Release Notes

IBM Sterling Certificate Wizard

Version 1.4.00

Update Date: December 31, 2013

Release Notes

IBM Sterling Certificate Wizard

Version 1.4.00

Update Date: December 31, 2013

Note

Before using this information and the product it supports, read the information in “Notices” on page 17.

This edition applies to 1.4.00 Version of IBM® Sterling Certificate Wizard and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Understanding Certificates and IBM Sterling Certificate Wizard

About Certificates and IBM Sterling Certificate Wizard

Use IBM® Sterling Certificate Wizard to generate and obtain the components necessary for secure connections that use the Secure Sockets Layer (SSL) protocol, the Transport Layer Security (TLS) Protocol, SMIME messaging, or SSH keys.

Establishing a secure connection using SSL or TLS requires a key certificate file, which contains information necessary for authentication. Before you can create the key certificate file, you must first acquire a certificate through a certificate authority (CA) or generate a self-signed certificate. Use Sterling Certificate Wizard to perform the following procedures:

- Generate the Certificate Signing Request (CSR)—Use this procedure to generate the certificate signing request. After generating the CSR, send this information to a CA. The CA issues a signed certificate and provides a trusted root to authenticate the certificate.
- Generate a Self-Signed Certificate—If you do not plan to use a third party as a certificate authority, generate a self-signed certificate to create the certificate file and the private key file.
- Create a Key Certificate File—After you receive a certificate from a CA or generate a self-signed certificate, use this procedure to create a key certificate file. You can also use this procedure to create a key certificate in Java™ Key Store format or in PKCS12 format.
- Generate a CA Certificate Chain
- Verify Certificate Files—After you create the key certificate file, use this procedure to verify that the key certificate file will function correctly when used to secure communications with a trading partner or to view the contents of a certificate file. You identify the files that you want verified including a key certificate file, a trusted root file, or a certificate file. Each file that you identify is verified separately and the results of each verification are submitted in individual windows. This is the only method available to view the contents of a certificate file since the file is encrypted.
- Import to Trust Store—Use this procedure to add a certificate to a Trust Store file.

  Establishing a secure connection using SSH requires a key pair, which contains information necessary for authentication. Use Sterling Certificate Wizard to generate the SSH keys.

Security Terms and Concepts

The following table defines terms and concepts used in conjunction with Sterling Certificate Wizard listed in alphabetical order. An understanding of this information will assist you in working with Sterling Certificate Wizard.

Term	Definition
Authentication	The process of verifying that a particular name really belongs to a particular entity and assurance that a message has not been modified in transit or storage.

Term	Definition
CA certificate	A certificate issued by a certificate authority (CA) and used by applications to verify that trusted certificates exchanged between nodes are created and certified by the issuing CA and to verify that the digitally signed CA certificate is a valid trusted root file.
Certifcate	A certificate is obtained from a certificate authority by generating a certificate signing request (CSR) that contains specific information in a specific format about the requester. It typically contains: (1) distinguished name and public key of the server or client; (2) distinguished name and digital signature of the certificate authority; (3) period of validity (certificates expire and must be renewed); and (4) administrative and extended information. The certificate authority analyzes the CSR fields, validates the accuracy of the fields, generates a certificate, and sends it to the requester. A certificate can also be self-signed.
Certificate Authority	A Certificate Authority (CA) is a company that is responsible for verifying and processing certificate requests, and issuing and managing certificates. The CA you choose should be one that your trading partners will trust. You must meet the requirements for the CA you choose.
Certificate revocation list	A list of certificates that have been revoked.
Certificate Signing Request (CSR)	An output file sent via E-mail to a Certificate Authority to request an X.509 certificate.
Cipher suite	A cryptographic algorithm used to encrypt and decrypt files and messages.
Cipher text	Data that has been encrypted. Cipher text is unreadable until it has been converted into plain text (decrypted) with a key.
Decryption	Any process to convert cipher text back into plain text.
Digital Certificate	A digital certificate is a specifically formatted document that allows you to authenticate or identify yourself to a Web browser, an E-mail reader, a secure server, or a client. It contains information on who you are, your relevant details, and who issued the certificate. A certificate can be tied to an E-mail address, a Web server or a company, and in each case the certificate can be used for different things. A basic E-mail certificate allows you to prove that you are who you say you are. It also allows you to store more information about yourself: your place of work, your telephone contact details—anything you want. The certificate also contains your public key. This means that your certificate becomes associated with your key.
Digital signature	When a message digest is encrypted with a private key, the result is a digital signature. Digital signatures allow a client to authenticate the server, because the client has the server's public key and can use it to decrypt the signature (created with the private key). The client knows the server is the only one who has the private key, so the server must be the one that sent the message. A server may also authenticate a client.
Encryption	Any process used to convert plain text into cipher text.
FTP	Internet application and network protocol for transferring files between host computers. File transfer protocol.

