, and deleting access when no longer needed?
- Are there periodic reviews of remote user privileges to ensure that capabilities remain commensurate with job functions?
- Do security event triggers generate alarms to provide administrator notification?
- Are security alarms properly categorized in terms of severity?
- Are the triggers modifiable by the administrator?
- Do the remote access security controls permit only authorized users (administrators) to grant access privileges to remote site equipment for new, authorized users?
- Do the remote access security controls allow network devices to be isolated when there is a compromise?
- Are there defined administrator responsibilities to isolate a compromised device?
- Do the remote access security controls include test, detecting and reporting communication errors (for example, high retransmission rate)?
- Is there a way to prevent bypass of the audit and alarm mechanisms by resetting remote access devices to invoke an insecure default configuration?
- Is periodic testing for unauthorized access, denial of service or other security weaknesses performed?
- Is there a defined practice of reviewing audit information on a periodic basis?

- Are there reporting capabilities to provide information on user profiles and access rules?
- Are there adequate controls to restrict access to and use of network troubleshooting equipment (for example, protocol analyzer)?
- Are there adequate controls to restrict access to and the use of network management software tools?
- Is there a capability to force reauthentication after the server has been unavailable?
- Is there a capability to force sign-off and prevent sign-on during system maintenance?
- Is there the means to run scheduled unattended backups of the remote site equipment?
- Are all security functions and software changes made only by an authorized administrator?
- Is there a way to ensure that only authorized legally acquired software (for example, applications, and tools) are installed and used on remote site equipment?
- Are backup copies of authorized software and documentation available?
- Are purchasing records and other proof of licensing requirements for software properly maintained?

A.1.8 Architecture and topology

- Is there network equipment in place that can separate traffic according to user communities?
- Is the remote access equipment interconnected with less trusted or untrusted (for example, Internet) networks?
- In a multiple remote site environment, are all sites maintained at the same security level?
- Are the remote access physical topology and network maps documented, verified and kept up to date?

Additional information on creating a security policy can be found in *Building Internet Firewalls*, by D. Brent Chapman & Elizabeth D. Zwicky.

Appendix B. Understanding the OIA

The OIA (Operator Information Area) is the area at the bottom of the window where session indicators and messages appear.

Listed below are the session information fields, with an explanation for each.

You can also find this list in the online help of IBM WebSphere Host On-Demand 5.0 that is included in the installation package.

Control Unit Status (Column 1)

M A connection to a Telnet server has been established

Connection Protocol (Column 2)

A The protocol is TCP/IP

System Available (Column 3)

***** the Session is connected to an application program (LU-LU connection)

p The session is connected to a host, but not to an application (SSCP-LU connection)

? The session is not connected or bind received

Security (Column 4)

When session data is being encrypted, a + appears in this column when using the standard emulator. If you are using Screen Customizer this area will be hidden. In addition to this indicator, there is another encryption indicator displayed on the emulator window, but it is not in the OIA area. It is in the lower right-hand corner of the window. If the session is encrypted a lock will be shown in the locked position. If the session is not encrypted, the lock will be shown in the unlocked position. This indicator is shown for IBM WebSphere Host On-Demand Versions 4 and 5.

Session Shortname (Column 7)

A single character (a-z) identifies the host session.

Input Inhibited (Column 9-17)

X [ ]	Time is required for the host system to perform a function (3270 session only). Please wait.
X SYSTEM	The host system has locked your keyboard. Please wait.
X <-o->	You tried to enter, insert, erase, or delete a character when the cursor was in a protected area. Move the cursor to an unprotected position and retry the operation (3270 session only).
X -f	You requested a function that is not supported in the current session.
X II	An operator input error occurred (5250 session).

Communications Messages (Columns 19-26)

These messages are preceded by a broken lightning bolt if the session is using the IBM3270 font; otherwise, they are preceded by COMM or PROG.

Communications Check

These messages indicate a communications problem between Host On-Demand and the server or host to which it is trying to connect.

- **COMM 654**

  The session could not establish a connection to the TN3270E server because the specified LU name is not valid. The LU name may not be valid for the following reasons:

  - The LU name is already in use by another session.
  - The LU name is not defined at the Telnet server.
  - The Telnet server does not support the LU type of the specified LU.
  - The LU name is not compatible with the requested LU type. For example, the session type is Display, but the specified LU is a Printer.
  - The Telnet server is unable to process this type of request. Contact your system administrator for help.
  - An unknown error occurred during Telnet device-type negotiations. Contact your system administrator for help.

  Ensure that your session's destination address, port, and LU name are correct. Also, ensure that your Telnet server is configured for the LU name

that you are requesting. To determine which error condition is occurring, take a Transport Level 1 trace of your session startup.

- **COMM 655**
  - The socket connection to the Telnet server has been established and the session is waiting for negotiation to finish.
  - The client has SSL off and has tried to connect to the server on an SSL port.
- **COMM 657**
  - The session is in the process of establishing the TCP/IP connection to the Telnet server.
  - For SSL:
    - The client has SSL on and has tried to connect to the server on a non-configured port. You will first receive a brief COMM 657, which changes to COMM 659. If Auto-reconnect is enabled, the emulator will cycle in this pattern; otherwise, COMM 659 remains.
    - The client has SSL on and has tried to connect to the server on a non-SSL port. You will first receive a COMM 657, which changes to COMM 659 after some time. If Auto-reconnect is enabled, the emulator cycles in this pattern; otherwise, it stays at COMM 659.

  When you close a session that displays COMM 657, there may be some delay before it closes. The delay varies. If you are in a hurry, close the browser.
- **COMM 658**

  The session is initializing the TCP/IP connection for TN3270E.
- **COMM 659**
  - The Telnet TCP connection to the session has not succeeded or has failed.
    - The TCP/IP connection to the Telnet3270 server could not be established.
    - You clicked Disconnect on the Communication menu.
    - The Telnet server closed the TCP/IP connection either by application control or because it detected an error.
    - For 5250, the specified workstation ID is already in use, and the host closed the connection.

    Ensure that your Telnet server and its port customization settings are correct. Also, ensure that your Telnet server is running and that it is

configured correctly. To determine which error condition is occurring, take a Transport Level 1 trace when the error occurs.

- For SSL:
  - The client has SSL off and has tried to connect to the server on a non-configured port.
  - The client has SSL on and has tried to connect to the server on a non-configured port. You will first receive a brief COMM 657, which changes to COMM 659. If Auto-reconnect is enabled, the emulator will cycle in this pattern; otherwise, COMM 659 remains.
  - The client has SSL on and has tried to connect to the server on a non-SSL port. You will first receive a COMM 657, which changes to COMM 659 after some time. If Auto-reconnect is enabled, the emulator cycles in this pattern; otherwise, it stays at COMM 659. The client has SSL on but cannot gain access to the key database on the server. This can happen if, for example, the database is not there, is corrupted, or does not have a password.

- **COMM 662**

  The server presented a certificate that was not trusted.

- **COMM 663**

  The server's certificate did not match its name. Because the session requested server authentication, the connection was refused.

- **COMM 664**

  A secure connection could not be completed.

- **COMM 665**

  The server's certificate is not yet valid.

- **COMM 666**

  The server's certificate has expired.

Program Check

These messages indicate that there is an error in the data stream sent from the host application.

- **Prog 750**

  A 3270 command was received that is not valid.

- **Prog 751**

  A START FIELD EXTENDED, MODIFY FIELD, or SET ATTRIBUTE order was received which specified a character set that is not valid.

- **Prog 752**

  A SET BUFFER ADDRESS, REPEAT TO ADDRESS, or ERASE UNPROTECTED TO ADDRESS order was received that specified an address that is not valid.

- **Prog 753**

  One or more of the following conditions occurred:

  - A READ MODIFIED, READ MODIFIED ALL, or READ BUFFER command that also contained data was received.
  - A REPEAT TO ADDRESS or GRAPHIC ESCAPE order was received that specified a character set that is not valid.
  - A START FIELD EXTENDED, MODIFY FIELD, or SET ATTRIBUTE order was received that specified an attribute value or character set that is not valid.

- **Prog 754**

  One of the following commands was received without the required parameters:

  - SET BUFFER ADDRESS
  - REPEAT TO ADDRESS
  - ERASE UNPROTECTED TO ADDRESS
  - START FIELD
  - START FIELD EXTENDED
  - MODIFY FIELD
  - SET ATTRIBUTE
  - GRAPHIC ESCAPE

- **Prog 755**

  A character code was received that is not valid.

