Securing applications with SSL
Securing applications with SSL
## Contents

### Part 1. Securing applications with SSL ................................. 1

- Chapter 1. Print this topic ............................................. 5
- Chapter 2. What’s new for V5R1 ........................................ 7
- Chapter 3. SSL usage of private keys on hardware ..................... 9
- Chapter 4. Supported SSL and Transport Layer Security (TLS) protocols .......... 11
- Chapter 5. Deciding where to obtain your digital certificates ............... 13
- Chapter 6. Setting up secure IBM HTTP Server (original) ............... 15
- Chapter 7. Setting up secure IBM HTTP Server (powered by Apache) .......... 19

- Chapter 8. Securing Client Access Express and Operations Navigator ............ 23
  - Setting up Client Access Express servers to use SSL .................. 23
  - Public Certificate Authorities in the Client Access Express key database ........ 24
  - Setting up SSL on Client Access Express clients .......................... 25
    - Installing the SSL component of Client Access Express .............. 25
    - Adding CA certificates from an iSeries ................................. 26
  - Enabling SSL on Operations Navigator ................................... 27
  - Enabling SSL for PC5250 sessions ........................................ 28
  - Enabling SSL for Data Transfer .......................................... 28
  - Enabling SSL for OBDC .................................................. 29

- Chapter 9. Securing Management Central .................................. 31
  - Configuring central system for server authentication .................... 33
  - Configuring endpoint systems for server authentication .................. 34
  - Configuring central system and endpoint systems for client authentication .......... 35

- Chapter 10. Securing Telnet ............................................. 37
  - Enabling client authentication for a PC5250 session .................... 38

- Chapter 11. Securing FTP ............................................... 41

- Chapter 12. Securing the Directory Services server ......................... 43

- Chapter 13. Securing DRDA/DDM ........................................ 45

- Chapter 14. Securing your Java applications with SSL ...................... 47
Part 1. Securing applications with SSL

Secure Sockets Layer (SSL) has become an industry standard for enabling applications for secure communication sessions over an unprotected network (such as the Internet). With the SSL protocol, you can establish secure connections between clients and server applications which provide authentication of one or both end points of the communication session. SSL also provides privacy and integrity of the data that client and server applications exchange. Netscape Communications Corporation originally developed SSL for securing web browser and server communications. However, the SSL specification was designed in such a way that other applications, such as TELNET and FTP, could also be enabled to use SSL.

As of release V5R1M0, the following iSeries applications are enabled for using SSL support:
- IBM HTTP Server for iSeries (original)
- IBM HTTP Server for iSeries (powered by Apache)
- FTP server
- Telnet server
- Distributed relational database architecture (DRDA) and distributed data management (DDM) server
- Management Central
- Directory Services Server (LDAP)
- Client Access Express applications, including Operations Navigator
- Applications that are written to the Client Access Express set of application programming interfaces (APIs)
- Programs developed with Developer Kit for Java and client applications that use IBM Toolkit for Java.
- Programs developed with Secure Sockets Layer (SSL) Application Programmable Interfaces (APIs) or Global Secure Toolkit (GSKit) APIs which can be used to SSL enable applications. See the Secure Sockets Layer (SSL) APIs for information on both SSL and GSKit APIs.

SSL is actually two protocols. The protocols are the record protocol and the handshake protocol. The record protocol controls the flow of the data between the two end points of an SSL session.

The handshake protocol authenticates one or both end points of the SSL session and establishes a unique symmetric key used to generate keys to encrypt and decrypt data for that SSL session. SSL uses asymmetric cryptography, digital certificates, and SSL handshake flows to authenticate one or both end points of the SSL session. Usually, the server is authenticated and optionally the client is authenticated. A digital certificate, issued by a Certificate Authority, can be assigned to each of the end points or to the applications using SSL on each end point of the connection.

With server authentication, the client will ensure that the server certificate is valid and that it is signed by a Certificate Authority which the client trusts. SSL will use asymmetric cryptography and handshake protocol flows to generate a symmetric key which will be used only for this unique SSL session. This key is used to generate a set of keys which are used for encrypting and decrypting data which will flow over the SSL session. Subsequently, when a SSL handshake has completed, one or both ends of the communication link will have been authenticated and a unique key will have been generated to encrypt and decrypt the data. Once the handshake is completed then application layer data will flow encrypted across that SSL session.

A digital certificate, issued by a Certificate Authority, can be assigned to each of the end points or to the applications using SSL on each end point of the connection. The digital certificate is comprised of a public key and some identifying information that has been digitally signed by a trusted Certificate Authority (CA). Each public key has an associated private key. The private key is not stored with or as part of the certificate. In both server and client authentication, the end point which is being authenticated must prove that it has access to the private key associated with the public key contained within the digital certificate.

Many applications allow the option to enable client authentication. With client authentication, the server will ensure that the client certificate is valid and that it is signed by a Certificate Authority which the server trusts. The following iSeries applications support client authentication:
SSL handshakes are performance intensive operations because of the cryptographic operations using the public and private keys. After an initial SSL session has been established between two end points, the SSL session information for these two end points and applications can be cached in secure memory to speed up subsequent SSL session enablements. When an SSL session is resumed, the two end points use an abbreviated handshake flow to authenticate that each has access to unique information without using the public or private keys. If both can prove that they have access to this unique information, then new symmetric keys are established and the SSL session is "resumed". For TLS Version 1.0 and SSL Version 3.0 sessions, cached information will not remain in the secure memory for greater than 24 hours. In V5R1M0, SSL handshake performance impacts on the main CPU can be minimized by using the 4758 Cryptographic Coprocessor (model 023) to do the asymmetric cryptography processing. You can choose to securely store private keys on the hardware card or store the private keys in software protected by the master key of the card. See SSL usage of private keys on hardware for details.

Prerequisite programs

You must install the following programs to enable SSL:

- IBM Digital Certificate Manager (DCM), option 34 of OS/400 (5722-SS1)
- TCP/IP Connectivity Utilities for iSeries (5722-TC1)
- IBM HTTP Server for iSeries (5722-DG1)
- One of the IBM Cryptographic Access Provider products: 5722-AC2 (56-bit) or 5722-AC3 (128-bit). The bit size for these products indicates the maximum size of the secret material within the symmetric keys that can be used in cryptographic operations. The size allowed for a symmetric key is controlled by the export and import laws of each country. A higher bit size results in a more secure connection. Some of these products are not available in all countries due to government export and import regulations.
- You may also want to install and configure the 4758–023 Cryptographic Coprocessor to use with SSL to speed up the SSL handshake processing. If you want to install the 4758–023, you must also install Option 35, Cryptographic Service Provider. For other requirements for the 4758 Cryptographic Coprocessor, see the 4758 Coprocessor requirements.

If you want to use SSL with any Client Access Express or IBM Toolbox for Java component you must also install at least one of the iSeries Client Encryption products: 5722-CE2 (56-bit) or 5722-CE3 (128-bit). Client Access Express needs one of these products in order to establish the secure connection.

Note: You do not need to install a Client Encryption Product to use the PC5250 emulator that is shipped with the Personal Communications product. Personal Communications has its own built-in encryption code.

On iSeries, you use IBM Digital Certificate Manager (DCM) to manage digital certificates. For information about using DCM, see the Information Center topic Using Digital Certificate Manager. Use the following links to learn how to enable SSL on your applications:

- What's new for V5R1
  This topic provides descriptions and links to new functions and topics in Securing applications with SSL.
- SSL usage of private keys on hardware
  This topic discusses how SSL uses private keys for digital certificates that have been stored on the 4758 Cryptographic Coprocessor.
• **Supported SSL and Transport Layer Security (TLS) protocols**
  This topic discusses the different types of functional differences between the different Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

• **Deciding where to obtain your digital certificates**
  Use this topic to determine what type and from what Certificate Authority you will obtain digital certificates. This section provides recommendations for using external and local Certificate Authorities.

• **Setting up secure of IBM HTTP Server (original)**
  Use this topic to secure IBM HTTP Server (original).

• **Setting up secure of IBM HTTP Server (powered by Apache)**
  Use this topic to secure IBM HTTP Server (powered by Apache) by using a virtual host to contain all SSL-protected directories.

• **Securing Client Access Express and Operations Navigator**
  Use this topic to secure Client Access Express and Operations Navigator using SSL.

• **Securing Management Central**
  Use this topic to secure Management Central systems and endpoint systems.

• **Securing Telnet**
  Use this topic to secure a Telnet server using SSL.

• **Securing FTP**
  Use this topic to secure a FTP server using SSL.

• **Securing the Directory Services Server**
  Use this topic to secure Directory Services Server (LDAP) using SSL.

• **Securing DRDA/DDM**
  Use this topic to secure Distributed relational database architecture (DRDA) and distributed data management (DDM) server using SSL.

• **Securing your Java applications with SSL**
  Use this topic to secure Java applications. This topic provides links to sample programs that show how you can secure a Java application.
Chapter 1. Print this topic

You can view or download a PDF version of this document for viewing or printing. You must have Adobe® Acrobat® Reader installed to view PDF files. You can download a copy from the Adobe home page. To view or download the PDF version, select Securing applications with SSL (about 215 KB or 54 pages).

To save a PDF on your workstation for viewing or printing:
1. Open the PDF in your browser (click the link above).
2. In the menu of your browser, click File.
3. Click Save As...
4. Navigate to the directory in which you would like to save the PDF.
5. Click Save.
Chapter 2. What’s new for V5R1

For V5R1, iSeries provides a number of enhancements for using SSL. The following list describes some of the new functional enhancements that pertain to SSL-enabled applications:

New support of Transport Layer Security
Based on SSL Version 3.0, Transport Layer Security (TLS) protocol is the latest industry standard SSL protocol and its specifications are defined by the Internet Engineering Task Force (IETF) in RFC 2246, "The TLS Protocol."