Term	Definition
Java	A programming language that allows development of applications that can run from any kind of device or machine—a PC, a Macintosh computer, a network computer, the Internet, or a mobile phone. The Java language makes it possible to develop software that is portable, modular, and secure.
Java Key Store (JKS)	A Java based format for creating a key certificate file.
JDK	The Java Development Kit (JDK) contains the software and tools that developers need to compile, debug, and run applets and applications written using the Java programming language.
JRE	The Java Runtime Environment (also known as the Java Runtime or JRE) consists of the Java virtual machine, the Java platform core classes, and supporting files. It is the runtime part of the Java Development Kit and provides no compiler, debugger, or tools. The JRE is the smallest set of programs and files that constitute the standard Java platform.
Key Certificate File	A file containing certificate and encrypted private key. This file is stored on the client that contains an encrypted message to identify the client and enable client/server authentication during secure FTP connections.
Keys	A collection of bits, usually stored in a file, which is used to encrypt or decrypt a message.
Passphrase	Similar to a password but can be made up of any number of characters that can be entered, including spaces. A passphrase is generally thought to be stronger than a password, although not many programs support the use of a passphrase.
Password	A character-limited word or phrase that establishes identity to allow access to a system. Generally, a password is composed of letters, numbers, or both.
PKCS #12 Certificate	A common certificate format containing both the public and private key information. The file uses and an extension of .pfx or .p12. This format is commonly used to securely store encrypted private keys and certificates. Key certificates exported from Gentran® Integration Suite are exported in this format.
Private Key	The secret key of a public-private key cryptography system. This key is used to sign outgoing messages, and is used to decrypt incoming messages.
Public Key	The public key of a public-private key cryptography system. This key is used to confirm signatures on incoming messages or to encrypt a file or message so that only the holder of the private key can decrypt the file or message. A public key is disseminated freely to clients and servers via certificates signed by a certificate authority (CA).
Public Key Cryptography Standards (PKCS)	A cryptography standard for exchanging personal identity information such as private keys and certificates.
Secure Sockets Layer (SSL)	Secure Sockets Layer (SSL) is a protocol that provides secure communications with transport protocols, including FTP, over TCP/IP. It is an open, non-proprietary Internet protocol that has been widely adopted as standard. SSL ensures point-to-point security, meaning that the data is secured as it is transmitted across a single socket.
Self-signed Certificate	A certificate that identifies your organization rather than a public certificate authority in the file. It is often used during the period between your request and receipt of a certificate from a public certificate authority. If self-signed certificates are used, the trusted root signing certificate must be installed in the client manually.

Term	Definition
Session Key	Crypto key intended to encrypt data for a limited period of time, typically only for a single communications session between a pair of entities. When the session is over, the key is discarded and a new one established when a new session takes place.
SMIME	(S-MIME) A specification for adding authentication and encryption security to MIME formatted messages.
Third-party Certificate	A certificate that identifies an organization other than those that are preconfigured for the application. If third-party certificates are used by the server, the corresponding trusted certificate must be installed in the client manually.
Transport Layer Security (TLS)	Transport Layer Security (TLS) is a protocol that provides secure communications with transport protocols, including FTP, over TCP/IP. It is an open, non-proprietary Internet protocol that has been widely adopted as standard. TLS ensures point-to-point security, meaning that the data is secured as it is transmitted across a single socket.
Trusted Root Certificate File	A file stored in a local directory on the client that contains a list of trusted sources. During secure connections, the client compares the server certificate, or vice versa, to the trusted root certificate file to determine if the server certificate was signed by a trusted source. The client can establish a secure FTP connection if a trusted source signed the server certificate.
Trust store	A file stored in a local directory on the client that contains a list of trusted sources in Java format. During secure connections, the client compares the server certificate, or vice versa, to the trusted root certificate file to determine if the server certificate was signed by a trusted source. The client can establish a secure connection if a trusted source signed the server certificate.
Unsecure connection	A connection that has no security.
X.509 Certificate	Public key certificate specification developed as part of the X.500 directory specification, and often used in public key systems.