- **Prog 756**

  A WRITE STRUCTURED FIELD command was received with a structured field that is not valid.

- **Prog 758**

  A SET REPLY MODE command was received with a mode that is not valid.

- **Prog 759**

  A WRITE STRUCTURED FIELD command was received with a structured field length that is not valid.

- **Prog 760**

  A WRITE STRUCTURED FIELD command was received with reserved fields that are not zero.

- **Prog 761**

  A WRITE STRUCTURED FIELD command was received with a partition identifier that is not valid.

- **Prog 780**

  An internal message was received with an incorrect direction.

- **Prog 797**

  SO was received; however, SO/SI are not paired correctly.

- **Prog 798**

  SO/SI or GRAPHIC ESCAPE was received in a DBCS field.

- **Prog 799**

  One or more of the following conditions occurred:

  - Address points to the second byte of a DBCS character.
  - A character attribute in a DBCS subfield is not valid.
  - STOP address is not valid.
  - General DBCS error.

Cursor's current line and column number (Columns 75-80)

Appendix C. Browser operations

Netscape and Internet Explorer operate differently when managing digital certificates. This appendix covers the most common functions for both browsers.

C.1 Netscape browser

Netscape Version 4.77 was used in documenting these procedures.

C.1.1 Connecting to a Web site with an unknown CA

When a user connects to a Web site signed by an unrecognized Certificate Authority, the Netscape browser displays many screens. The following process shows you how to allow Netscape to access the Web site with an unknown CA.

The first notification you will see is shown in Figure 149.

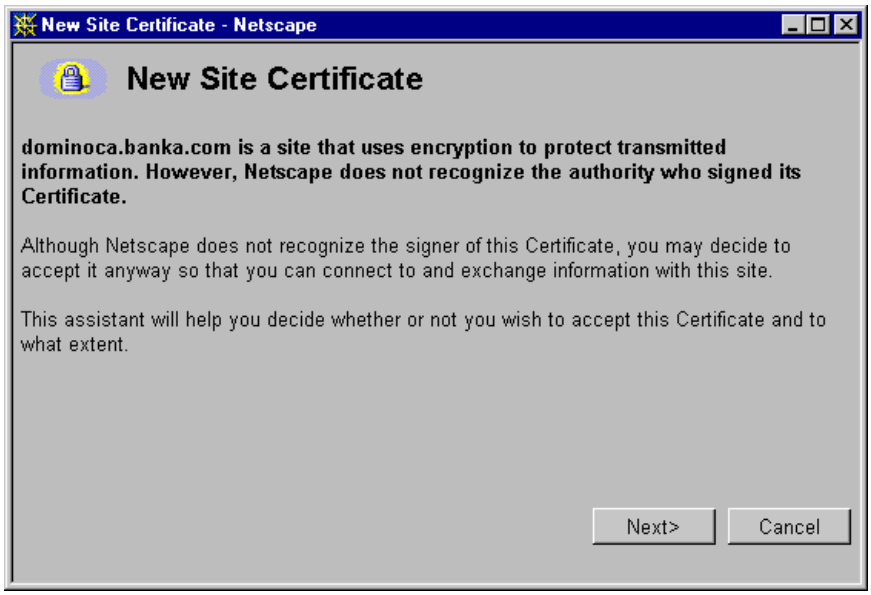

Figure 149. New Site Certificate in Netscape

Follow these instructions to accept the certificate.

1. Click **Next** to see the basic information (Figure 150) about the certificate that was presented.

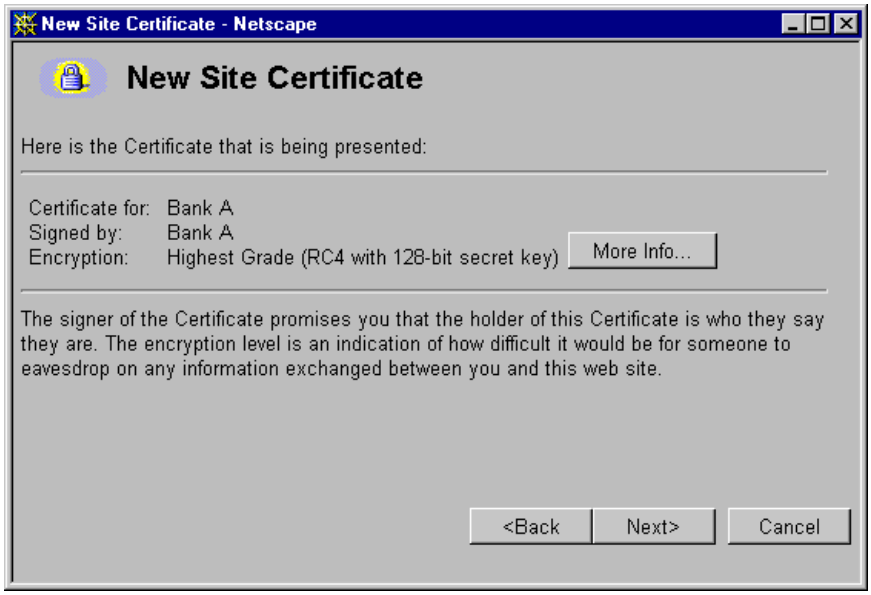

Figure 150. View New Site Certificate in Netscape

2. You may click **More Info...** to see the details of the certificate being processed. When ready to proceed click **Next** to see your options as shown in Figure 151.

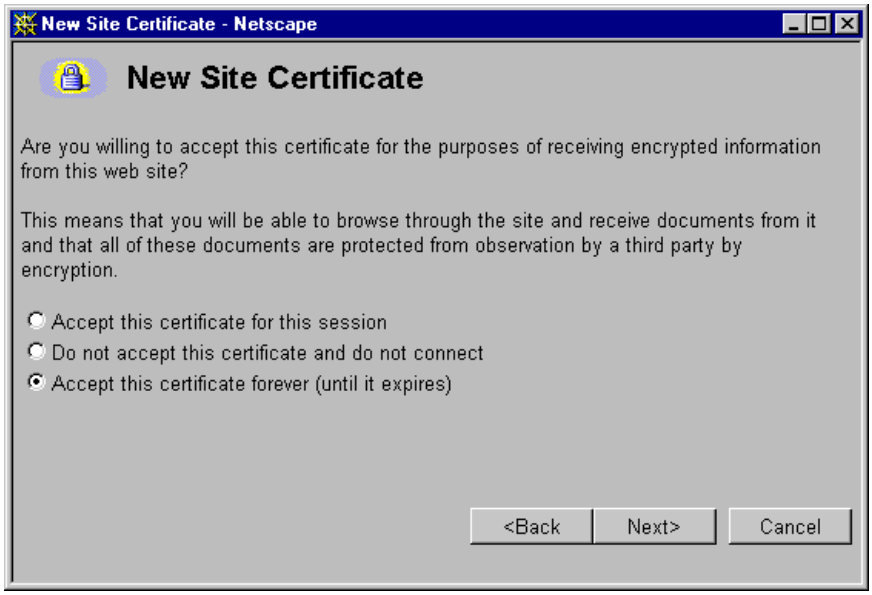

Figure 151. Select acceptance level of New Site Certificate in Netscape

3. There will be three choices. If you trust the certificate, select **Accept this certificate forever** then click **Next** to proceed to the next window and store the certificate.

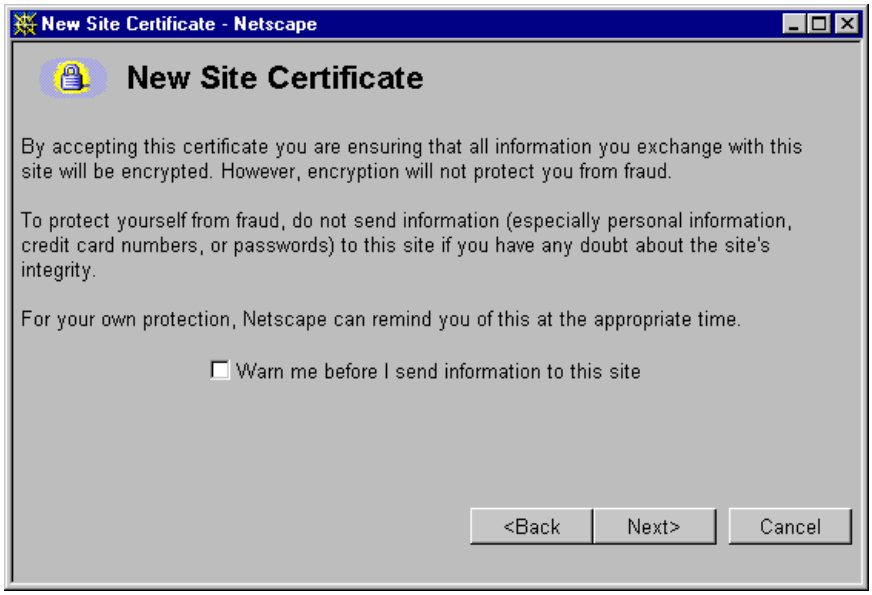

Figure 152. Warn before sending information for New Site Certificate in Netscape

4. Selecting the check box will warn the user each time information is sent to the Web site. Many users prefer the check box to remain unchecked. Click **Next** to move to the final window as shown in Figure 153, then click **Finish**.