See Supported SSL and Transport Layer Security (TLS) protocols for an overview of the differences between SSL Version 3.0 and TLS Version 1.0.

Any iSeries applications that is enabled for SSL will automatically obtain TLS support unless the application has specifically requested to use only SSL Version 3.0 or SSL Version 2.0.

New support for secure File Transfer Protocol (FTP)
Also for V5R1, iSeries FTP server now provides SSL support for securing file transfer. See Securing FTP to learn how to configure this support.

IBM 4758 Cryptographic Coprocessor provides SSL performance enhancements
In V5R1, you can use the IBM 4758 Cryptographic Coprocessor to improve SSL performance and provide more secure private key storage. For example, IBM Digital Certificate Manager (DCM) now allows you to use the coprocessor to store and encrypt private keys. See SSL usage of private keys on hardware to learn how SSL can be used with the coprocessor.

SSL support for securing your Java applications
In addition to the above functional enhancements, you can secure Java programs and other applications with SSL. See Securing your Java applications with SSL for details and links to example Java programs.

New Global Secure Kit (GSKit) application programming interfaces (APIs)
The OS/400 Global Secure Toolkit (GSKit) and OS/400 Secure Sockets Layer (SSL) application programming interfaces (APIs) enable and facilitate secure communications between processes on a network. Just like the SSL APIs, GSKit APIs allow you to access SSL and TLS functions from your socket application program. GSKit APIs provide more options and functionality than the SSL APIs and are the preferred method to secure applications. For a description of GSKit APIs, see Secure Sockets Layer (SSL) APIs in the API Reference information in the Information Center.

Advanced Encryption Standard (AES) cipher algorithm
We support the 128-bit AES cipher in V5R1. AES is a new encryption algorithm that replaces the DES algorithm. See the draft RFC entitled “AES Ciphersuites for TLS” on the IETF RFC draft web site. You can search for the title to access the draft version of this RFC.

New enhancements for using SSL with programming with sockets APIs
In addition to new GSKit APIs, other functional enhancements have been added to aid application developers to work with SSL and socket APIs. These enhancements include:

- **SSL-enabled Asynchronous Input/Output (I/O)**
  See Asynchronous I/O in the Socket programming for details.

- **Serviceability enhancements**
  See Serviceability enhancements in the Sockets programming information for details.

- **General SSL system messages**
  See Secure Sockets Layer (SSL) APIs in the API reference information for details.
Chapter 3. SSL usage of private keys on hardware

New for V5R1, you can use the IBM 4758-023 Cryptographic Coprocessor to more securely store cryptographic keys and to off-load the cryptographic operations that are completed during SSL-session establishment. The coprocessor is a hardware card that you can install on your iSeries system to diminish the CPU impacts that occur with asymmetric cryptographic operations used during the SSL handshake process.

You can use the 4758-023 Cryptographic Coprocessor with SSL in two different ways. You can use your 4758-023 Cryptographic Coprocessor with the OS/400 Digital Certificate Manager (DCM) to provide secure private key storage and to assist performance by off loading cryptographic operations that are completed during SSL-session establishment.

Note: The 4758-001 Cryptographic Coprocessor, which is a predecessor card, cannot be used with DCM, and therefore cannot be used to enhance SSL performance.

By using the 4758-023 Cryptographic Coprocessor to handle SSL private keys that are used during the SSL handshake operations, you can enhance performance and capability of SSL connections. To support load balancing and performance scaling, the iSeries allows you to use multiple (up to eight) 4758-023 Cryptographic Coprocessors with SSL. When using multiple coprocessors, the DCM configuration gives you the following options for using hardware to generate and store the private key that is associated with a digital certificate.

Private key generated in hardware and stored in hardware
Private keys are created and retained on the coprocessor. With this option, the private key never leaves the coprocessor, and thus the private key cannot be used or shared with another coprocessor. This means that if you want to use multiple cards for your applications SSL handshake processing, then you and your application have to manage multiple private keys and certificates.

Private key generated in hardware and stored in software
This option allows a single private key to be shared among multiple coprocessors. The private key is generated by one of the coprocessors and then is encrypted under the master key of that coprocessor. The master key is used to encrypt all keys that exist outside of the coprocessor. The private key is never present in clear form (unencrypted form) outside of the coprocessor. For multiple coprocessors to use the same private key, they must all have the same master key. If you entered the master key in parts for one coprocessor, you must enter the same master key parts for all of the other coprocessors. If a random master key was generated inside of the coprocessor, then you must use a process called master key cloning to get the same master key in the other coprocessors. You can use the 4758 Cryptographic Coprocessor Configuration utility, found on the AS/400 Tasks page, to enter the master key in parts or to clone the master key between coprocessors. Because you can encrypt and store keys on the coprocessor, you can prevent exposure of keys to unauthorized users.

For more information on using SSL with the 4758–023 Cryptographic Coprocessor and DCM, see Introduction to the 4758 Cryptographic Coprocessor and Digital certificates for SSL secure communications. You can also view the online help for DCM for information on using SSL with the Cryptographic Coprocessor.
Chapter 4. Supported SSL and Transport Layer Security (TLS) protocols

There are multiple versions of the SSL protocol defined. The latest version, Transport Layer Security (TLS) Version 1.0, provides an evolutionary upgrade from SSL Version 3.0. The OS/400 implementation supports the following versions of the SSL and TLS protocols:

- TLS Version 1.0
- TLS Version 1.0 with SSL Version 3.0 compatibility

**Note:** Specifying TLS Version 1.0 with SSL Version 3.0 compatibility means that TLS will be negotiated if possible and if that is not possible then SSL Version 3.0 will be negotiated. If SSL Version 3.0 cannot be negotiated then the SSL handshake will fail.

**Note:** We also support TLS Version 1.0 with SSL Version 3.0 and SSL Version 2.0 compatibility. This is specified with the protocol value of **ALL**, which means that TLS will be negotiated if possible and if that is not possible then SSL Version 3.0 will be negotiated. If SSL Version 3.0 cannot be negotiated, SSL Version 2.0 will be negotiated. If SSL Version 2.0 cannot be negotiated, then the SSL handshake will fail.

- SSL Version 3.0
- SSL Version 2.0
- SSL Version 3.0 with SSL Version 2.0 compatibility

**SSL Version 3.0 versus SSL Version 2.0**

SSL version 3.0 is an almost totally different protocol compared to SSL Version 2.0. Some of the major differences between the two protocols include:

1. SSL Version 3.0 handshake protocol flows are different than SSL Version 2.0 handshake flows.
2. SSL Version 3.0 uses the BSAFE 3.0 implementation from RSA Data Security, Inc. BSAFE 3.0 includes a number of timing attack fixes and the SHA-1 hashing algorithm. The SHA-1 hashing algorithm is considered to be more secure than the MD5 hashing algorithm. Having SHA-1 allows SSL Version 3.0 to support additional cipher suites which use SHA-1 instead of MD5.
3. SSL Version 3.0 protocol reduces man-in-the-middle (MITM) type of attacks from occurring during SSL handshake processing. In SSL Version 2.0, it was possible, though unlikely, that a MITM attack could accomplish cipher specification weakening. Weakening the cipher could possibly allow an unauthorized person to break the SSL session key since the secret material within the generate secret key would be considerable shorter.

**TLS Version 1.0 versus SSL Version 3.0**

Based on SSL Version 3.0, Transport Layer Security (TLS) Version 1.0 is the latest industry standard SSL protocol. Its specifications are defined by the Internet Engineering Task Force (IETF) in RFC 2246, "The TLS Protocol." The major goal of TLS is to make SSL more secure and to make the specification of the protocol more precise and complete. TLS provides these enhancements over SSL Version 3.0:

- a more secure MAC algorithm
- more granular alerts
- clearer definitions of "gray area" specifications

Any iSeries applications that is enabled for SSL will automatically obtain TLS support unless the application has specifically requested to use only SSL Version 3.0 or SSL Version 2.0.
The following list describes the security improvements that TLS provides:

1. **Key-Hashing for Message Authentication**
   TLS uses Key-Hashing for Message Authentication Code (HMAC), which ensures that a record cannot be altered while travelling over an open network such as the Internet. SSL Version 3.0 also provides keyed message authentication, but HMAC is considered more secure than the (Message Authentication Code) MAC function that SSL Version 3.0 uses.

2. **Enhanced Pseudorandom Function (PRF)**
   PRF is used for generating key data. In TLS, the PRF is defined with the HMAC. The PRF uses two hash algorithms in a way which guarantees its security. If either algorithm is exposed then the data will remain secure as long as the second algorithm is not exposed.

3. **Improved finished message verification**
   Both TLS Version 1.0 and SSL Version 3.0 provide a finished message to both end points that authenticates that the exchanged messages were not altered. However, TLS bases this finished message on the PRF and HMAC values, which again is more secure than SSL Version 3.0.

4. **Consistent certificate handling**
   Unlike SSL Version 3.0, TLS attempts specify the type of certificate which must be exchanged between TLS implementations.

5. **Specific alert messages**
   TLS provides more specific and additional alerts to indicate problems that either session end point detects. TLS also documents when certain alerts should be sent.
Chapter 5. Deciding where to obtain your digital certificates

You obtain your digital certificates from Certificate Authorities (CA). Before you enable SSL communications for any application, you must decide whether you use a public CA or use a private CA to issue certificates.

Digital certificates from a public CA

Public CAs include, among others, VeriSign, Equifax, and Thawte. These commonly used Certificate Authorities are examples of public CAs.

Note: Some public CAs support the Public Key Infrastructure for X.509 certificates standard which was defined by the Internet Engineering Task Force (IETF) in RFC 2560, Internet Public Key Infrastructure Online Certificate Status Protocol. PKIX provides new standards for obtaining certificates and provides better standards between different digital certificate implementations. iSeries supports the ability to create certificates using a PKIX-compliant Certificate Authority (CA) through IBM Digital Certificate Manager.