Chapter 2. Supported System Certificate Formats

About Supported System Certificate Formats

Sterling Certificate Wizard produces the following system certificate formats:

Certificate Format	Description
DES MD5	Contains a standard key certificate. The private key is encrypted using PBE version 1.5 DES MD5.
AES-256 SHA-256	Contains a standard key certificate. The private key is encrypted using PBE version 2.0 AES-256 SHA-256.
AES-256 SHA1	Contains a standard key certificate. The private key is encrypted using PBE version 2.0 AES-256 SHA1.
AES-128 SHA1	Contains a standard key certificate. The private key is encrypted using PBE version 2.0 AES-128 SHA1.
3DES SHA1	Contains a standard key certificate. The private key is encrypted using PBE version 2.0 3DES SHA1.
PKCS#12 (3DES)	The Personal Information Exchange Syntax Standard is a portable format that is used to store or transport a user's private key, certificates and other embedded objects. The private key is encrypted using 3DES 192.
JKS	The Java Key Store is a proprietary keystore type that contains a user's private key and certificates in a protected format.

Use the following table to select a system certificate format to use with your IBM product.

Product	System Certificate Format to Use
IBM® Sterling Secure Proxy 3.3.01, 3.4.0, and 3.4.1.0	PKCS#12, DES MD5, PKCS#5 Ver 2.0 (AES-256 SHA-256, AES 256 SHA1, AES 128 SHA1, 3DES SHA1)
IBM® Sterling External Authentication Server 2.3.01, 2.4.0, and 2.4.1	JKS
IBM® Sterling Connect:Direct®® for z/OS® 4.6, 4.7, 4.8, and 5.0	PKCS#12 (3DES)
IBM® Sterling Connect:Direct® for UNIX 2.4.05 and 2.5.00	PKCS#5 Ver 2.0 (AES-256 SHA1, AES-128 SHA1, 3DES SHA1), DES MD5
IBM® Sterling Connect:Direct® for UNIX 3.7 and 3.6.01	DES MD5
IBM® Sterling Connect:Direct® for i5/OS® 3.6 and 3.5	DES MD5

Product	System Certificate Format to Use
IBM® Sterling Connect:Direct® for HP NonStop 3.4.03	AES-256 SHA1, AES-128 SHA1, 3DES SHA1, DES MD5
IBM® Sterling Connect:Direct® for OpenVMS 3.4	AES-256 SHA1, AES-128 SHA1, 3DES SHA1, DES MD5
IBM® Sterling Connect:Direct® for Microsoft Windows 4.4	AES-256 SHA1, AES-128 SHA1, 3DES SHA1, DES MD5
IBM® Sterling Connect:Direct® for Microsoft Windows 4.2 and 4.3	DES MD5
IBM® Sterling Connect:Direct® FTP+ 1.1	DES MD5
IBM® Sterling Connect:Direct® Select 1.1 and 1.2	DES MD5
IBM® Sterling Connect:Enterprise®® for z.OS® 1.4 and 1.5	PKCS#12 (3DES)
IBM® Sterling Connect:Enterprise® for UNIX 2.3, 2.4, 2.4.01, and 2.4.02	DES MD5
IBM® Sterling Connect:Enterprise® HTTP Option 1.3 and 1.4	DES MD5
IBM® Sterling Connect:Enterprise® Command Line Client 1.2 and 1.3	DES MD5, PKCS#12 (3DES)
IBM® Sterling Connect:Enterprise® Secure Client 1.3.02 and 1.4	DES MD5, PKCS#12 (3DES)
IBM® Sterling Control Center 4.1 and 4.2	JKS
IBM® Sterling Gentran Integration Suite 4.2 and 4.3	DES MD5, PKCS#12 (3DES)

For the most current compatibility information, please check the For the most current compatibility information, please check the IBM Sterling Certificate Wizard Release Notes®.