Figure 153. New Site Certificate

C.1.2 Accept a CA as a trusted root

When you need to install a new CA into your browser, Netscape brings up many windows to inform and help the user. The process for the Netscape browser starts with the window shown in Figure 155.

1. To accept a Certificate Authority, you will navigate to the CA's Web page which will have a link similar to the one shown in Figure 154. Selecting that link will initiate the process of installing the CA's trusted root by displaying the window shown in Figure 155.

Figure 154. Certificate authority Web page

2. Click **Next** in this window and in the following window to continue.

Figure 155. New Certificate Authority

3. Click **Next** to view the window (Figure 156) that identifies the Certificate Authority.

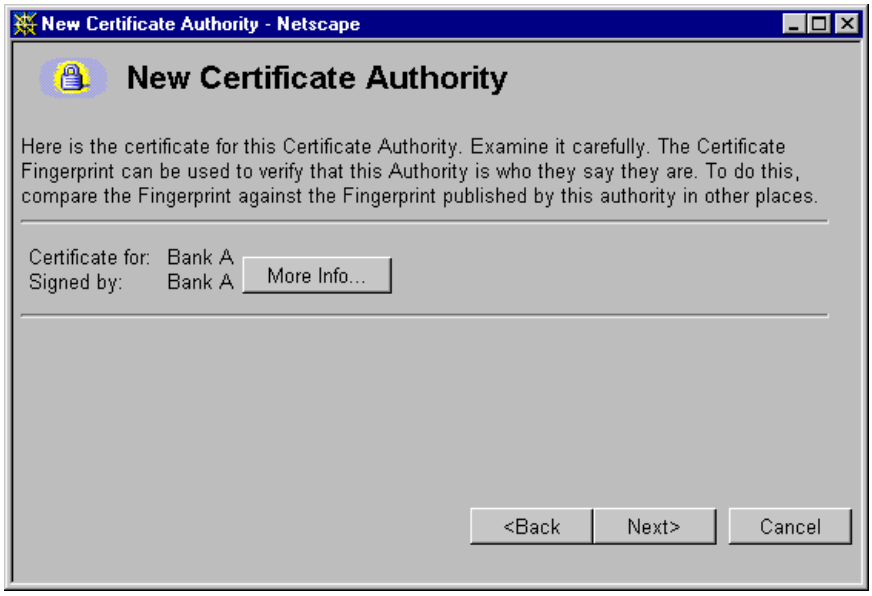

Figure 156. Identify a new Certificate Authority

4. You may click **More Info...** to view information about the CA. When you are ready to proceed, click **Next**.

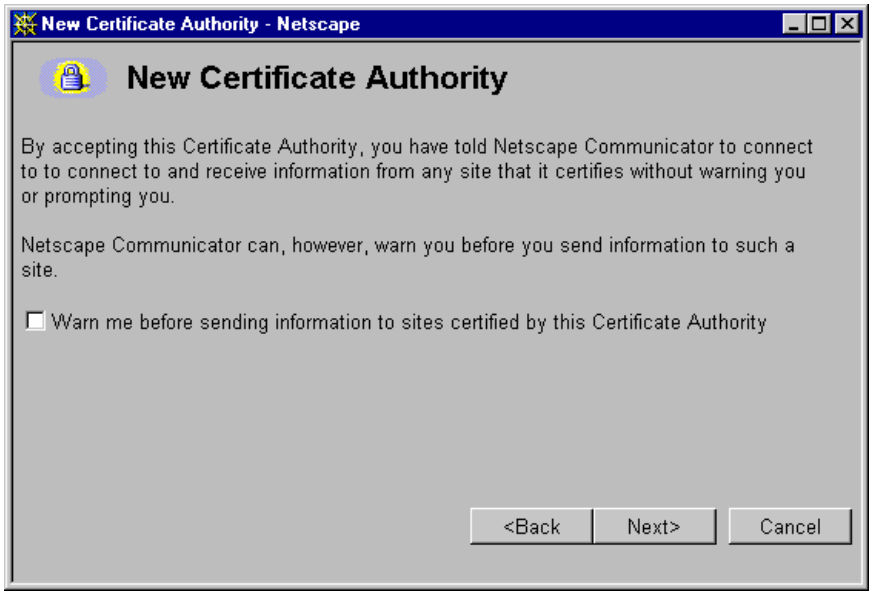

Figure 157. New Certificate Authority warning notice

5. You have the option to warn a user each time information is sent to the Web site by selecting the check box. Many users prefer the check box to remain unchecked. Click **Next** when you are ready to accept the certificate and install it.
6. The window shown in Figure 158 will be displayed allowing you to enter a name for this certificate. The name will help the user to identify the CA in his or her browser. Click **Finish** to store the certificate.

Figure 158. Name the new Certificate Authority

C.2 Internet Explorer

Microsoft's Internet Explorer Version 5.50.4522.1800 was used in documenting these procedures.

C.2.1 Connecting to a Web site with an unknown CA

When you encounter a Web site that presents a certificate, the browser checks to see if it recognizes the signer of the certificate by checking if the date is valid, and if the certificate contains the common name that matches the name of the server referenced in the URL (server authentication). If the common name matches and the certificate is valid, but the signer is unknown, you will be presented with the window shown in Figure 159.

Figure 159. Security alert

1. Clicking **Yes** at this time allows you to connect this time. It will not add the certificate to your list of trusted sites, and you will be shown this window every time you access this site using HTTPS. To add the certificate to the list of trusted sites, you must click **View Certificate**, which will display the window shown in Figure 160.
2. From the window shown in Figure 160, you can view additional details about the certificate, and install the certificate by clicking **Install Certificate.**

Figure 160. View Certificate

3. Installing the certificate launches the Certificate Import wizard introductory window, shown in Figure 161.

Figure 161. Certificate Import wizard

4. Click **Next** to start the wizard.

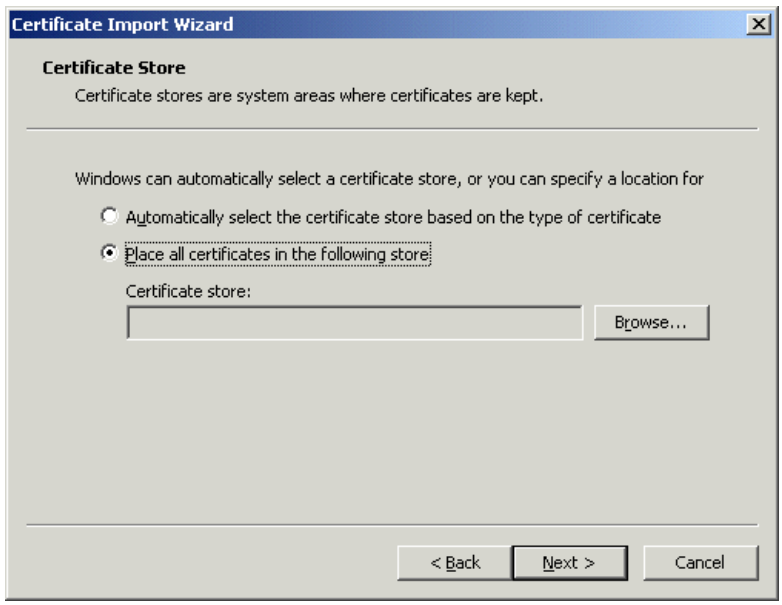

Figure 162. Certificate store

5. When Figure 162 appears, you must decide whether to let the wizard determine the appropriate location to store the certificate, or to let the user manually select the location. Generally the wizard will select the correct location. However, if the certificate is self-signed, the wizard will frequently select the wrong location. You may elect to use the manual method by selecting **Place all certificates in the following store**, then click **Browse** to view the list of certificate stores shown in Figure 163.