Advantages of using a public CA:

- Your company does not need to operate a Certificate Authority, which means less administrative overhead when you do not need to issue certificates to users.
- Your SSL configuration is simplified, because many public CA certificates come pre-installed in the key databases of client applications.

Disadvantages of using a public CA:

- You must ensure that the Certificate Authority’s requirements to verify identification meet your security needs.
- You must trust the Certificate Authority not to lose or compromise its keys.
- The cost of obtaining certificates.

Certificates from public CAs are best for:

- Deployment on the Internet, such as with an e-commerce or other secure public web site.
  When you allow general public access to an application, public CA certificates can help protect data flow and restrict access to web pages between clients and the server. In Client Access Express, most public CA certificates come pre-installed. Therefore, the client can authenticate your system’s certificate easily to initiate SSL communications. Additionally, you could choose to authenticate clients to protected areas of your web site if they have the proper public certificates.
- Users that are loosely affiliated with your organization, such as business partners or vendors.
  Users may already be using certificates from a well-known CA for other purposes. Therefore, choosing certificates from a well-known CA creates less of an administrative burden on both the affiliated organization and your system administrator.

Digital certificates from private CAs

You can use Digital Certificate Manager (DCM) (Option 34 of OS/400) to create and operate a locally defined and controlled Certificate Authority.

Advantages of using private CAs

- You can control which users or applications can be issued certificates. You can issue certificates that are based on your own certificate guidelines, rather than relying on a public CA’s policies.
- Clients specify whether or not to trust certificates that are issued by a particular iSeries private CA. The certificate’s content may also be used to determine trust.

Disadvantages of using private CAs
• Using a private CA requires more administrative overhead.
  When you use a private CA, each client application that uses SSL to communicate with a server must obtain and store local copy of the private CA’s public certificate. Depending on the number of clients in your network, this may result in a higher management cost than purchasing certificates from a public CA whose CA certificate is often built into the client’s certificate database.

**Digital certificates from private CAs are best for**
  • Deployment on a corporate intranet.
    Using locally issued certificates allows you to more tightly control who has certificates and what they can access with those certificates.
  • Users that you want to have access to resources under an user profile.
  • Testing secure applications and environments without spending money on certificates.

For more information about CAs, types of digital certificates, and how certificates are issued, see [Understanding digital certificates](#).
Chapter 6. Setting up secure IBM HTTP Server (original)

You can configure IBM HTTP Server to use SSL for secure communications. Using SSL with either version of IBM HTTP Server will allow you to provide confidential information over the Internet and reduce the risk of that information falling into the wrong hands. Before you can configure HTTP Server to use SSL, you must have installed the prerequisite programs and set up digital certificates on your system. These steps assume that you have already configured an HTTP server.

To enable IBM HTTP Server (original) to use SSL, follow these steps:

1. Complete the Security Configuration form for your HTTP Server
   a. Start your web browser and go to the IBM HTTP Server front page.
   b. Click Configuration and Administration from the left navigational frame.
   c. Select the HTTP Server for which you would like to enable SSL from the pull-down menu.
   d. Click Security configuration.
   e. Complete the Security Configuration form:
      1) Select Allow SSL connections. The HTTP server (original) will allow secure sockets layer connections. (Directive: SSLMode)
      2) Specify a port number on which this server will accept SSL communications. (Directive: SSLPort)

         Note: Port 443 is the default SSL port; however, if another server is using port 443 for secure communications, you will need to specify another port.

   3) Select one of the following options for SSL client authentication (Directive: SSLClientAuth):
      a) Select None if you do not want to require the server to request a certificate from the client for use in establishing an SSL session. This is the default setting.
      b) Select Optional if the server should request a certificate from the client, but that the client is not required to return one. If a certificate is returned, it is validated. If the client does not return a certificate or if the certificate is not valid, then the application is responsible for providing a method of authenticating the client. In the case where the certificate is provided, the received certificate is validated. The session will be established if the client does not have a certificate of if validation is successful or validation fails because the certificate is self-signed, expired, or does not have a trusted root, the secure session will start. For all other validation failure cases the secure session does not start.
      c) Select Required if you want the server to request a valid certificate from a client. If the client does not have a valid certificate then the SSL session is not established.

   4) Click Apply to update your server with the changes you made to the form.

      You receive confirmation when your server processes the form. Note the Application ID that this process generates. You will need this ID in IBM Digital Certificate Manager to associate a certificate with the correct of the HTTP server. The HTTP server automatically generates the application ID for the server in DCM for you.

         Note: The HTTP server screen will instruct you to stop and restart the server. DO NOT do this until you reach Step 2n of this procedure. If you do, the HTTP server will not start.

2. Associate a certificate with the HTTP server
   b. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See Using Digital Certificate Manager for information on setting up a certificate system.
   c. Click the Select a Certificate Store button.
   d. Select *SYSTEM. Click Continue.
   e. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
f. When the left navigational menu reloads, expand **Manage Applications**.

g. Click **Update Certificate Assignment**.

h. On the next screen, select **Server** application. Click **Continue**.

i. Select the application name for the HTTP server that you want to secure.

j. Click **Update certificate assignment** to assign a certificate to the HTTP server. This certificate will establish its identity to clients.

   **Note:** If you specify that the HTTP server will use a certificate that a private CA issued, then users of that server must obtain a local copy of that CA certificate before they can establish an SSL connection.

k. Select a certificate from the list to assign to the server.

l. Click **Assign new certificate**.

m. DCM reloads the **Update Certificate Assignment** page with a message confirming the changes.

3. **Restart the server**

The HTTP Server (original) now supports four new directives that allow the server administrator to control specific SSL attributes. These allow a server administrator to specify the SSL version, enable caching of SSL information, control how long SSL cache entries exist, and specify a list of one or more cipher suites to be used. The names of the new directives are **SSLVersion**, **SSLDisableCache**, **SSLV3Timeout**, and **CipherSuiteList**. The HTTP Server graphical user interface does not add these directives to the configuration file. To use these directives, they must be added directly to the configuration file of the HTTP Server (original). For instructions on accessing the HTTP server (original) configuration file, see the [Product Documentation Center](#).

The following list describes these new directives:

**CipherSuiteList <cipher-suite>**

Where `<cipher-suite>` is a hexadecimal string indicating the set of cipher suites to be used for SSL handshakes. This parameter is required. This directive allows the server administrator to specify one or more SSL cipher suites that will be used for this HTTP Server instance. You might want to specify ciphers of a particular strength or key length for use with this HTTP Server (original).

   **Note:** If the directive is not included in the server configuration, the server will default to a set of predefined cipher-suite values based on the 5722–ACx product which is installed.

**SSLVersion <protocol value>**

This directive will allow a specific version of SSL to be used for the SSL connection. The values for the protocol value that are supported for this directive include:

- **ALL**  
  Supports TLS Version 1.0 with SSL Version 3.0 and SSL Version 2.0 compatibility. The protocol value of **ALL** means that TLS will be negotiated if possible and if that is not possible then SSL Version 3.0 will be negotiated. If SSL Version 3.0 cannot be negotiated, SSL Version 2.0 will be negotiated. If SSL Version 2.0 cannot be negotiated, then the SSL handshake will fail.

- **SSLV2**  
  Only SSL Version 2.0 will be allowed. (This is not recommended.)

- **SSLV3**  
  Only SSL Version 3.0 will be allowed.

- **TLSV1**  
  Only TLS Version 1.0 will be allowed.

- **TLSV1_SSLV3**  
  TLS Version 1.0 or SSL Version 3.0 will be allowed. The protocol value of **TLSV1_SSLV3**
means that TLS will be negotiated if possible and if that is not possible then SSL Version 3.0 will be negotiated. If SSL Version 3.0 cannot be negotiated then the SSL handshake will fail.

**Note:** If a parameter is NOT specified or is invalid, the SSLVersion will default to **ALL** value.

**SSLDisableCache** <**TRUE** or **FALSE**>

This value can be either **TRUE** or **FALSE**. If the value specified is invalid, SSL Session ID caching is enabled. SSL session caching is enabled by default. You do not need to specify this directive if you want caching to occur.

- **SSLDisableCache TRUE** : SSL Session Caching is **disabled**.
- **SSLDisableCache FALSE** : SSL Session Caching is **enabled**.

**Note:** If "SSLDisableCache TRUE" appears in the HTTP Server Configuration file, the value of SSLV3Timeout is ignored.

This directive indicates whether SSL should allow SSL session information to be cached on this machine for use in future SSL handshake negotiations. If this value is **FALSE** then caching will be allowed and abbreviated SSL handshake processing will be attempted on subsequent SSL connections from the same client after an initial successful SSL handshake. Allowing caching can decrease the amount of time and reduce the CPU utilization it takes to negotiate a subsequent SSL session with the same set of peers. This is accomplished by using an abbreviated SSL handshake protocol flow to fully authenticate the end points.

**SSLV3Timeout** <**time in seconds**>

The <**time in seconds**> value specified is an integer from 1 to 86400 that specifies the units, in seconds, of the time-out value. If a value is not specified, or is out of range, the SSLV3Timeout will be set to the default value of 86400 seconds (= 24 hours).

**Note:** If "SSLDisableCache TRUE" appears in the HTTP Server Configuration file, the value of SSLV3Timeout is ignored.

This directive is used to specify a time-out value, in seconds, for SSL Version 3.0 session caching. When a TLS Version 1.0 or SSL Version 3.0 session is negotiated with a client, a cache entry is added to the SSL cache. As long as that cache entry is valid, that same client can reconnect to this server using abbreviated SSL handshakes. After the time specified on this directive has expired, the TLS Version 1.0 or SSL Version 3.0 session cache entry for a client is removed from the cache. A subsequent SSL connection request from that same client will cause a full SSL handshake to be executed.