Chapter 3. Using Sterling Certificate Wizard

Generating a Certificate Signing Request

About this task

If you plan to use a CA certificate for SSL or TLS authentication, you can use the following procedure to generate the certificate signing request (CSR) that you send to the certificate authority (CA) to request the CA certificate. When Sterling Certificate Wizard generates a CSR, it creates two files: a private key and a file that contains the CSR.

Procedure

1. Click the **Generate CSR** tab.
2. Provide the following information and click **Next**.

Field Name	Description	Required?
Common Name	The URL or TCP/IP address of the server, up to 64 characters.	Yes
Country	Select the country from the list. If the country that you want is not in the list, select **Add a New One** from the drop-down list. Type the 2-character country code in the text box. **Note:** The country code is not permanently added to the list.	No
State/Province	Select the state or province from the list for United States values or type a value up to 128 characters.	No
City/Locality	Type a city or locality name, up to 128 characters.	No
Organization/ Company Name	Type an organization or company name, up to 64 characters.	No
Organizational Unit(s)	Type an organizational unit (OU), up to 64 characters. Type up to 4 OU values. Press **Enter** to separate each value.	No
Email Address	Type the Email address for the contact at your site, up to 128 characters.	No

3. Provide the following information and click **Next**.

Field Name	Description	Required?
Private Key Length	Select the length of the private key, from the drop-down list. Valid values include 512, 768, 1024, 2048, and 4096. **Note:** Not all servers support the large key length values.	Yes
Passphrase	Type the passphrase to use for encrypting the certificate's private key. The passphrase should be 6–256 characters. Write down this passphrase. You need this information to use the private key.	Yes

Field Name	Description	Required?
Confirm Passphrase	Retype the passphrase you selected for verification.	Yes

4. In the **Cipher** drop-down list, select a cipher to encrypt the private key. Choose one of the following:

Cipher	Description
AES256 SHA-256	Uses PKCS #5 version 2.0
AES256 SHA1	Default cipher. Uses PKCS #5 version 2.0
AES128 SHA1	Uses PKCS #5 version 2.0
3DES SHA1	Uses PKCS #5 version 2.0
DES MD5	Uses PKCS #5 version 1.5

5. Specify a name for the private key file and the CSR file, or click **Next** to accept the default names.
6. Click **Next**. A summary screen displays the values you selected for the CSR.
7. Click **Next**. A CSR is created.
8. Either copy the text for the CSR and paste this information into an email or a text file and send the information to the certificate authority (CA) or send the CSR file to the CA.

Completing the Certificate Signing Request

After you copy the CSR information that was generated by Sterling Certificate Wizard and send it to a CA, the CA verifies and processes the certificate request and issues a digitally signed certificate. It then sends you the certificate that identifies your site to other sites and provides a trusted root certificate to use to verify the certificate file. Save the certificate file and the trusted root file to the Sterling Certificate Wizard directory and complete the procedure Generating a Key Certificate, PKCS12, or Java Keystore File.

Generating a Key Certificate, PKCS12, or Java Keystore File

About this task

Note: Sterling Certificate Wizard requires that certificates received from CAs be Base64-encoded PEM format and encapsulated by the strings “-----BEGIN CERTIFICATE-----” and “-----END CERTIFICATE-----” or DER format.

Verify that your certificate is in this format before attempting to generate a key certificate file.

Use the following procedure to create a key certificate file in standard format, JKS format, or PKCS 12 format.

Procedure

1. Click the **Generate Key Certificate** tab.
2. To identify the output format of the key certificate file being created, select one of the following options from the **Output Keycert/Keystore Format** field:
   - **Standard**.
   - **JKS**.

- **PKCS12**.