Figure 163. Available certificate store locations

6. For a server certificate select **Root Certificate Authorities** and click **OK**.

Figure 164. Certificate import complete

7. The window shown in Figure 164 will appear showing the store selected. Click **Finish** to store the certificate in the indicated location.
8. A final confirmation window (Figure 165) will appear displaying critical information about the certificate and asking for a final confirmation to store the certificate. Click **Yes** to store the certificate.

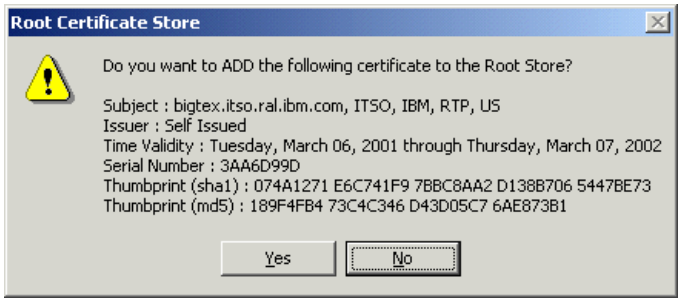

Figure 165. Store certificate confirmation

C.2.2 Accept a CA as a trusted root

When you need to install a new CA into your browser, the process is very similar to that already discussed in C.2.1, “Connecting to a Web site with an

unknown CA" on page 245. The process for Internet Explorer starts by navigating to the CA's Web page, as shown in Figure 154 on page 242.

1. To accept the Certificate Authority, select the highlighted link to initiate the process of installing the CA's trusted root. The first window displayed will be the instruction window shown in Figure 166.

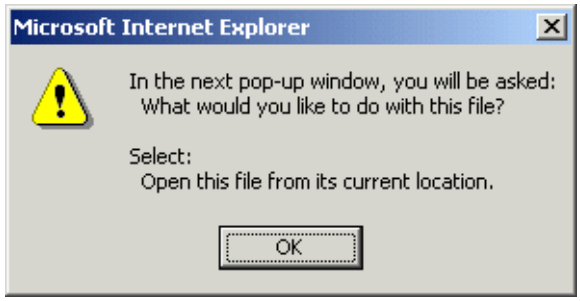

Figure 166. Install CA instructions for Internet Explorer

2. Clicking **OK** will result in the window shown in Figure 167.

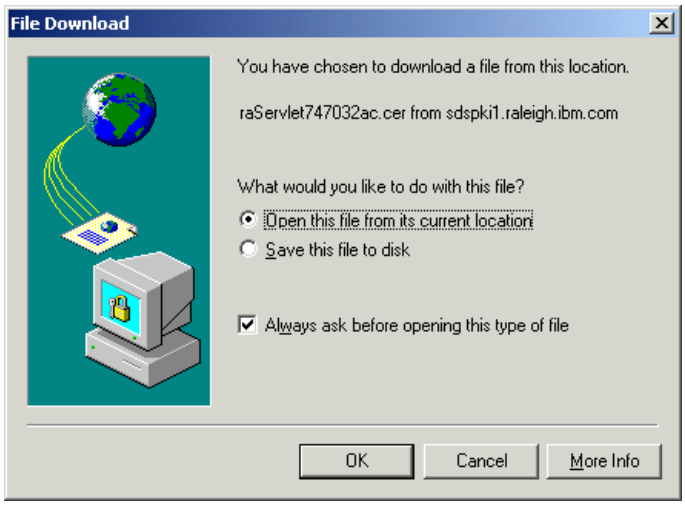

Figure 167. Execute install certificate program

3. As instructed, you must select **Open this file from its current location**, then click **OK** to launch the program to install the certificate.
4. From this point forward, follow the instructions in C.2.1, "Connecting to a Web site with an unknown CA" on page 245, beginning with step 3.

Appendix D. Sample Telnet-negotiated traces

The following are samples of IBM WebSphere Host On-Demand Level 3 traces taken of a successful and unsuccessful Telnet-negotiated session. The output has been reformatted to fit this document.