**Note:** SSL Version 2.0 cache cannot be controlled with this directive. The maximum time an SSL Version 2.0 cache exists is 100 seconds.
Chapter 7. Setting up secure IBM HTTP Server (powered by Apache)

For V5R1, the IBM HTTP server implementation supports HTTP server (powered by Apache). Apache is a public domain Web server that is known for its dynamic features and low cost (it is free). You can configure SSL on an HTTP server (powered by Apache). It is recommended that you create a virtual host to contain all directories and files that you wish to protect with SSL. A virtual host contains directories and files that can be served with a different web address from the same HTTP server (powered by Apache). By using a virtual host to contain all SSL-protected directories, you can better manage your HTTP server and control access to sensitive resources. You can configure a virtual host two ways:

- Add the appropriate directives to the configuration file of the HTTP server (powered by Apache).
- Use the Create virtual host wizard in the HTTP Server interface. This wizard allows you to add a virtual host and its directories.

Note: The following tasks assume that you have already configured an HTTP server and have created a virtual host on that server. You can also enable SSL for the entire HTTP Server (powered by Apache).

Before you can configure HTTP Server (powered by Apache) to use SSL, you must have installed the prerequisite programs and set up digital certificates on your system.

In these tasks, the corresponding HTTP Server (powered by Apache) directive has been provided in parentheses following the description of the SSL—related setting. To access the directives in the HTTP Server documentation, go to the Product Documentation Center, expand HTTP Server (powered by Apache)–> Reference–> Directives. From the list of directives that loads in the right frame, select the SSL directive that you are want to learn more about.

To enable SSL on a virtual host on IBM HTTP Server (powered by Apache), follow these steps:

1. Enable SSL on virtual host
   a. Start your web browser and go to the IBM HTTP Server front page.
   b. Click Configuration and Administration from the left navigational frame.
   c. Expand Configurations.
   d. From the pull-down menu, select the HTTP Server (powered by Apache) on which you created a virtual host.
   e. Click Virtual Host www.myvirtualhost.com.
   f. Click SSL General Settings. This will allow you to configure the basic setting for SSL enablement:
   g. Complete the SSL General Settings form:
      1) Select Enable SSL. This enables SSL server support for the virtual host. (Directives: SSLEnable, SSLDisable)
      2) Enter an Application name. This identifies the virtual host as an SSL-enabled application. Use this name when you associate a certificate in Digital Certificate Manager to the virtual host. (Directive: SSLAppName)
      3) Select the SSL cache enabled if you want to configure options for the following SSL cache settings:
         - SSL cache time-out: Enter a the number of seconds that you want an SSL session to be active without requiring a full SSL handshake with the client. A higher value indicates an abbreviated handshake, which is less secure, but has less performance drain. A lower value provides a full handshake which more secure; however; performance may suffer if the value is lower, because full handshakes are slower than an abbreviated handshake. You can set time-out values for the following SSL sessions:
Version 2 sessions: You can specify between 1 and 100 seconds as valid values for this field. If the specified value is greater than 100 seconds, or less than 1, the default value of 100 seconds is used as the time-out value.

Version 3 and TLS Version 1 sessions: You can specify between 1 and 86400 seconds as valid values for this field. If the specified value is greater than 86400, or less than 1, the default value of 86400 seconds (or one day) is used as the time-out value.

(Directives: SSLCacheEnable, SSLCacheDisable)

4) Click OK.

2. Configure SSL Connection Settings (optional)
   b. Click SSL Connection Settings. The SSL Client Handshake form provides methods for controlling how SSL negotiates with clients during the handshake process.
   c. Complete the SSL Connection Settings form:
      1) SSL version to negotiate: Use this to specify the SSL version that is negotiated with the client during the SSL handshake. The specified version must be negotiated or access to the directory is denied. The drop-down menu allows you to select the appropriate SSL version. The default value is "all". (Directive: SSLVersion)
      2) Ciphers available during negotiation: This allows you to specify a cipher spec used for the SSL connection. The table allows you to add, remove, and organize the cipher entries. (Directive: SSLCipherSpec)

3. Enable client authentication for virtual host (optional)

   Note: These steps are optional. By enabling client authentication, any users that accesses the virtual host will need a valid user certificate to access any of the directories that are contained within the virtual host.
   a. Start your web browser and go to the IBM HTTP Server front page.
      Note: If you already have IBM HTTP Server interface open, go to Step 2e.
   b. Click Configuration and Administration from the left navigational frame.
   c. Expand Configurations.
   d. From the pull-down menu, select the HTTP Server (powered by Apache) on which you created a virtual host.
   e. Click Virtual Host www.myvirtualhost.com.
   f. Click SSL Client Authentication.
   g. You can select the following options for controlling how the virtual host handles digital certificates upon initial client request (Directive: SSLClientAuth):
      • Do not use: Certificates are not used for client authentication.
      • Request client certificate before making connection: The server does not require valid certificates from a requesting client, but the server will accept certificates. The certificate does not need to be associated with a user ID or a validation list.
      • Require valid certificate for connection: The server requires a valid certificate to access the specified resource. The certificate does not need to be associated with a user ID or a validation list.

4. Add the virtual host as an application in DCM
   b. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See Using Digital Certificate Manager for information on setting up a certificate system.
   c. Click the Select a Certificate Store button.
   d. Select *SYSTEM. Click Continue.
   e. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
f. When the left navigational menu reloads, expand Manage Applications.
g. Click Add Application. Click Continue.
h. On the next screen, select Server application. Click Continue.
i. Complete the Add Application form:
   
   **Note:** Not all form values are described here. Only those options that are needed have been provided. For details on the other values on this form, see the DCM help that is associated with the Add Application form.
   
   1) In the Application ID field, type the name of the virtual host application ID that you created in Step 1g.
   2) Select Yes to indicate that the application supports client authentication.
   3) Select Yes to indicate that the application requires client authentication.
      
      **Note:** This option is Yes only if you selected Require valid certificate for connection when you specified client authentication (Step 2g).
   4) Select Application description and provide a brief description of the application that you are adding.

j. Click Add.

5. Associate a certificate with the virtual host
   
   b. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See Using Digital Certificate Manager for information on setting up a certificate system.
   c. Click the Select a Certificate Store button.
   d. Select *SYSTEM. Click Continue.
   e. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   f. When the left navigational menu reloads, expand Manage Applications.
   g. Click Update certificate assignment to assign a certificate to the virtual host. This certificate will establish its identity to clients. Click Continue.
      
      **Note:** If you specify that the virtual host will use a certificate that a private CA issued, then users of that server must obtain a local copy that CA certificate before they can establish an SSL connection.
   h. On the next screen, select Server application. Click Continue.
   i. On the Update Certificate Assignment page, select the application description that you created in Step 3i.
   j. Click Update Certificate Assignment.
   k. Select a certificate that you want to associate with the virtual host.
   l. Click Assign New Certificate.
   m. DCM reloads the Update Certificate Assignment page with a message confirming the changes.

6. Restart the server.
Chapter 8. Securing Client Access Express and Operations Navigator

Using SSL with Client Access Express allows your PC clients to communicate securely with your iSeries systems. All Client Access Express functions except Multimedia (USF), Messaging Application Program Interface (MAPI), and Incoming Remote Command (IRC) can use SSL. Some examples of Client Access Express applications that can communicate over a Secure Sockets Layer (SSL) connection include:

- Operations Navigator
- PC5250 emulations
- Data Transfers
- ODBC

You can also use the Client Access Express application programming interfaces (APIs) to create applications for Client Access Express that can use SSL for secure communications.

To configure Client Access Express to use SSL, you must perform these tasks:

1. Make sure that you have all of the prerequisite programs installed.
2. Set up the Client Access Express servers on your system to use SSL.
3. Set up Client Access Express to use SSL on each of your client PCs.

Setting up Client Access Express servers to use SSL

Before you can SSL-enable Client Access Express and its associated applications, including Operations Navigator, Data Transfer, PC5250, and ODBC, you must have installed the prerequisite programs and set up digital certificates on your system. CA certificates from several public Internet Certificate Authorities are built into the default Client Access Express key database. Using a certificate from one of those public CAs will simplify setting up your Client Access Express clients to use SSL. For a complete list of servers that you must enable for each Client Access function, see Informational APAR I12227.

To set up Client Access Express servers to use SSL, follow these steps:

2. Follow these steps to associate a certificate with the Client Access Express application IDs:
   a. Click the Select a Certificate Store button.
   b. Select "SYSTEM. Click Continue.
   c. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   d. When the left navigational menu reloads, expand Manage Applications.
   e. Click Update certificate assignment.
   f. On the next screen, select Server application. Click Continue.
   g. Select the application name for the Client Access server that you want to secure. If you have not secured any servers yet, click the button next to the application name for the sign on server, OS/400 TCP Signon Server.
   h. Click Update Certificate Assignment to assign a certificate to the OS/400 TCP Signon Server to use to establish its identity to Client Access Express clients.

   Note: If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client's key database, you will need to add it in order to use SSL. Adding CA certificates from an iSeries on page 26 provides information on how to do this. Finish this procedure before beginning that one.
   i. Select a certificate from the list to assign to the server.
   j. Click Assign New Certificate.
   k. DCM reloads the Update Certificate Assignment page with a message confirming the changes.

© Copyright IBM Corp. 2000, 2001
I. To add more Client Access servers, click **Cancel** once to return to the **Update Certificate Assignment** page and select another application.

You need to enable one or more additional Client Access Express servers depending on what applications you plan to run. The table below shows the application names and their corresponding application IDs for these servers. Repeat steps 2f-2k for the appropriate servers.