3. Type the full path to the file that contains your private key in the **Private Key File Name** field, or click **Browse** to navigate to the file.
4. Type the passphrase associated with the private key in the **Private Key Passphrase** field.
5. To add a certificate to the key certificate file, type the full path to the file that contains the certificate in the **Certificate File Name** field and click **Add**, or click **Browse** to navigate to the file.
6. Repeat step 5 to create a chained key certificate file. The files in the chain are displayed in the **Certificate Chain List** window.

   Note: Certificates in Certificate Chain List that are not part of the certificate chain will not be added to the key certificate.
7. If the **Standard** output format was selected in Step 2, select a cipher in the **Cipher for Encrypting Private Key** drop-down list. Choose one of the following:

Cipher	Description
AES256 SHA-256	Uses PKCS #5 version 2.0
AES256 SHA1	Default cipher. Uses PKCS #5 version 2.0
AES128 SHA1	Uses PKCS #5 version 2.0
3DES SHA1	Uses PKCS #5 version 2.0
DES MD5	Uses PKCS #5 version 1.5

8. To check for the presence of the CA Root certificate in the keycert chain, select the **CA Root must be present** check box. If the check box is selected and the CA Root certificate is not present, you will receive a warning message.
9. To include the CA Root certificate in the output keycert file, select the **Include CA Root in Keycert** check box.
10. To create a standard key certificate, type the full path to the file that will be created for the key certificate or click **Browse** to navigate to an existing file in the **Key Certificate File Name** field.
11. To create a JKS key:
    - Type the path to the file of the Key Store in the **Keystore File Name** field, or click **Browse** to navigate to the file.
    - Specify an alias to identify the JKS key certificate in the key store file in the **Unique Key/Cert Alias** field.
    - Specify a new passphrase for the JKS key certificate in the **Key Store Passphrase** field.
12. To create a PKCS12 key:
    - Type the full path to the file of the Key Store in the **Keystore File Name** field, or click **Browse** to navigate to the file.
    - Specify a new passphrase for the key certificate in the **Key Store Passphrase** field.
13. Click **Generate**. A message dialog box is displayed when the key certificate file is generated successfully.

    Note: If you are creating a key certificate file using chained certificates, the certificates are verified to be in the chain when you generate the key certificate file.

14. Click **OK**.

Generating a CA Certificate Chain

Use this procedure to generate a CA certificate chain file that contains one or more root CA certificates and additional certificates that form certificate chains starting from the CA root certificate. The output file from this screen can be used as the trusted CA file in SSL/TLS handshakes.

Procedure

1. Click the **Generate CA Certificate Chain** tab.
2. To add a certificate to the certificate chain file, type the full path and file name of the certificate file and click **Add**, or click **Browse** to navigate to the file in the **Certificate File Name** field.
3. Repeat step 2 to add additional certificates. Certificates are displayed in the **Certificate Chain List**.
4. Type the full path and file name where the newly created certificate chain file will be created, or click **Browse** to navigate to the file in the **CA Chained Certificate Output File Name** field.
5. Click **Generate**. A message dialog box displays information about the newly created certificate chain file.
6. Click **OK**.

About Generating SSH Keys

When you create a SSH key pair, two files are generated: a private key file and a public key file with the same name but with the .pub extension. You can also change the passphrase used to encrypt the private key and convert SSH public keys from one format to another.

Generating a Key Pair

To generate an SSH key pair:

Procedure

1. Click the **Generate SSH Keys** tab.
2. From the **Action** drop-down list, select **Generate Key Pair**.
3. Type an optional comment to include with the public key file that is generated.
4. In the **Output File** field, type the full path and name of the private key file to create, or click **Browse** to navigate to the file. The public key is created using the same path and name with a .pub extension.
5. Provide the following information and click **Next**.

Field Name	Description	Required?
Passphrase	The passphrase, from 6 to 28 characters, to use to encrypt the private key. You need to remember the passphrase to use the private key.	Yes
Key Length	Select the length of the private key, from the drop-down list. Valid values include 512, 768, 1024, 2048, 3072, and 4096. **Note:** Not all servers support the large key length values.	Yes

Field Name	Description	Required?
Key Type	Select RSA or DSA key type from the drop-down list.	Yes
Public Key Format	The key pair is created using the OpenSSH public key format.	Yes

6. Click **Generate**. A message dialog box is displayed when the key pair files are generated successfully. Longer key lengths may take longer to generate.