D.1 Successful negotiation

```
6@0@02/09/2001 16:20:05:083@null@err@ECL0037: Server 9.24.104.113:23 does not
support Telnet-negotiated security.
4@0@02/09/2001 16:21:43:975@Transport@A@---TN3270 : open() processing started.
4@1@02/09/2001 16:21:43:985@Transport@A@---TN3270 : DNS randomize host name =
9.24.104.113, TN3270 ::tel_init()
4@2@02/09/2001 16:21:43:995@Transport@A@---TN3270 : initialize() non-SSL socket
created.
4@3@02/09/2001 16:21:44:085@Transport@A@---TN3270 : Connected to 9.24.104.113, port
= 6623, TN3270 ::tel_init()
4@4@02/09/2001 16:21:44:145@Transport@A@---TN3270 : execute() Entry.
4@5@02/09/2001 16:21:44:155@Transport@A@-->TN3270 : Outbound Data Received:
length = 3, TN3270 ::read_instream()
4@6@02/09/2001 16:21:44:155@Transport@A@-->TN3270 : < . >
4@7@02/09/2001 16:21:44:155@Transport@A@   High =    FF2
4@8@02/09/2001 16:21:44:155@Transport@A@   Low  =    FDE
4@9@02/09/2001 16:21:44:155@Transport@A@Receive_data count is 3, TN3270
::receive_data()
4@10@02/09/2001 16:21:44:155@Transport@A@<--TN3270 : Response     CMD = WILL    OPT
= STARTTLS
4@11@02/09/2001 16:21:44:155@Transport@A@<--TN3270 : Inbound Data Sent:
length = 3, TN3270 ::sendData()
4@12@02/09/2001 16:21:44:155@Transport@A@<--TN3270 : < . >
4@13@02/09/2001 16:21:44:155@Transport@A@   High =    FF2
4@14@02/09/2001 16:21:44:155@Transport@A@   Low  =    FBE
4@15@02/09/2001 16:21:44:155@Transport@A@-->TN3270 : Negotiate    CMD = DO      OPT
= STARTTLS
4@16@02/09/2001 16:21:44:155@Transport@A@<--TN3270 : Response      IAC SB STARTTLS
FOLLOWS IAC SE
4@17@02/09/2001 16:21:44:155@Transport@A@<--TN3270 : Inbound Data Sent:
length = 6, TN3270 ::sendData()
4@18@02/09/2001 16:21:44:155@Transport@A@<--TN3270 : < .   0>
4@19@02/09/2001 16:21:44:155@Transport@A@   High =    FF20FF
4@20@02/09/2001 16:21:44:155@Transport@A@   Low  =    FAE1F0
4@21@02/09/2001 16:21:44:155@Transport@A@---TN3270 : Telnet.sendFollows() Do not
respond to any more telnet flows until session is secure
```

```
4@22@02/09/2001 16:21:44:155@Transport@A@---TN3270 : execute() Exit.
4@23@02/09/2001 16:21:44:396@Transport@A@---TN3270 : execute() Entry.
4@24@02/09/2001 16:21:44:396@Transport@A@-->TN3270 : Outbound Data Received:
length = 6, TN3270 ::read_instream()
4@25@02/09/2001 16:21:44:396@Transport@A@-->TN3270 : < .   0>
4@26@02/09/2001 16:21:44:396@Transport@A@   High =    FF20FF
4@27@02/09/2001 16:21:44:396@Transport@A@   Low  =    FAE1F0
4@28@02/09/2001 16:21:44:396@Transport@A@Receive_data count is 6, TN3270
::receive_data()
4@29@02/09/2001 16:21:44:396@Transport@A@<--TN3270 :
Process_SB_STARTTLS_FOLLOWS()TN3270 Start SSL to secure the Telnet Socket
4@30@02/09/2001 16:21:44:396@Transport@A@---TN3270 : secureSocket() start SSL on
existing NT connection
6@1@02/09/2001 16:21:49:984@null@err@ECLO008: Could not create a secure connection
to server "9.24.104.113:6623".
4@31@02/09/2001 16:21:50:004@Transport@A@---TN3270 [4]: Failed to securely connect
to host 9.24.104.113, port = 6623, TN3270 ::secureSocket()
4@32@02/09/2001 16:21:50:004@Transport@A@---TN3270 : execute() Exit.
4@33@02/09/2001 16:21:50:014@Transport@A@available() threw exception.Message-Socket
closed,exception-java.net.SocketException: Socket closed
4@34@02/09/2001 16:21:50:014@Transport@A@---TN3270 : execute() Entry.
4@35@02/09/2001 16:21:50:014@Transport@A@---TN3270 : Exception 2, TN3270
::needToRun()
4@36@02/09/2001 16:21:50:014@Transport@A@---TN3270 : execute()Exception not null
and count<0 .
4@37@02/09/2001 16:21:50:014@Transport@A@---TN3270 : execute() Call terminate().
4@38@02/09/2001 16:21:50:014@Transport@A@---TN3270 : syncTerminate() Entry.
4@39@02/09/2001 16:21:50:014@Transport@A@---TN3270 : Begin session termination.,
TN3270 ::tel_disc()
4@40@02/09/2001 16:21:50:084@Transport@A@---TN3270 : open() processing started.
4@41@02/09/2001 16:21:50:084@Transport@A@---TN3270 : DNS randomize host name =
9.24.104.113, TN3270 ::tel_init()
4@42@02/09/2001 16:21:50:094@Transport@A@---TN3270 : initialize() non-SSL socket
created.
4@43@02/09/2001 16:21:50:104@Transport@A@---TN3270 : Connected to 9.24.104.113,
port = 6623, TN3270 ::tel_init()
4@44@02/09/2001 16:21:50:104@Transport@A@---TN3270 : execute() Exit.
4@45@02/09/2001 16:21:50:164@Transport@A@---TN3270 : execute() Entry.
4@46@02/09/2001 16:21:50:164@Transport@A@-->TN3270 : Outbound Data Received:
length = 3, TN3270 ::read_instream()
4@47@02/09/2001 16:21:50:164@Transport@A@-->TN3270 : < . >
4@48@02/09/2001 16:21:50:164@Transport@A@   High =    FF2
4@49@02/09/2001 16:21:50:164@Transport@A@   Low  =    FDE
4@50@02/09/2001 16:21:50:164@Transport@A@Receive_data count is 3, TN3270
::receive_data()
```

```
4@51@02/09/2001 16:21:50:164@Transport@A@<--TN3270 : Response     CMD = WILL    OPT
= STARTTLS
4@52@02/09/2001 16:21:50:164@Transport@A@<--TN3270 : Inbound Data Sent:
length = 3, TN3270 ::sendData()
4@53@02/09/2001 16:21:50:164@Transport@A@<--TN3270 : < . >
4@54@02/09/2001 16:21:50:164@Transport@A@   High =    FF2
4@55@02/09/2001 16:21:50:164@Transport@A@   Low  =    FBE
4@56@02/09/2001 16:21:50:164@Transport@A@-->TN3270 : Negotiate     CMD = DO      OPT
= STARTTLS
4@57@02/09/2001 16:21:50:164@Transport@A@<--TN3270 : Response      IAC SB STARTTLS
FOLLOWS IAC SE
4@58@02/09/2001 16:21:50:164@Transport@A@<--TN3270 : Inbound Data Sent:
length = 6, TN3270 ::sendData()
4@59@02/09/2001 16:21:50:164@Transport@A@<--TN3270 : < .   0>
4@60@02/09/2001 16:21:50:164@Transport@A@   High =    FF20FF
4@61@02/09/2001 16:21:50:164@Transport@A@   Low  =    FAE1F0
4@62@02/09/2001 16:21:50:164@Transport@A@---TN3270 : Telnet.sendFollows() Do not
respond to any more telnet flows until session is secure
4@63@02/09/2001 16:21:50:164@Transport@A@---TN3270 : execute() Exit.
4@64@02/09/2001 16:21:50:464@Transport@A@---TN3270 : execute() Entry.
4@65@02/09/2001 16:21:50:464@Transport@A@-->TN3270 : Outbound Data Received:
length = 6, TN3270 ::read_instream()
4@66@02/09/2001 16:21:50:464@Transport@A@-->TN3270 : < .   0>
4@67@02/09/2001 16:21:50:464@Transport@A@   High =    FF20FF
4@68@02/09/2001 16:21:50:464@Transport@A@   Low  =    FAE1F0
4@69@02/09/2001 16:21:50:464@Transport@A@Receive_data count is 6, TN3270
::receive_data()
4@70@02/09/2001 16:21:50:464@Transport@A@<--TN3270 :
Process_SB_STARTTLS_FOLLOWS()TN3270 Start SSL to secure the Telnet Socket
4@71@02/09/2001 16:21:50:464@Transport@A@---TN3270 : secureSocket() start SSL on
existing NT connection
4@72@02/09/2001 16:21:50:945@Transport@A@---TN3270 : secureSocket() SSL socket
created.
6@2@02/09/2001 16:21:50:955@null@A@ECLOOO5: A SSL connection has been established
with host "9.24.104.113" using encryption suite SSL_RSA_WITH_RC4_128_SHA.
4@73@02/09/2001 16:21:50:965@Transport@A@---TN3270 : execute() Exit.
4@74@02/09/2001 16:21:51:005@Transport@A@---TN3270 : execute() Entry.
4@75@02/09/2001 16:21:51:005@Transport@A@-->TN3270 : Outbound Data Received:
length = 3, TN3270 ::read_instream()
4@76@02/09/2001 16:21:51:005@Transport@A@-->TN3270 : < . >
4@77@02/09/2001 16:21:51:005@Transport@A@   High =    FF2
4@78@02/09/2001 16:21:51:005@Transport@A@   Low  =    FD8
4@79@02/09/2001 16:21:51:005@Transport@A@Receive_data count is 3, TN3270
::receive_data()
```

```
4@80@02/09/2001 16:21:51:005@Transport@A@-->TN3270 : Negotiate     CMD = DO      OPT
= TN3270-E
4@81@02/09/2001 16:21:51:005@Transport@A@<--TN3270 : Response     CMD = WILL    OPT
= TN3270-E
4@82@02/09/2001 16:21:51:005@Transport@A@<--TN3270 : Inbound Data Sent:
length = 3, TN3270 ::sendData()
4@83@02/09/2001 16:21:51:005@Transport@A@<--TN3270 : < . >
4@84@02/09/2001 16:21:51:005@Transport@A@   High =    FF2
4@85@02/09/2001 16:21:51:005@Transport@A@   Low  =    FB8
4@86@02/09/2001 16:21:51:005@Transport@A@---TN3270 : execute() Exit.
4@87@02/09/2001 16:21:51:025@Transport@A@---TN3270 : execute() Entry.