**Table 1. Application IDs for Client Access Express servers**

<table>
<thead>
<tr>
<th>Server Name</th>
<th>Application ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS/400 TCP Central Server</td>
<td>QIBM_OS400_QZBS_SVR_CENTRAL</td>
</tr>
<tr>
<td>OS/400 TCP Database Server</td>
<td>QIBM_OS400_QZBS_SVR_DATABASE</td>
</tr>
<tr>
<td>OS/400 TCP Data Queue Server</td>
<td>QIBM_OS400_QZBS_SVR_DTAQ</td>
</tr>
<tr>
<td>OS/400 TCP File Server</td>
<td>QIBM_OS400_QZBS_SVR_FILE</td>
</tr>
<tr>
<td>OS/400 TCP Network Print Server</td>
<td>QIBM_OS400_QZBS_SVR_NETPRT</td>
</tr>
<tr>
<td>OS/400 TCP Remote Command Server</td>
<td>QIBM_OS400_QZBS_SVR_RMTCMD</td>
</tr>
<tr>
<td>OS/400 TCP Signon Server</td>
<td>QIBM_OS400_QZBS_SVR_SIGNON</td>
</tr>
</tbody>
</table>

Other functions that are related to Client Access Express and that can use SSL include IBM HTTP Server (original), IBM HTTP Server (powered by Apache), Management Central, Directory Services Server, Telnet, and DRDA/DDM.

### Public Certificate Authorities in the Client Access Express key database

CA certificates from the following well-known Internet Certificate Authorities are built into the Client Access Express key database and marked as trusted. Using a certificate from one of these public CAs simplifies setting up your Client Access Express clients to use SSL:
- Thawte Personal Premium CA
- Thawte Personal Freemail CA
- Thawte Personal Basic CA
- Thawte Premium Server CA
- Thawte Server CA
- RSA secure server CA (also obtained from VeriSign)
- VeriSign class 4 public primary CA
- VeriSign class 3 public primary CA
- VeriSign class 2 public primary CA
- VeriSign class 1 public primary CA

### Additional information for IBM Toolbox for Java users:

If you plan to use SSL with IBM Toolbox for Java, note that a key database file that supports server certificates from a set of trusted authorities is shipped with IBM Toolbox for Java. You can get a certificate from one of the following companies:
- VeriSign, Inc.
- Thawte Consulting
- RSA Data Security, Inc.

**Note:** Users of Operations Navigator require SSL enablement of the Java toolbox if they want to SSL-enable most functions of Operations Navigator.
Setting up SSL on Client Access Express clients

To enable SSL on Client Access Express clients, follow these steps:

1. **Install the optional SSL component** of Client Access Express.
2. **Add the CA certificate** to the Client Access Express key database.
   
   If your Client Access Express servers use certificates from certain well-known Certificate Authorities, skip this step.
3. **Enable SSL for Client Access Express functions.**

   To manage your system from a PC through a secure connection, you can **SSL-enable Operations Navigator**. By default, if SSL is enabled for Operations Navigator, it is also enabled for all other Client Access Express functions that can use SSL. You can also enable or disable SSL separately for the following functions:
   
   - PC5250 emulation
   - Data Transfer
   - ODBC

Installing the SSL component of Client Access Express

Installing the optional SSL component of Client Access Express is the first step in **Setting up SSL on Client Access Express clients**. Before you install the SSL component, make sure that you have all of the prerequisite programs installed. Additionally, you should already have **set up the Client Access Express servers to use SSL**.

To install the SSL component of Client Access Express on a PC client, first install at least one of the Client Encryption (5722-CEx) products on your system. For V5R1 Client Access Express, you must install the V5R1 version of CE2 or CE3. Older versions of CE2 or CE3 are not compatible with V5R1 Client Access Express.

Depending on whether or not you already have Client Access Express installed, follow the appropriate steps below to install its SSL component.

If you do not have Client Access Express installed, follow these steps:

1. Because of export regulations for products that contain encryption technology, the Client Encryption products are installed with data authority set to *PUBLIC EXCLUDE. Authorize the appropriate userid to the Client Encryption product from a command prompt. The product is located at QIBM/ProdData/CA400/Express/SSL/SSLnum, where num is the bit key length (56 or 128) for the version of the Client Encryption product that you are using.

   **Note:** The SSL component will NOT appear in the Client Access Express selective install component list until you have set up these permissions correctly.

2. Follow the instructions for installing Client Access Express in the "Using NetServer to install Client Access Express" section of the **Client Access Express for Windows - Setup** book. When the install wizard displays the **Component Selection** window, be sure to select **Secure Sockets Layer (SSL)** and the appropriate **Client Encryption** product under it.

If you already have Client Access Express installed, follow these steps:

1. Because of export regulations for products that contain encryption technology, the Client Encryption products are installed with data authority set to *PUBLIC EXCLUDE. Follow these steps to authorize the appropriate userid to the Client Encryption product:
   
   b. Expand your system.
   c. Expand **File Systems**.
   d. Expand **Integrated File System**.
e. Expand Root.

f. Expand QIBM.

g. Expand ProdData.

h. Expand CA400.

i. Expand Express.

j. Expand SSL.

k. Right-click on SSLnum (where num is the bit key length (56 or 128) for the version of the Client Encryption product that you are using) and select Permissions.

l. Use the Permissions dialog to give Public Read and Execute authorities, or to add specific users and groups to the object’s authorization list.

2. Use Client Access Express Selective Setup to add the Secure Sockets Layer (SSL) component and the appropriate Client Encryption product.

To start Selective setup, from the Windows desktop, click Start → Programs → IBM AS400 Client Access Express → Selective Setup.

Note: You must use iSeries NetServer to install the SSL component directly from the system, even if you previously installed Client Access Express through another method. This is due to regulations that govern distribution of programs that use encryption technology. For information on the iSeries NetServer requirement for Client Access Express, see Installing Client Express.

Adding CA certificates from an iSeries

You can add a CA certificate to your PC directly from the iSeries key database. Some public CA certificates come pre-loaded in the key database. The server establishes an SSL connection to the client using a certificate from that CA. You can also view the list of currently installed certificates. You can also add a certificate from an alternate source.

To add a CA certificate from the iSeries:

1. Open Operations Navigator.

2. Right-click the name of your system.


4. Select the Secure Sockets tab.

5. Click Download. This will download the iSeries Certificate Authority certificate automatically into the certificate key database.

6. You will be prompted for your key database password. Unless you have previously changed the password from the default, enter ca400. A confirmation message displays. Click OK.

The download button automatically updates the IBM Toolbox for Java PC key database. The download button will only install the a private CA certificate. If you are using a CA that is not shipped with Client Access Express, you must manually add it to the two the Client Access Express and IBM Toolbox for Java PC key databases. See Adding a CA certificate to the key database from another source for details.

Adding a CA certificate to the key database from another source

Adding a CA certificate to the Client Access Express key database allows the iSeries server to establish an SSL connection to the client using that CA. Some public CA certificates come pre-loaded in the key database. You can also view the list of currently installed certificates.

If you use a server certificate that was signed by a public CA certificate that does not come pre-loaded with Client Access Express, you will need to import the public CA certificate file into both of the Client Access Express key databases on your PC. The CA certificate file can be obtained from e-mail sent from the public CA, or by exporting an existing public CA from your browser. There are two ways to handle the
import. The first was is to use the Certificate Authority Downloader Utility. This utility handles importing a CA certificate from a file into both of the Client Access Express key databases on your PC. Read the help text that comes with the utility for instructions.

The second way is to use Client Access Express IBM Key Management and Java KeyringDB tools to import the CA certificate file into the PC key databases.

**Listing the CA certificates in the Client Access Express key database**

If a Certificate Authority (CA) certificate is in the Client Access Express key database, then iSeries can establish an SSL connection to the client using a certificate from that CA.

**Note:** A CA certificate may be in the key database, but marked as not trusted. In this case iSeries cannot establish an SSL connection to the client using a certificate from that CA.

To view a list of the CA in the Client Access Express key database, follow these steps:

2. Select the Secure Sockets tab.
3. Click IBM Key Management.
4. You will be prompted for your key database password. Unless you have previously changed the password from the default, enter ca400. A confirmation message displays. Click OK.

### Enabling SSL on Operations Navigator

Using SSL with Operations Navigator will allow you to administer your iSeries systems through a secure connection. Before you can use SSL with Operations Navigator, you must have already done the following:

- Installed the prerequisite programs.
- Set up digital certificates on your iSeries.
- Installed the SSL component of Client Access Express.

In addition, before you can administer specific iSeries functions and servers through Operations Navigator using a secure connection, you must set up the appropriate servers to use SSL. Operations Navigator uses IBM Toolbox for Java and uses the same servers.

To enable SSL for Operations Navigator, follow these steps:

2. Right-click on the system that you want to connect to using SSL, then select Properties.
3. Click the Secure Sockets tab.
4. Select Use Secure Sockets Layer (SSL).
5. Click Verify connection. Verification messages will begin to appear immediately. After all servers have been verified, click OK to close the verification box. If any of the servers fail verification, click the message to get details about why it failed.
6. Click OK.
7. Restart Operations Navigator. You need to restart Operations Navigator for the SSL setting to take effect.

After you restart Operations Navigator, the icon that represents your system that you just secured should display a picture of a lock, signifying that the system is secure. If you manage multiple systems with the Management Central component of Operations Navigator, you can secure it as well.

By default, enabling SSL for an Operations Navigator connection to a system also enables SSL for connections that are made to that system through other Client Access Express applications. For the following applications, you can override this setting if you do not want them to use SSL:

- **PC5250 emulation**
- **Data Transfer**
Note: For any Client Access Express application to use SSL, you must also set up the appropriate Client Access Express servers to use SSL.

Enabling SSL for PC5250 sessions

If SSL is enabled in Operations Navigator, then by default all of your Client Access Express PC5250 terminal and printer sessions are ready for SSL connections. You can override Operations Navigator settings within PC5250 sessions.

Note: For PC5250 to use SSL, you must also set up the appropriate Client Access Express servers to use SSL.