Changing a Passphrase

To change the passphrase on an SSH key pair:

Procedure

1. Click the **Generate SSH Keys** tab.
2. From the Action drop-down list, select **Change Passphrase**.
3. In the **Input Private Key File** field, type the full path and file name of the file that contains the private key file information, or click **Browse** to navigate to the file.
4. In the dialog, provide the following information:

Field Name	Description	Required?
Old Passphrase	Type the existing passphrase associated with the private key.	Yes
New Passphrase	Type the new passphrase to use for encrypting the private key. The passphrase should be 6–28 characters. Write down this passphrase. You need this information to use the private key.	Yes

5. Click **Generate**. A message dialog box displays confirmation about the passphrase has been changed.
6. Click **OK**.

Converting an SSH Public Key to a Different Format

To convert the format of an SSH public key:

Procedure

1. Click the **Generate SSH Keys** tab.
2. Select the conversion to perform:
   - **Convert from OpenSSH to IETF SSH Key**
   - **Convert from IETF SSH to OpenSSH Key**
3. In the **Input Public Key File** field, type the full path and file name of the file that contains the public key, or click **Browse** to navigate to the file.
4. In the **Output Public Key File** field, type the full path and file name where the newly created public key will be written, or click **Browse** to navigate to the file.
5. Click **Generate**. A message dialog box displays confirmation about the newly created public key file.
6. Click **OK**.

Verifying a Certificate or Key Certificate File

Use this procedure to verify the contents of a digital certificate, a key certificate, or trusted root certificate. If you provide both the trusted root certificate and the certificate or key certificate, the certificate is verified against the trusted root certificate.

About this task

Note: For key certificates, PKCS12 and Java Keystore files are not currently supported.

Procedure

1. Click the **Verify Certificate** tab.
2. To verify a key certificate:
   - Type the passphrase associated with the key certificate in the **Passphrase** field.
   - Type the full path to the key certificate file, or click **Browse** to navigate to the file in the **Keycert file** field.
3. To verify a digital certificate, type the full path to the certificate file in the **Certificate File** field, or click **Browse** to navigate to the file.
4. In the **Trusted Root File** field, type the full path to the trusted root certificate file used to verify the key certificate or the certificate specified above or click **Browse** to navigate to the file.
5. Click **Verify**. Message dialog boxes display verification results for the key certificate file or certificate file you selected.

About Self-Signed Certificates

Consider the following when deciding to use CA or self-signed certificates.

Self-signed certificates

- Best use--Recommended for testing secure connections.
- ID Verification--Does not provide a reliable method to verify a certificate holder’s identity. You must rely on the authenticity of the self-signed certificate of your trading partner who gave you the certificate.
- Management problems--You and your trading partners must manage your self-signed certificates manually. Each time your certificate expires, you must create a new certificate and send it to all of your trading partners so that they can update the security settings of their software. If the client authentication option is used, your trusted CA certificate file will need to have as many certificates as there are trading partners using self-signed certificates. This could lead to performance and manageability issues.
- Cost--Certificates are free.

CA certificates

- Best use--Recommended for securing sessions in production environments.
- ID verification--A Certificate Authority signs a certificate to verify the identity of the certificate holder. This ensures that the certificate holder is who they say they are.

- Certificate management--All trading partners can use the same CA root certificate for session authentication. They do not need to update the root certificate when your server certificate expires.
- Cost--Certificates are not free, but only one is required for a server.

CW_Create_SS_Certificate

If you want to use a self-signed certificate to validate secure connections, use this procedure to create a self-signed certificate.