4@88@02/09/2001 16:21:51:025@Transport@A@-->TN3270 : Outbound Data Received:
length = 7, TN3270 ::read_instream()
4@89@02/09/2001 16:21:51:025@Transport@A@-->TN3270 : < .    0>
4@90@02/09/2001 16:21:51:025@Transport@A@   High =    FF200FF
4@91@02/09/2001 16:21:51:025@Transport@A@   Low  =    FA882F0
4@92@02/09/2001 16:21:51:025@Transport@A@Receive_data count is 7, TN3270
::receive_data()
4@93@02/09/2001 16:21:51:025@Transport@A@<--TN3270 : Response      IAC SB TN3270E
DEVICE_TYPE REQUEST
4@94@02/09/2001 16:21:51:025@Transport@A@<--TN3270 : Inbound Data Sent:
length = 19, TN3270 ::sendData()
4@95@02/09/2001 16:21:51:025@Transport@A@<--TN3270 : < .  ...(        . 0>
4@96@02/09/2001 16:21:51:025@Transport@A@   High =    FF200444233332324FF
4@97@02/09/2001 16:21:51:025@Transport@A@   Low  =    FA82792DD3278D3D5F0
4@98@02/09/2001 16:21:51:025@Transport@A@---TN3270 : execute() Exit.
4@99@02/09/2001 16:21:51:045@Transport@A@---TN3270 : execute() Entry.
4@100@02/09/2001 16:21:51:045@Transport@A@-->TN3270 : Outbound Data Received:
length = 28, TN3270 ::read_instream()
4@101@02/09/2001 16:21:51:045@Transport@A@-->TN3270 : < .   ..(        . .   .+   0>
4@102@02/09/2001 16:21:51:045@Transport@A@   High =    FF200444233332324054335433FF
4@103@02/09/2001 16:21:51:045@Transport@A@   Low  =    FA82492DD3278D3D5121034E01F0
4@104@02/09/2001 16:21:51:045@Transport@A@Receive_data count is 28, TN3270
::receive_data()
4@105@02/09/2001 16:21:51:055@Transport@A@<--TN3270 : Response      IAC SB TN3270E
FUNCTIONS REQUEST
4@106@02/09/2001 16:21:51:055@Transport@A@<--TN3270 : Inbound Data Sent:
length = 10, TN3270 ::sendData()
4@107@02/09/2001 16:21:51:055@Transport@A@<--TN3270 : < .  .    0>
4@108@02/09/2001 16:21:51:055@Transport@A@   High =    FF200000FF
4@109@02/09/2001 16:21:51:055@Transport@A@   Low  =    FA837024F0
4@110@02/09/2001 16:21:51:055@Transport@A@---TN3270 : execute() Exit.
4@111@02/09/2001 16:21:51:085@Transport@A@---TN3270 : execute() Entry.
4@112@02/09/2001 16:21:51:085@Transport@A@-->TN3270 : Outbound Data Received:
length = 10, TN3270 ::read_instream()
```

```
4@113@02/09/2001 16:21:51:085@Transport@A@-->TN3270 : < .       0>
4@114@02/09/2001 16:21:51:085@Transport@A@   High =    FF200000FF
4@115@02/09/2001 16:21:51:085@Transport@A@   Low  =    FA834024F0
4@116@02/09/2001 16:21:51:085@Transport@A@Receive_data count is 10, TN3270
::receive_data()
4@117@02/09/2001 16:21:51:085@Transport@A@<--TN3270 : Response      IAC SB TN3270E
FUNCTION IS
4@118@02/09/2001 16:21:51:115@Transport@A@---TN3270E: execute() Exit.
4@119@02/09/2001 16:21:51:206@Transport@A@---TN3270E: execute() Entry.
4@120@02/09/2001 16:21:51:206@Transport@A@-->TN3270E: Outbound Data Received:
length = 42, TN3270E::read_instream()
4@121@02/09/2001 16:21:51:206@Transport@A@-->TN3270E: <        j. . dg8.  .     &
=   TELNET  .>
4@122@02/09/2001 16:21:51:206@Transport@A@   High =
00000300099380 88F8008000015007000ECDDCE0FE
4@123@02/09/2001 16:21:51:206@Transport@A@   Low  =
3000011331000047800200000800 0E0063535530FF
4@124@02/09/2001 16:21:51:206@Transport@A@Receive_data count is 42,
TN3270E::receive_data()
4@125@02/09/2001 16:21:51:206@Transport@A@---TN3270: Using Bind Screen size type
x7E.
4@126@02/09/2001 16:21:51:206@Transport@A@--->TN3270E: EOR command detected at
position      41
4@127@02/09/2001 16:21:51:216@Transport@A@-->TN3270E: EOR keyboard unlock
performed.
4@128@02/09/2001 16:21:51:216@Transport@A@---TN3270E: execute() Exit.
4@129@02/09/2001 16:21:51:226@Transport@A@---TN3270E: execute() Entry.
4@130@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: Outbound Data Received:
length = 661, TN3270E::read_instream()
4@131@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: <     B .   MSG10 OE/390 (03)
A&Raleigh - International Technical Support Organi>
4@132@02/09/2001 16:21:51:226@Transport@A@   High =
000000C17144DECFF4DC6FFF44FF51C5D898888464C9A8998A899894E888988894EA9999A4D98898
4@133@02/09/2001 16:21:51:226@Transport@A@   Low  =
0000052D0100427100651390 0D03D1109135978000953595139651303538593130247769306971 59
4@134@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: <zation - ITSO - ITSO1
B1System OE/390 (03) C2  .IEnter: .&                      >
4@135@02/09/2001 16:21:51:226@Transport@A@   High =
A8A899464CEED464CEEDF1CFEAAA894DC6FFF44FF51CF414CC9A897145141444444444444444444
4@136@02/09/2001 16:21:51:226@Transport@A@   Low  =
9139650009326000932611212823540651390 0D03D13201A955359A1A0D0300000000000000000000
4@137@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: <                         .R
<ZTSO03 userID - TSO on MVSO3 (9TSO28 userID - TSO >
4@138@02/09/2001 16:21:51:226@Transport@A@   High =
444444444444444444444441414D414EEEDFF4AA89CC464EED4994DEEFF14FEEDFF4AA89CC464EED4
```

```
4@139@02/09/2001 16:21:51:226@Transport@A@   Low  =
00000000000000000000000D01B901C93260304259940003260650452031D93262804259940003260
4@140@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: <on MVS28 |ITS039 userID -
TSO on MVS39 &RCICS         - TCP/IP CICS on MVS18 JZN>
4@141@02/09/2001 16:21:51:226@Transport@A@   High =
994DEEFF14CEEDFF4AA89CC464EED4994DEEFF15DCCCE44444444464ECD6CD4CCCE4994DEEFF1DED
4@142@02/09/2001 16:21:51:226@Transport@A@   Low  =
650452281F93263904259940003260650452391093932000000000003371970393206504521811 95
4@143@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: <VAS20       - NetView Access
on MVS20 K9IMS          - TCP/IP IMS on MVS18 MISYS>
4@144@02/09/2001 16:21:51:226@Transport@A@   High =
ECEFF444444464D8AE88A4C888AA4994DEEFF1DFCDE444444444464ECD6CD4CDE4994DEEFF1DCEEE
4@145@02/09/2001 16:21:51:226@Transport@A@   Low  =
5122000000000055359560133522065045220129942000000000000337197094206504521814928 2
4@146@02/09/2001 16:21:51:226@Transport@A@-->TN3270E: <6         - RALYDPD6 NR  0-
Your IP Address:     9.24.105.218    PLYour Telnet Po>
4@147@02/09/2001 16:21:51:226@Transport@A@   High =
F44444444464DCDECDCF1DD41D6E9A94CD4C8898AA744444F4FF4FFF4FFF4441DDE9A94E8998A4D9
4@148@02/09/2001 16:21:51:226@Transport@A@   Low  =
60000000000091384746159016086490970144952 2A000009B24B105B2180001738649035355307 6
4@149@02/09/2001 16:21:51:236@Transport@A@-->TN3270E: <rt:  01951             P0---
---------------------------------------- Q;-----L>
4@150@02/09/2001 16:21:51:236@Transport@A@   High =
9A744FFFFF4444444444441DF6666666666666666666666666666666666666666666661D566666D
4@151@02/09/2001 16:21:51:236@Transport@A@   Low  =
93A000195100000000000017000000000000000000000000000000000000000000018E000003
4@152@02/09/2001 16:21:51:236@Transport@A@-->TN3270E: <ast Command:          R LU:
RA03TN01         Sense Code:          R3Date: 02/0>
4@153@02/09/2001 16:21:51:236@Transport@A@   High =
8AA4C9998987400000000001D4DE74DCFFEDFF444444444E89A84C98874000000001DFC8A874FF6F
4@154@02/09/2001 16:21:51:236@Transport@A@   Low  =
12303644154A0000000000019034A0910335010000000002552503645A0000000001934135A00210
4@155@02/09/2001 16:21:51:236@Transport@A@-->TN3270E: <9/01 Time: 17:20:59 .>
4@156@02/09/2001 16:21:51:236@Transport@A@   High =    F6FF4E89874FF7FF7FFFE
4@157@02/09/2001 16:21:51:236@Transport@A@   Low  =    910103945A017A20A59FF
4@158@02/09/2001 16:21:51:236@Transport@A@Receive_data count is 661,
TN3270E::receive_data()
4@159@02/09/2001 16:21:51:246@Transport@A@--->TN3270E: EOR command detected at
position      660
4@160@02/09/2001 16:21:51:306@Transport@A@---TN3270E: execute() Exit.
4@161@02/09/2001 16:21:53:369@Transport@A@---TN3270E: execute() Entry.
4@162@02/09/2001 16:21:54:070@Transport@A@---TN3270E: execute() Exit-
InterruptedIOException.
4@163@02/09/2001 16:21:56:073@Transport@A@---TN3270E: execute() Entry.
```

```
4@164@02/09/2001 16:21:56:283@Transport@A@---TN3270E: execute() Exit-
InterruptedIOException.
4@165@02/09/2001 16:21:58:286@Transport@A@---TN3270E: execute() Entry.
```