To set up new PC5250 sessions, refer to the instructions in the Express User's Guide. You can access the User's Guide through the Client Access Express folder on your PC. To change existing PC5250 sessions to use either of the above options, follow these steps:

1. Open Operations Navigator.
2. Right-click the name of your system that you would like to end.
3. Select Display emulator.
4. Select the Communication menu, then select Configure.
5. Click Properties.
6. In the Connection dialog, select the Use Secure Sockets Layer (SSL).
7. If you have required client authentication for Telnet sessions and have more than one valid client certificate, you will also need to select either Select certificate when connecting or Use default to determine which client certificate to use. If they have more than one valid/trusted client certificate to choose from.
8. Click OK.
9. Click OK.

Note: To have PC5250 clients authenticated, an administrator must first enable SSL for Telnet. Then clients must create a valid user certificate for themselves to enable client authentication for PC5250.

Enabling SSL for Data Transfer

If an SSL connection to a system is enabled in Operations Navigator, then by default the client's Data Transfer connection to that system is also SSL-enabled. You can override Operations Navigator settings within Data Transfer.

Note: For Data Transfer to use SSL, you must also set up the appropriate Client Access Express servers to use SSL.

To set up new data transfer sessions, refer to the instructions in the Express User's Guide. You can access the User's Guide through the Client Access Express folder on your PC. To change Data Transfer sessions for either of the above options, follow these steps:

1. Open the appropriate Data Transfer session.
2. From the File menu, select the Properties.
3. Click the Connection tab.
4. On the Connection page, select the appropriate security setting.
5. Click OK.
6. (optional) To save the new settings, select Save from the File menu.
Enabling SSL for OBDC

If an SSL connection to a system is enabled in Operations Navigator, then by default OBDC requests to that system are also SSL enabled. You can override Operations Navigator settings within OBDC. To set up new OBDC data sources, refer to the instructions in the Express User’s Guide. You can access the User’s Guide through the Client Access Express folder on your PC.

To change OBDC sessions for either of these options, follow these steps:
1. Open OBDC Administration.
2. On the User DSN page, double-click on the user Data Source for your iSeries system.
4. In the Connection Options dialog, select the appropriate security setting.
5. Click OK.
6. Click OK again.

Note: For OBDC to use SSL, you must also set up the appropriate Client Access Express servers to use SSL.
Chapter 9. Securing Management Central

With the Management Central component of Operations Navigator, you can manage multiple systems through a single central system. By using SSL with Management Central, you can manage those systems securely. To use SSL with Management Central, you must have installed the service pack PTF SI01907 for V5R1 Client Access Express (5722–XE1) on the PCs that use Management Central. You must also have installed the following fixes (PTFs) for Operating System/400 (5722-SS1) on your V5R1 central system and on all your endpoint systems (which should all be running V5R1):

- SI01375
- SI01376
- SI01377
- SI01378
- SI01838

To use SSL with Management Central, you must secure Client Access Express and Operations Navigator on the PC from which you run Management Central.

In a Management Central environment, you have two authentication levels:

**Server authentication**

Provides authentication of the endpoint system server certificate. The central system acts as an SSL client when connecting to an endpoint system. The endpoint system acts as an SSL server and must prove its identity by providing a certificate that was issued by a Certificate Authority that the central system trusts. For every endpoint system, you must have a valid certificate issued by a trusted CA.

**Client and server authentication**

Provides authentication of both the central system and the endpoint system certificates. This is considered a stronger security level than the server authentication level. In other applications, this is known as client authentication, where the client must supply a valid certificate and prove that it has access to the private key associated with the certificate. When the central system (SSL client) attempts to establish a connection with an endpoint system (SSL server), the central system and the endpoint system authenticate each other’s certificates for CA authenticity.

Unlike other applications, Management Central also provides authentication through a validation list, called Trusted Group validation list. Generally the validation list stores information that identifies the user, such as a user identification, and authentication information, such as password, personal identification number, or digital certificate. This authentication information is encrypted.

With other applications, you typically do not specify enabling "server and client authentication" because server authentication almost always occurs during an SSL session enablement. Many applications have optional client authentication that you can configure. Management Central uses the term "server and client authentication" instead of client authentication because of the dual role that the central system plays in the network. When PC users connect to the central system and SSL is enabled, the central system acts as a server; however, when the central system is connecting to an endpoint system, it acts as a client. The following illustration shows how the central system operates as both a server and client in a network.
To enable SSL you must complete the following:

**For server authentication (required)**

1. Configuring central system for server authentication
2. **Configuring endpoint systems for server authentication**

**Note:** You must complete these steps before completing client authentication.

**For client authentication (optional)**

1. **Configuring central system and endpoint systems for client authentication**

---

**Configuring central system for server authentication**

SSL allows you to secure transmissions between a central system and an endpoint system. SSL provides transport and authentication of certificates and encryption of data. An SSL-connection can only occur between an SSL-enabled central system and an SSL-enabled endpoint system. Your system administrator must do the required administration and configuration task before SSL will work properly. If SSL is not configured properly, then Management Central cannot communicate to the endpoint system. You must do the server authentication setup before you can do client authentication.

Before you can configure Management Central to use SSL, you must have installed the prerequisite programs and set up digital certificates on your iSeries.

1. **Associate a certificate with Management Central server**
   a. Start IBM Digital Certificate Manager on the central system.
   b. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See [Using Digital Certificate Manager](#) for information on setting up a certificate system.
   c. Click the **Select a Certificate Store** button.
   d. Select *SYSTEM. Click **Continue**.
   e. Enter the appropriate password for *SYSTEM certificate store. Click **Continue**.
   f. When the left navigational menu reloads, expand **Manage Applications**.
   g. Click **Update certificate assignment**.
   h. On the next screen, select **Server** application. Click **Continue**.
   i. Select the **AS/400 Management Central Server**. You must also assign the certificate to all the Client Access servers because Management Central uses those servers as well. See [Setting up Client Access Express servers to use SSL](#) for details.
   j. Click **Update Certificate Assignment** to assign a certificate to the Management Central server to use to establish its identity to Client Access Express clients.

   **Note:** If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client’s key database, you will need to add it in order to use SSL. [Adding CA certificates from an iSeries](#) on page 26 provides information on how to do this. Finish this procedure before beginning that one.
   k. Select a certificate from the list to assign to the server.
   l. Click **Assign New Certificate**.
   m. DCM reloads to the **Update Certificate Assignment** page with a confirmation message. When you are finished setting up the certificates for the Management Central server, click **Done**.

2. **Enable SSL on Management Central**

**Note:** If you have SSL turned on for Operations Navigator, you must turn it off before enabling SSL on Management Central. If SSL is on in Operations Navigator, it will try to use SSL to connect to your Management Central central system. However, since SSL is not yet configured on your central system, you will not be able to connect and you will not be able to configure SSL for Management Central.

   a. In Operations Navigator, right-click on **Management Central**, then select **Properties**.
   b. Click the **Security** tab.
c. Select **Use Secure Sockets Layer (SSL)**
d. Select **Server** for the authentication level.
e. Click **OK** to set this value on the central system.
f. At this point, you will be asked to restart the Management Central Server. Do **NOT** restart the Management Central Server at this time. You must first configure all **endpoint systems** to use SSL and server authentication.

---

### Configuring endpoint systems for server authentication

After you have enabled SSL on the **central system for server authentication**, you need to enable SSL for all endpoint systems for server authentication. To configure endpoint systems to use SSL and server authentication, complete these tasks:

1. **Configure Application Administration to allow SSL**
   a. In Operations Navigator, expand **My Connections** (or your active environment).
   b. Right-click the system name of a system on which you want to use SSL, and select **Application Administration**. Be sure to do this for your central system and all endpoint systems on which you want to use SSL.
   c. Click the **Host Applications** tab.
   d. Expand **Digital Certificate Manager (DCM)**.
   e. Select **SYSTEM certificate store** and click **Customize**.
   f. In the users and groups list, expand **All Users**.
   g. From the users and groups list, select **QYPSJSVR** and click **Add**.
   h. Click **OK** to close Customize Access.
   i. Click **OK** to close Application Administration.
   j. Repeat steps 1b-1i for your central system and all endpoint systems on which you want to use SSL.

2. **Compare and update system values for the endpoint systems**
   a. Right-click **AS/400 System Groups**.
   b. Select **New System Groups**.
   c. Define a new system group that includes all the endpoint systems to which you want to connect using SSL.
   d. To display your new group, refresh the list of system groups.
   e. Under **My Connections** (or your active environment), right-click your central system, select **Inventory**, and then select **Collect**. Collect the system values inventory for the central system.
   f. When the inventory has been collected, right-click the new system group and select **System Values** and then **Compare and Update**.
   g. Verify the central system displays in the **Model system** field on the **Compare and Update System Values** dialog.
   h. Select the Management Central category and verify that Use Secure Sockets Layer is set to **Yes**. Also verify that the **SSL authentication level is set to Server**. These values are set on the central system in Step 2 of **Configuring central system for server authentication**.
   i. Click **OK** to set these values on the endpoint systems in the new system group.
   j. Wait for the **Compare and Update** to complete before restarting the Management Central Server. The **Compare and Update** may take a few minutes.

3. **Restart the Management Central server on the central system and all endpoint systems**
   a. In Operations Navigator, expand **My Connections** (or your active environment).
   b. Expand the system that you are using as your central system.
   c. Expand **Network→ Servers→TCP/IP**.
4. Configure SSL on your central system
   a. In Operations Navigator, expand My Connections.
   b. Right-click your central system, and select Properties.
   c. Click the Secure Sockets tab and select Use Secure Sockets Layer (SSL) for connection.
   d. Exit Operations Navigator and restart it.
   