Procedure

1. Click the **Self-signed Certificate** tab.
2. Provide the following information and click **Next:**

Field Name	Description	Required?
Common Name	The host, URL or TCP/IP address of the server, up to 64 characters.	Yes
Country	Select the country from the list. If the country that you want is not on the list, select **Add a new one** from the drop-down list. Type the 2-character country code in the text box, and press **Enter**. **Note:** The country code is not permanently added to the list.	No
State/Province	Select the state or province from the list for United States values or type a value up to 128 characters.	No
City/Locality	Type a city or locality name, up to 128 characters.	No
Organization/ Company Name	Type an organization or company name, up to 64 characters.	No
Organizational Unit(s)	Type an organizational unit (OU), up to 64 characters. Type up to 4 OU values. Press **Enter** to separate each value.	No
Email Address	Type the Email address for the contact at your site, up to 128 characters.	No

3. Select one of the following options to identify the type of certificate to create:
   - **Server** to create a server certificate.
   - **Client** to create a client certificate.
   - **Custom** to define the certificate options to activate.
4. If you selected Custom in the previous step, select one or more of the following options to identify how the self-signed certificate will be used, and click **Next**.

Option	Description
Digital Signature	The key will be used as a digital signature to support entity authentication and data origin authentication with integrity.
KeyEncipherment	The public key will be used for key transport, such as using an RSA key for key management.
KeyAgreement	The public key will be used for key agreement, such as using a Diffie-Hellman key for key management.
CrlSign	The public key will be used for verifying a signature on revocation information.

Option	Description
Decipher Only	Used with the KeyAgreement option, when the public key may be used only for deciphering data while performing key agreement. **Note:** Turn on KeyAgreement in order for this option to be valid.
NonRepudiation	The certificate will be used for non-Repudiation, to protect against the signing entity falsely denying some action. This does not include certificate or CRL signing.
DataEncipherment	The public key will be used for enciphering user data. This cannot be used to encipher cryptographic keys.
KeyCertSign	The public key will be used for verifying a signature on Certificate Authority certificates.
Encipher Only	Use with the KeyAgreement option, when the public key may be used only for enciphering data while performing key agreement. **Note:** Turn on KeyAgreement in order for this option to be valid.

5. Select the length of the private key, from the drop-down list. Valid values include 512, 768, 1024, 2048, and 4096. Not all servers support the large key length values.
6. Type the following information about the certificate:
   - The serial number in the **Certificate Serial Number** box, from 1—256 numeric characters
   - How many days the certificate is valid in the **How Long Key Is Valid** box, from 1—9999 days
   - The passphrase to use for encrypting the certificate's private key in the **Passphrase** field, from 6–256 characters. You will need to know this passphrase to use the private key.
   - Retype the passphrase you selected for verification in the **Confirm Passphrase** field.
7. Click **Next**
8. In the **Cipher** drop-down list, select a cipher for encrypting the private key. Choose one of the following:

Cipher	Description
AES256 SHA-256	Uses PKCS #5 version 2.0
AES256 SHA1	Default cipher. Uses PKCS #5 version 2.0
AES128 SHA1	Uses PKCS #5 version 2.0
3DES SHA1	Uses PKCS #5 version 2.0
DES MD5	Uses PKCS #5 version 1.5

9. Type file names for the private key file and the self-signed certificate file, or click **Next** to accept the default names. A screen displays the information you provided in the previous steps.
10. Click **Next**. A notification screen is displayed when the self-signed certificate is generated.
11. Click **Finish** to create the self-signed certificate. A message is displayed when the file is created.
12. Click **OK** to close the message. Longer key lengths may take longer to generate.

Creating a Java Trust Store or Add a Certificate to a Java Trust Store

Use the following procedure to create a Java Trust Store, or add a certificate to an existing Java Trust Store. A Trust Store is a java key store defined by Sun Microsystems to store certificates.

Procedure

1. Click the **Import to Trust Store** tab.
2. Type the path and file name of the certificate that you want to import in the **Certificate File Name** field.
3. Type the path and file name of your Trust Store file in the **Trust Store File Name** field.
4. Type an alias to identify the entry in the Trust Store in the **Unique Certificate Alias** field.
5. Type the passphrase for the Trust Store in the **Trust Store Passphrase** field.
6. Click **Import**. A message is displayed when the certificate is imported to the Trust Store.

   Note: If the certificate file contains a chain of certificates, the alias is assigned to the first certificate. Aliases for additional certificates are created by appending a sequence number (alias1,alias2,alias3...).

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2011. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2011.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise, Gentran®, Gentran:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce®, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Product Number:

Printed in USA