D.2 Unsuccessful negotiation

```
4@297@08/21/2000 15:41:56:240@Transport@B@---TN3270 : open() processing started.
4@298@08/21/2000 15:41:56:240@Transport@B@---TN3270 : DNS randomize host name =
NcOd149, TN3270 ::tel_init()
4@299@08/21/2000 15:41:56:240@Transport@B@---TN3270 : initialize() non-SSL socket
created.
4@300@08/21/2000 15:41:56:460@Transport@B@---TN3270 : Connected to ncod149, port =
23, TN3270 ::tel_init()
4@301@08/21/2000 15:41:56:620@Transport@B@-->TN3270 : Outbound Data Received:
length = 3, TN3270 ::read_instream()
4@302@08/21/2000 15:41:56:620@Transport@B@-->TN3270 : < . >
4@303@08/21/2000 15:41:56:620@Transport@B@   High =    FF1
4@304@08/21/2000 15:41:56:620@Transport@B@   Low  =    FD8
4@305@08/21/2000 15:41:56:620@Transport@B@Receive_data count is 3, TN3270
::receive_data()
4@306@08/21/2000 15:41:56:620@Transport@B@<--TN3270 : Response      CMD = WILL
OPT = STARTTLS
4@307@08/21/2000 15:41:56:620@Transport@B@<--TN3270 : Inbound Data Sent:
length = 3, TN3270 ::sendData()
4@308@08/21/2000 15:41:56:620@Transport@B@<--TN3270 : < . >
4@309@08/21/2000 15:41:56:620@Transport@B@   High =    FF2
4@310@08/21/2000 15:41:56:620@Transport@B@   Low  =    FBE
4@311@08/21/2000 15:41:56:680@Transport@B@-->TN3270 : Negotiate     CMD = DO
OPT = TERMINAL TYPE
4@312@08/21/2000 15:41:56:680@Transport@B@<--TN3270 : Response      CMD = WILL
OPT = TERMINAL TYPE
4@313@08/21/2000 15:41:56:680@Transport@B@<--TN3270 : Inbound Data Sent:
length = 3, TN3270 ::sendData()
4@314@08/21/2000 15:41:56:680@Transport@B@<--TN3270 : < . >
4@315@08/21/2000 15:41:56:680@Transport@B@   High =    FF1
4@316@08/21/2000 15:41:56:680@Transport@B@   Low  =    FB8
4@317@08/21/2000 15:41:56:680@Transport@B@---TN3270 : execute() Exit.
4@318@08/21/2000 15:41:56:900@Transport@B@-->TN3270 : Outbound Data Received:
length = 6, TN3270 ::read_instream()
4@319@08/21/2000 15:41:56:900@Transport@B@-->TN3270 : < .   0>
4@320@08/21/2000 15:41:56:900@Transport@B@   High =    FF10FF
4@321@08/21/2000 15:41:56:900@Transport@B@   Low  =    FA81F0
4@322@08/21/2000 15:41:56:900@Transport@B@Receive_data count is 6, TN3270
::receive_data()
```

```
4@323@08/21/2000 15:41:56:900@Transport@B@<--TN3270 : Response      IAC SB
TERMINAL_TYPE IS IBM-3278-2-E IAC SE
4@324@08/21/2000 15:41:56:900@Transport@B@<--TN3270 : Inbound Data Sent:
length = 18, TN3270 ::sendData()
4@325@08/21/2000 15:41:56:900@Transport@B@<--TN3270 : < .  ..(        . 0>
4@326@08/21/2000 15:41:56:900@Transport@B@   High =    FF10444233332324FF
4@327@08/21/2000 15:41:56:900@Transport@B@   Low  =    FA8092DD3278D2D5F0
4@328@08/21/2000 15:41:56:900@Transport@B@---TN3270 : execute() Exit.
4@329@08/21/2000 15:41:57:010@Transport@B@-->TN3270 : Outbound Data Received:
length = 3, TN3270 ::read_instream()
4@330@08/21/2000 15:41:57:010@Transport@B@-->TN3270 : < . >
4@331@08/21/2000 15:41:57:010@Transport@B@   High =    FF2
4@332@08/21/2000 15:41:57:010@Transport@B@   Low  =    FD8
4@333@08/21/2000 15:41:57:010@Transport@B@Receive_data count is 3, TN3270
::receive_data()
4@334@08/21/2000 15:41:57:010@Transport@B@-->TN3270 : Negotiate     CMD = DO
OPT = TN3270-E
4@335@08/21/2000 15:41:57:010@Transport@B@<--TN3270 : Response      CMD = WILL
OPT = TN3270-E
4@336@08/21/2000 15:41:57:010@Transport@B@<--TN3270 : Inbound Data Sent:
length = 3, TN3270 ::sendData()
4@337@08/21/2000 15:41:57:010@Transport@B@<--TN3270 : < . >
4@338@08/21/2000 15:41:57:010@Transport@B@   High =    FF2
4@339@08/21/2000 15:41:57:010@Transport@B@   Low  =    FB8
4@340@08/21/2000 15:41:57:010@Transport@B@---TN3270 : execute() Exit.
4@341@08/21/2000 15:41:57:060@Transport@B@-->TN3270 : Outbound Data Received:
length = 7, TN3270 ::read_instream()
4@342@08/21/2000 15:41:57:060@Transport@B@-->TN3270 : < .    0>
4@343@08/21/2000 15:41:57:060@Transport@B@   High =    FF200FF
4@344@08/21/2000 15:41:57:060@Transport@B@   Low  =    FA882F0
4@345@08/21/2000 15:41:57:060@Transport@B@Receive_data count is 7, TN3270
::receive_data()
4@346@08/21/2000 15:41:57:060@Transport@B@<--TN3270 : Response      IAC SB TN3270E
DEVICE_TYPE REQUEST
4@347@08/21/2000 15:41:57:060@Transport@B@<--TN3270 : Inbound Data Sent:
length = 19, TN3270 ::sendData()
4@348@08/21/2000 15:41:57:060@Transport@B@<--TN3270 : < .  ...(        . 0>
4@349@08/21/2000 15:41:57:060@Transport@B@   High =    FF200444233332324FF
4@350@08/21/2000 15:41:57:060@Transport@B@   Low  =    FA82792DD3278D2D5F0
4@351@08/21/2000 15:41:57:060@Transport@B@---TN3270 : execute() Exit.
4@352@08/21/2000 15:41:57:060@Transport@B@-->TN3270 : Outbound Data Received:
length = 26, TN3270 ::read_instream()
4@353@08/21/2000 15:41:57:060@Transport@B@-->TN3270 : < .   ..(        . +|.    0>
4@354@08/21/2000 15:41:57:060@Transport@B@   High =    FF200444233332324044533 3FF
4@355@08/21/2000 15:41:57:060@Transport@B@   Low  =    FA82492DD3278D2D51EF6035F0
```

```
4@356@08/21/2000 15:41:57:060@Transport@B@Receive_data count is 26, TN3270
::receive_data()
4@357@08/21/2000 15:41:57:060@Transport@B@<--TN3270 : Response      IAC SB TN3270E
FUNCTIONS REQUEST
4@358@08/21/2000 15:41:57:060@Transport@B@<--TN3270 : Inbound Data Sent:
length = 10, TN3270 ::sendData()
4@359@08/21/2000 15:41:57:060@Transport@B@<--TN3270 : < .  .    0>
4@360@08/21/2000 15:41:57:060@Transport@B@   High =    FF200000FF
4@361@08/21/2000 15:41:57:060@Transport@B@   Low  =    FA837024F0
4@362@08/21/2000 15:41:57:120@Transport@B@---TN3270 : execute() Exit.
4@363@08/21/2000 15:41:57:170@Transport@B@-->TN3270 : Outbound Data Received:
length = 10, TN3270 ::read_instream()
4@364@08/21/2000 15:41:57:170@Transport@B@-->TN3270 : < .       0>
4@365@08/21/2000 15:41:57:170@Transport@B@   High =    FF200000FF
4@366@08/21/2000 15:41:57:170@Transport@B@   Low  =    FA834024F0
4@367@08/21/2000 15:41:57:170@Transport@B@Receive_data count is 10, TN3270
::receive_data()
4@368@08/21/2000 15:41:57:170@Transport@B@<--TN3270 : Response      IAC SB TN3270E
FUNCTION IS
6@4@08/21/2000 15:41:57:170@Host Access Class Library@null@err@ECL0037: Server
ncod149:23 does not support Telnet-negotiated security.
4@369@08/21/2000 15:41:57:170@Transport@B@---TN3270 : syncTerminate() Entry.
4@370@08/21/2000 15:41:57:170@Transport@B@---TN3270 : Begin session termination.,
TN3270 ::tel_disc()
4@371@08/21/2000 15:41:57:170@Transport@B@---TN3270 : execute() Exit.
```