   **If you want to enable client authentication, go to Configuring central system and endpoint systems for client authentication.**

---

**Configuring central system and endpoint systems for client authentication**

After you have completed server authentication configuration for Management Central, you can configure server and client authentication. Client authentication provides validation of Certificate Authority and trusted group for both the endpoint systems and the central systems. When the central system (SSL client) tries to use SSL to connect to an endpoint system (SSL server), the central system and the endpoint system authenticate each other’s certificates through client authentication (called Certificate Authority and Trusted Group authentication in Management Central). To ensure this level of validation, complete these steps:

Because you assigned a certificate when you configured server authentication, you do not need to complete steps within DCM when configuring client authentication.

1. **Configure the Management Central server to use SSL and client and server authentication**
   a. In Operations Navigator, right-click Management Central and select Properties.
   b. Click the Security tab and select Use Secure Sockets Layer (SSL).
   c. Select Client and server for the authentication level.
   d. Click OK to set this value on the central system.
   e. At this point, you will be asked to restart the Management Central server. You must first complete Steps 2 and 3 before you can restart the Management Central servers.

2. **Compare and update system values for the endpoint systems**
   a. Right-click AS/400 System Groups.
   b. Select New System Groups.
   c. Define a new system group that includes the central system and all the endpoint systems to which you want to connect using client and server authentication.
   d. To display your new group, refresh the list of system groups.
   e. Under My Connections (or your active environment), right-click your central system, select Inventory, and then select Collect. Collect the system values inventory for the central system.
   f. When the inventory has been collected, right-click the new system group and select System Values and then Compare and Update.
   g. Verify that the system that you set up as the central system displays in the Model system field on the Compare and Update System Values dialog.
   h. Select the Management Central category and verify that Use Secure Sockets Layer is set to Yes. Also verify that the SSL authentication level is set to Client and Server. These settings should have been configured on the central system in Step 1.
i. Click OK to set these values on the endpoint systems. Wait for these tasks to finish before restarting the server on the central system and endpoint systems.

3. **Copy the validation list to the endpoint systems**
   a. In Operations Navigator, expand **Management Central**.
   b. Expand **Definitions**.
   c. Right-click **Package**, and select **New Definition**.
   d. In the **New Definition** window, complete these fields:
      - **Name**: Enter a name of the validation list.
      - **Description**: Enter a description of the validation list.
      - **Source system**: Enter the name of the central system from which you will be copying the validation list.
      - **Selected files and folders**: Specify the validation list file of the central system. Your specified file name is /QSYS.1ib/QUSRYS.1ib/QYPSVLDL.VLDL.
   e. Click the **Options** tab. Because your target systems have the same file name already, select **Replace the existing file with the file being sent**.
   f. Click **Advanced**. In the **Advanced Options** window, specify **Yes** to allow object differences during a restore operation.
   g. Click OK. Refresh the list of definitions to display your new package.
   h. Right-click the new package, and select **Send**. In the next window, select the trusted group and click OK. The Send task will always fail on the central system, because that is always the source system. The Send task should complete successfully on all endpoint systems.

4. **Restart the server on both the central system and the endpoint systems**
   a. In Operations Navigator, expand **My Connections** (or your active environment).
   b. Expand the system that you are using as your central system.
   c. Expand **Network→ Servers→TCP/IP**.
   d. Right-click **Management Central** and select **Start**. (If **Start** is not available, select **Stop**. When the Management Central server has stopped, right-click it again and select **Start**.)
   e. Expand the endpoint system where you want to restart the server.
   f. Expand **Network→ Servers→TCP/IP**.
   g. Right-click **Management Central** and select **Start**.
   h. Repeat Steps 4e-4g for each endpoint system.

To see a list of frequently asked questions about using SSL with Management Central, see the [Management Central web page](#).
Chapter 10. Securing Telnet

When you enable the Telnet server on your system to use SSL, you can establish secure Telnet connections to your system from Client Access Express or from any other SSL-enabled Telnet client, such as a Personal Communications emulator. Before you can configure the Telnet server to use SSL, you must have installed the prerequisite programs and set up digital certificates on your system. New for V5R1, Telnet server supports client authentication as an optional component in SSL configuration. Client Authentication occurs when the server verifies the identity of the client by authenticating the client certificate passed up to the server application.

1. **Associate a certificate with the Telnet server**
   b. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See [Using Digital Certificate Manager](#) for information on setting up a certificate system.
   c. Follow these steps to associate a certificate with the Telnet server:
      d. Click the Select a Certificate Store button.
      e. Select *SYSTEM. Click Continue.
      f. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
      g. When the left navigational menu reloads, expand Manage Applications.
      h. Click Update certificate assignment.
      i. On the next screen, select Server application. Click Continue.
      j. Select the OS/400 TCP/IP Telnet Server.
      k. Click Update Certificate Assignment to assign a certificate to the OS/400 TCP/IP Telnet Server to use to establish its identity to Client Access Express clients.
         
         **Note:** If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client’s key database, you will need to add it in order to use SSL. [*Adding CA certificates from an iSeries* on page 23](#) provides information on how to do this. Finish this procedure before beginning that one.
      l. Select a certificate from the list to assign to the server.
      m. Click Assign New Certificate.
      n. DCM reloads to the Update Certificate Assignment page with a confirmation message. When you are finished setting up the certificates for the Telnet server, click Done.

2. **Enable client authentication for the Telnet server (optional step)**
   If you want the Telnet server to authenticate clients, you must update the application specifications in IBM Digital Certificate Manager.
   a. Start IBM Digital Certificate Manager. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See [Using Digital Certificate Manager](#) for information on setting up a certificate system.
   b. Click the Select a Certificate Store button.
   c. Select *SYSTEM. Click Continue.
   d. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   e. When the left navigational menu reloads, expand Manage Applications.
   f. Click Update application definition.
   g. On the next screen, select Server application. Click Continue.
   h. Select the OS/400 TCP/IP Telnet Server.
   i. Click Update Application Definition.
   j. In the table that displays, select Yes to require client authentication.
   k. Click Apply.
I. DCM reloads to the **Update Application Definition** page with a confirmation message. When you are finished updating the application definition for the Telnet server, click **Done**.

For an example of what a client needs to do to enable SSL for a Telnet application, see [Enabling client authentication for a PC5250 session](#).

3. **Enable SSL on the Telnet server**
   a. Open Operations Navigator.
   b. Expand **My Connections** – **Network** – **Servers** – **TCP/IP**.
   c. Right-click **Telnet**.
   d. Select **Properties**.
   e. Select the **General** tab.
   f. Choose one of these options for SSL support:

   - **Secure only**
     Select this to allow only SSL sessions with the Telnet server.
   - **Non-secure only**
     Select this to prohibit secure sessions with the Telnet server. Attempts to connect to an SSL port will not connect.
   - **Both secure and non-secure**
     Allows both secure and non-secure sessions with the Telnet server.

---

**Enabling client authentication for a PC5250 session**

After you have configured SSL for the Telnet server and specified to use client authentication, users will be required to provide a valid and trusted client certificate to the Telnet server for each connection attempt.

Clients need to create a user certificate and import that certificate to IBM Key Management database before client authentication will work.

1. **Create a user certificate in DCM**
   a. Start IBM Digital Certificate Manager. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See [Using Digital Certificate Manager](#) for information on setting up a certificate system.
   b. Expand **Create Certificate**.
   c. Select **User Certificate**. Click **Continue**.
   d. Complete the **User Certificate** form. Only those fields marked "Required" need to be completed. Click **Continue**.
   e. Depending on the browser you use, you will be asked to generate a certificate that will be loaded into your browser. Follow the directions provided by the browser.
   f. When the **Create User Certificate** page reloads, click **Install Certificate**. This will install the certificate in the browser.
   g. Using either your browsers certificate database, export the certificate to your PC. You will be prompted for a password to protect this file. You will use this password when you complete step 2 must place the certificate in a password protected file.

   **Note:** Microsoft Internet Explorer 5 or Netscape 4.5 are required to use the export and import functions.

2. **Import the certificate to the IBM Key Management**

   **Note:** You must add the Certificate Authority that created the client certificate to the PC key database, otherwise the import of the client certificate will not work.
   a. Click **Start** – **Programs** – **IBM AS/400 Client Access Express** – **Client Access Properties**.
   b. Select the **Secure Sockets** tab.
c. Click IBM Key Management.

d. You will be prompted for your key database password. Unless you have previously changed the password from the default, enter ca400. A confirmation message displays. Click OK.

e. From the pull-down menu, select Personal certificates.

f. Click Import.

g. In the Import key display, enter the file name and path for the certificate. Click OK.

h. Enter the password for the protected file. This is the same password that you created in Step 1g. Click OK. When the certificate has been successfully added to your personal certificates in IBM Key Management, you can use PC5250 emulator or any other Telnet application.

3. Start a PC5250 emulator session from Operations Navigator

a. Open Operations Navigator.

b. Right-click the name of your system that you have set up client authentication for Telnet.

c. Select Display emulator.

d. Select the Communication menu, then select Configure.

e. Click Properties.

f. In the Connection dialog, select the Use Secure Sockets Layer (SSL).

g. If you have more than one client certificate, select either Select certificate when connecting or Use default to determine which client certificate to use.

h. Click OK.

i. Click OK.
Chapter 11. Securing FTP

New for V5R1, the File Transfer Protocol (FTP) server provides enhanced security while sending and receiving files over an untrusted network. FTP server uses SSL to secure passwords and other sensitive data during an information exchange. FTP supports either SSL or TLS protected sessions, including client authentication and automatic sign-on. Most SSL-enabled applications connect a client to separate TCP ports, one port for "unprotected" sessions and the other for secure sessions. However, secure FTP is a bit more complex. A client can choose a secure FTP port (usually TCP port 990), where the session is set up with SSL initially. However, a client can also connect to a non-encrypted TCP port (usually TCP port 21) and then negotiates authentication and encryption options. V5R1 FTP provides for both of these options. Before you can configure the FTP server to use SSL, you must have installed the prerequisite programs and set up digital certificates on your iSeries.