Appendix E. Special notices

This publication is intended to help I/T managers and architects to plan and implement a secure host integration solution. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM WebSphere Host On-Demand V5.04 and IBM WebSphere Host Publisher V2.2. See the PUBLICATIONS section of the IBM Programming Announcement for IBM WebSphere Host On-Demand V5.04 and IBM WebSphere Host Publisher V2.2 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee

that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

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

e (logo)® 	Redbooks
IBM ®	Redbooks Logo 
AIX	RS/6000
Application System/400	SecureWay
AS/400	System/390
CICS	VTAM
Netfinity	WebSphere
OS/2	World Registry
OS/390	Lotus
OS/400	Lotus Notes
RACF	Domino
S/390	Notes

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

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

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

Appendix F. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

F.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 271.

- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149
- *Secureway Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements,* SG24-5631
- *Global Server Certificate Usage with OS/390 Webservers,* SG24-5623
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*, SG24-5659
- *A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions,* SG24-5201
- *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, SG24-5309
- *Secure e-business in TCP/IP Networks on OS/390 and z/OS*, SG24-5383
- *Building Integration Objects With IBM SecureWay Host Publisher Version 2.1*, SG24-5385
- *Enterprise-Wide Security Architecture and Solutions Presentation Guide*, SG24-4579
- *Java 2 Network Security,* SG24-2109
- *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855
- *Safe Surfing: How to Build a Secure WWW Connection*, SG24-4564
- *Remote Access to AS/400 with Windows 2000 VPN Clients,* a Redpaper found at
  `http://www.redbooks.ibm.com/redpapers/pdfs/redp0036.pdf`
- *Domino Certification Authority and SSL Certificates,* a Redpaper found at
  `http://www.redbooks.ibm.com/redpapers/pdfs/redp0046.pdf`
- *Deploying a Public Key Infrastructure*, SG24-5512

- *IBM Communications Server for AIX, V6 New Features and Implementation Scenarios*, SG24-5947
- *IBM Communications Server for OS/390 TCP/IP 2000 Update Technical Presentation Guide*, SG24-6162

F.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

F.3 Other resources

These publications are also relevant as further information sources:

- *Hacking Exposed Network Security Secrets & Solutions*, by Stuart McClure and Joel Scambray; 1999; Osborne/McGraw-Hill; ISBN 0-07-212127-0
- *Practical Firewalls,* by Terry Ogletree; 2000; Que/MacMillan; ISBN 0-7897-2416-2
- *Building Internet Firewalls,* by D. Brent Chapman & Elizabeth D. Zwicky; O'Reilly; 2000; ISBN 1-56592-871-7
- *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design,* by Electronic Frontier Foundation, John Gilmore (Editor), 1988
- *OS/390 System SSL Programming Guide and Reference,* SC24-5877
- *V2R10: OS/390 IBM Communications Server IP Migration*, SC31-8512
- The following information APARs:
  - II12362 *V2R10: IP Configuration Guide*, SC31-8725-00

- II12363 *V2R10: IP Configuration Reference*, SC31-8726-00
- II12364 *V2R10: IP Quick Reference,* SX75-0121-0)
- II12365 *V2R10: IP User's Guide,* GC31-8514-04
- II12366 *V2R10: IP Diagnosis Guide,* SC31-8521-04
- II12369 *V2R10: IP Messages Volume 3,* SC31-8674-05
- II12370 *V2R10: IP and SNA Codes*, SC31-8571-04
- *OS/390 IBM Communications Server Express Logon User's Guide,* found at ftp://ftp.software.ibm.com/software/network/library/whitepapers/elf.pdf
- *z/OS V1R1.0 CS: IP Configuration Guide*, SC31-8775
- *z/OS V1R1.0 CS: IP Configuration Reference*, SC31-8776
- *z/OS V1R1.0 CS: IP Quick Reference,* SX75-0124
- *z/OS V1R1.0 CS: IP User's Guide,* GC31-8780
- *z/OS V1R1.0 CS: IP Diagnosis Guide,* SC31-8782
- *z/OS V1R1.0 CS: IP Messages Volume 3,* SC31-8785
- *z/OS V1R1.0 CS:IP and SNA Codes*, SC31-8791
- *z/OS V1R1.0 CS: IP Migration*, SC31-8773

F.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- http://www.ibm.com/Security/ IBM's home for security information
- http://www.w3.com World Wide Web Consortium
- http://www.verisign.com/ VeriSign home page
- http://www.suitable.com/Doc_CodeSigning.shtml/ Suitable systems, signed applet security
- http://developer.netscape.com/docs/manuals/signedobj/trust/index.htm/ Netscape's developer's site discussing object signing
- http://www.rsasecurity.com/rsalabs/faq/ RSA's FAQ web site
- http://www.iana.org/numbers.htm Internet Assigned Numbers Authority, Protocol Numbers and Assignment Services
- http://www.rsa.com/rsalabs/newfaq/ Code Signing for Java

- ftp://ftp.software.ibm.com/software/network/library/whitepapers/elf.pdf IBM Express Logon User's Guide
- http://java.sun.com/products/jsse/ Sun's Java Secure Socket Extension
- http://www.cs.ucsd.edu/users/bsy/dobbertin.ps Cryptanalysis of MD5 Compression
- http://www.ietf.org/rfc.html IETF Request for Comments
- http://home.netscape.com/eng/ssl3/draft302.txt SSL V3.0 draft RFC

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Abbreviations and acronyms

3DES	Triple Digital Encryption Standard
ACL	Access Control List
AES	Advanced Encryption Standard
AH	Authentication Header
AS/400	Application System/400
CA	Certificate Authority
DCAS	Digital Certificate Access Facility
DMZ	demilitarized zone
ELF	Express Logon Facility
HMAC	Hashed Message Authentication Code
IBM	International Business Machines Corporation
IKE	Internet Key Exchange
IP	Internet Protocol
IPSec	IP Security Architecture
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITSO	International Technical Support Organization
JDK	Java Development Kit
JVM	Java virtual machine
LDAP	Lightweight Directory Access Protocol
L2TP	Layer 2 Tunneling Protocol
LUC	license use count
LUM	license use management
MAC	Message Authenticating code
MD5	RSA Message Digest 5 Algorithm
NAT	Network Address Translation
NIST	National Institute for Standards and Technology
NSA	National Security Agency
OEM	Original Equipment Manufacturer
OS/390	Operating System for the System/390 platform
OS/400	Operating System for the AS/400 platform
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
QOS	Quality Of Service
RACF	Resource Access Control facility
RADIUS	Remote Authentication Dial-In User Service
SET	Secure Electronic Transactions
SHA	Secure Hash Algorithm

S/MIME	Secure Multimedia Internet Mail Extensions
SOHO	Small Office/Home Office
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator
USS	Unformatted System Services
VPN	virtual private network
WWW	World Wide Web

Index

E

F

G

H

I

J

K

L

M

T

U

V

W

X

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5988-00 IBM Host Integration in a Secure Network: A Practical Approach
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

IBM Host Integration in a Secure Network: A Practical Approach

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM Host Integration in a Secure Network: A Practical Approach

IBM WebSphere Host On-Demand and Host Publisher

Integration into a Notes Domino environment

Other common secure scenarios

By their very nature, e-business applications are more vulnerable to attack than any type of previous I/T system. Security needs to be at the heart of these systems because the costs of getting it wrong can be enormous. The publicity associated with a breach of security reduces customer, supplier and Business Partner confidence in the enterprise and the share price plummets. There also may be actual monetary theft, or confidential information might be compromised. Whatever the details, the result is a loss to the business, and in the extreme case, the loss of the business itself.

Appropriate security needs to be in place for all systems - whether they are simply company Web pages with no access to the company intranet at all, or full-blown customer-to-business or business-to-business systems with external customers or partners having access to company systems. In this redbook, we show you some of the best security solutions that IBM Host Integration customers have available.

Network security is implemented to protect two objects: (1) the data that is transmitted on the network, and (2) the computers that are connected to the network. This redbook is directed at I/T managers and architects planning for, or implementing, an IBM Host Integration solution. It explores aspects of network security while implementing an IBM Web-to-host integration solution, consisting of IBM WebSphere Host Publisher and IBM WebSphere Host On-Demand. It describes the native security capabilities of each product and how they may be used by themselves and with other network security products to provide a secure environment.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5988-00 ISBN 0738419303