1. Associate a certificate with the FTP server
   a. Start IBM Digital Certificate Manager. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See Using Digital Certificate Manager for information on setting up a certificate system.
   b. Click the Select a Certificate Store button.
   c. Select *SYSTEM. Click Continue.
   d. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   e. When the left navigational menu reloads, expand Manage Applications.
   f. Click Update certificate assignment.
   g. On the next screen, select Server application. Click Continue.
   h. Select the OS/400 TCP/IP FTP Server.
   i. Click Update Certificate Assignment to assign a certificate to the OS/400 TCP/IP FTP Server.
   j. Select a certificate from the list to assign to the server.
   k. Click Assign New Certificate.
   l. DCM reloads to the Update Certificate Assignment page with a confirmation message. When you are finished setting up the certificates for the FTP server, click Done.

2. Enable client authentication for the FTP server (optional step)
If you want the FTP server to authenticate clients, you must update the application specifications in IBM Digital Certificate Manager.

   Note: If the FTP client connects and client authentication is enabled for the server, the client must still enter a user subcommand. Once the user subcommand information is entered, FTP server will check that the user matches the profile associated with the client certificate that the client sent to the server as part of the SSL handshake. If the user matches the client certificate, no password is needed and FTP server will log the user onto the system. The user subcommand is needed because there is no mechanism in the FTP protocol to "inform" the client that it’s logged on without the command.

   a. Start IBM Digital Certificate Manager. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See Using Digital Certificate Manager for information on setting up a certificate system.
   b. Click the Select a Certificate Store button.
   c. Select *SYSTEM. Click Continue.
   d. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   e. When the left navigational menu reloads, expand Manage Applications.
   f. Click Update application definition.
   g. On the next screen, select Server application. Click Continue.
h. Select the **OS/400 TCP/IP FTP Server**.
i. Click **Update Application Definition**.
j. In the table that displays, select **Yes** to require client authentication.
k. Click **Apply**.
l. DCM reloads to the **Update Application Definition** page with a confirmation message. When you are finished updating the application definition for the FTP server, click **Done**.

3. **Enable SSL on the FTP server**
   a. Open Operations Navigator.
   b. Expand **My Connections**→**Network**→**Servers**→**TCP/IP**.
   c. Right-click **FTP**.
   d. Select **Properties**.
   e. Select the **General** tab.
   f. Choose one of these options for SSL support:
      - **Secure only**
        Select this to allow only SSL sessions with the FTP server. Connections may be made to the non-secure FTP port, but the FTP client must negotiate an SSL session before the user is allowed to log in.
      - **Non-secure only**
        Select this to prohibit secure sessions with the FTP server. Attempts to connect to an SSL port will not connect.
      - **Both secure and non-secure**
        Allows both secure and non-secure sessions with the FTP server.

**Note:** You do not need to restart the FTP server. It will dynamically detect that a certificate has been assigned to it. If it does not dynamically detect this change, verify that you have the latest PTFs applied to your iSeries system.
Chapter 12. Securing the Directory Services server

Using SSL with Directory Services server will allow you to manage your directory securely. You can configure the Directory Services server to accept SSL connections from Directory Services clients. You can also configure your system to use SSL when publishing information to the Directory Services server. Directory Services Server supports client authentication. Before you can configure a Directory Services server to use SSL, you must have installed the prerequisite programs and set up digital certificates on your system.

1. **Associate a certificate with the Directory Services server**
   a. If you want to manage your Directory Services server through an SSL connection from Operations Navigator, configure Client Access Express to use SSL. If you are planning to allow both SSL and non-SSL connections to the directory server, you may choose to skip this step.
   b. Start IBM Digital Certificate Manager.
   c. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. See Using Digital Certificate Manager for information on setting up a certificate system. There are three Directory Services servers that you must provide certificates for to enable SSL on Directory Services server. They are: Directory Services server, Directory Services publishing, and Directory Services client.
   d. Click the Select a Certificate Store button.
   e. Select *SYSTEM. Click Continue.
   f. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   g. When the left navigational menu reloads, expand Manage Applications.
   h. Click Update certificate assignment.
   i. On the next screen, select Server application. Click Continue.
   j. Select the Directory Services server.
   k. Click Update Certificate Assignment to assign a certificate to the Directory Services server to use to establish its identity to Client Access Express clients.

   **Note:** If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client’s key database, you will need to add it in order to use SSL. [Adding CA certificates from an iSeries on page 26](#) provides information on how to do this. Finish this procedure before beginning that one.
   l. Select a certificate from the list to assign to the server.
   m. Click Assign New Certificate.
   n. DCM reloads to the Update Certificate Assignment page with a confirmation message. When you are finished setting up the certificates for the Directory Services server, click Done.

2. **Associate a certificate for the Directory Services publishing.** (optional step) If you also want to enable publishing from the system to a Directory Services server through an SSL connection, you must also associate a certificate with the Directory Services publishing.
   b. Click the Select a Certificate Store button.
   c. Select *SYSTEM. Click Continue.
   d. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   e. When the left navigational menu reloads, expand Manage Applications.
   f. Click Update certificate assignment.
   g. On the next screen, select Client application. Click Continue.
   h. Select the Directory Services publishing.
i. Click **Update Certificate Assignment** to assign a certificate to the Directory Services publishing that will establish its identity.

**Note:** If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client’s key database, you will need to add it in order to use SSL. [Adding CA certificates from an iSeries](#) on page 26 provides information on how to do this. Finish this procedure before beginning that one.

j. Select a certificate from the list to assign to the server.

k. Click **Assign new certificate**.

l. DCM reloads to the **Update Certificate Assignment** page with a confirmation message.

**Note:** These steps assume that you are already publishing information to the Directory Services server with a non-SSL connection. See **Publishing information to the directory server** for complete information on setting up publishing.

3. **Associate a certificate for the Directory Services client.** (optional step) If you have other applications that use SSL connections to a Directory Services server, you must also need to associate a certificate with a the Directory Services client.
   
   
b. Click the **Select a Certificate Store** button.
   
c. Select *SYSTEM. Click **Continue**.
   
d. Enter the appropriate password for *SYSTEM certificate store. Click **Continue**.
   
e. When the left navigational menu reloads, expand **Manage Applications**.
   
f. Click **Update certificate assignment**.
   
g. On the next screen, select **Client** application. Click **Continue**.
   
h. Select the **Directory Services client**.
   
i. Click **Update Certificate Assignment** to assign a certificate to the Directory Services client that will establish its identity.

   **Note:** If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client’s key database, you will need to add it in order to use SSL. [Adding CA certificates from an iSeries](#) on page 26 provides information on how to do this. Finish this procedure before beginning that one.

j. Select a certificate from the list to assign to the server.

k. Click **Assign New Certificate**.

l. DCM reloads to the **Update Certificate Assignment** page with a confirmation message.

For additional information, see [Using Secure Sockets Layer (SSL) security with the Directory Services server](#). For additional information on DCM, refer to the online help or to [Using Digital Certificate Manager](#).
Chapter 13. Securing DRDA/DDM

You can use SSL to secure database client connections to the Distributed Data Management (DDM) server. This provides secure connections when you work with clients such as IBM Toolbox for Java and Client Access OLE DB Provider that support SSL for record level access. It also provides secure connections with any DRDA application requester products or DDM file I/O clients that support SSL available from independent software vendors. Before you can configure the DDM server to use SSL, you must have installed the prerequisite programs and set up digital certificates on your iSeries.

1. Associate a certificate to the DDM server
   a. If you want to use secure connections to the server from Client Access Express, configure Client Access Express to use SSL.
   b. Start IBM Digital Certificate Manager.
   c. If you need to obtain or create certificates, or otherwise setup or change your certificate system, do so now. For information on setting up a certificate system, see Using Digital Certificate Manager.
   d. Click the Select a Certificate Store button.
   e. Select *SYSTEM. Click Continue.
   f. Enter the appropriate password for *SYSTEM certificate store. Click Continue.
   g. When the left navigational menu reloads, expand Manage Applications.
   h. Click Update certificate assignment.
   i. On the next screen, select Server application. Click Continue.
   j. Select the OS/400 DDM/DRDA Server - TCP/IP.
   k. Click Update Certificate Assignment to assign a certificate to the DDM/DRDA server to use to establish its identity to Client Access Express clients.

   **Note:** If you choose a certificate from a CA whose CA certificate is not in your Client Access Express client's key database, you will need to add it in order to use SSL. Adding CA certificates from an iSeries on page 26 provides information on how to do this. Finish this procedure before beginning that one.
   l. Select a certificate from the list to assign to the server.
   m. Click Assign New Certificate.
   n. DCM reloads to the Update Certificate Assignment page with a confirmation message. When you are finished setting up the certificates for the DDM/DRDA server, click Done.

To use SSL with the iSeries DDM/DRDA TCP/IP server, your client must connect to the correct port on the server.

For additional information on DRDA Security, see DRDA Security using TCP/IP.
Chapter 14. Securing your Java applications with SSL

You can use SSL to secure communications for the applications that you develop with Developer Kit for Java. Client applications that use IBM Toolbox for Java can also take advantage of SSL. The process for enabling SSL for your own Java applications is slightly different than enabling it for the other applications discussed in Securing applications with SSL. However, it is similar in that you must install the prerequisite programs and set up digital certificates on your system.

See the following IBM Toolbox for Java topics, for more information:

- Making a Java application secure with secure sockets layer provides an overview of tasks that you need to complete to work with SSL and a Java application.
- Changing your Java code to use secure sockets layer provides an overview of changes that you need to implement SSL with Java applications.
- Examples: Modifying your Java client to use secure sockets layer provides example Java client programs that use SSL.
- Examples: Modifying your Java server to use secure sockets layer provides example Java server programs that use SSL.