IBM Sterling External Authentication Server

Implementation Guide

Version 2.3

Copyright

This edition applies to the 2.3 version of IBM Sterling External Authentication Server and to all subsequent releases and modifications until otherwise indicated in new editions.

Before using this information and the product it supports, read the information in *Notices* on page 181.

SEASIG1201

Contents

About Sterling External Authentication Server

Sterling External Authentication Server (EA) implements authentication and validation for Sterling Commerce products. EA clients connect to a server and administrators use the GUI to configure EA.

SSL or TLS authentication validates the connection between EA and the client. The client sends a request with a certificate chain and/or a user ID and password. EA uses the certificate validation or authentication definition to perform the operations.

SSH authentication uses the configuration to validate the client request. EA uses information in the profile to bind to an LDAP directory and look up the SSH key. It performs an assertion to match the key from the client to the keys found in the LDAP directory.

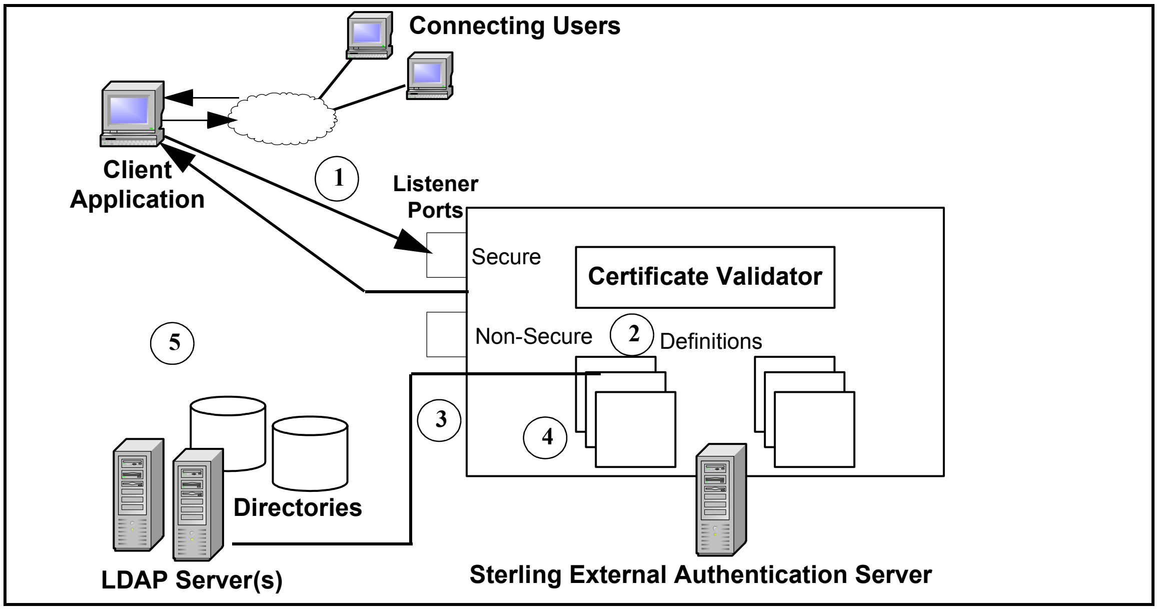
EA supports a flexible configuration to meet your certificate validation and user authentication needs. You can configure TCP ports, SSL or TLS, server connections, and logging.

After you configure the system, create certificate validation and user authentication definitions. A certificate validation specifies how to validate certificates against certificate revocation lists (CRLs) and allows validation based on your definition. An Authentication definition configures multifactor authentication using SSL client certificates, SSH keys, user IDs and passwords, client IP addresses, and RSA SecurID as factors. It can also include application outputs to map attributes, such as return user ID and password information, to outputs you specify.

- Certificate Validation
- User Authentication and Authorization Steps

Certificate Validation

The following illustrates how a client, EA, and directories in the LDAP server work together:

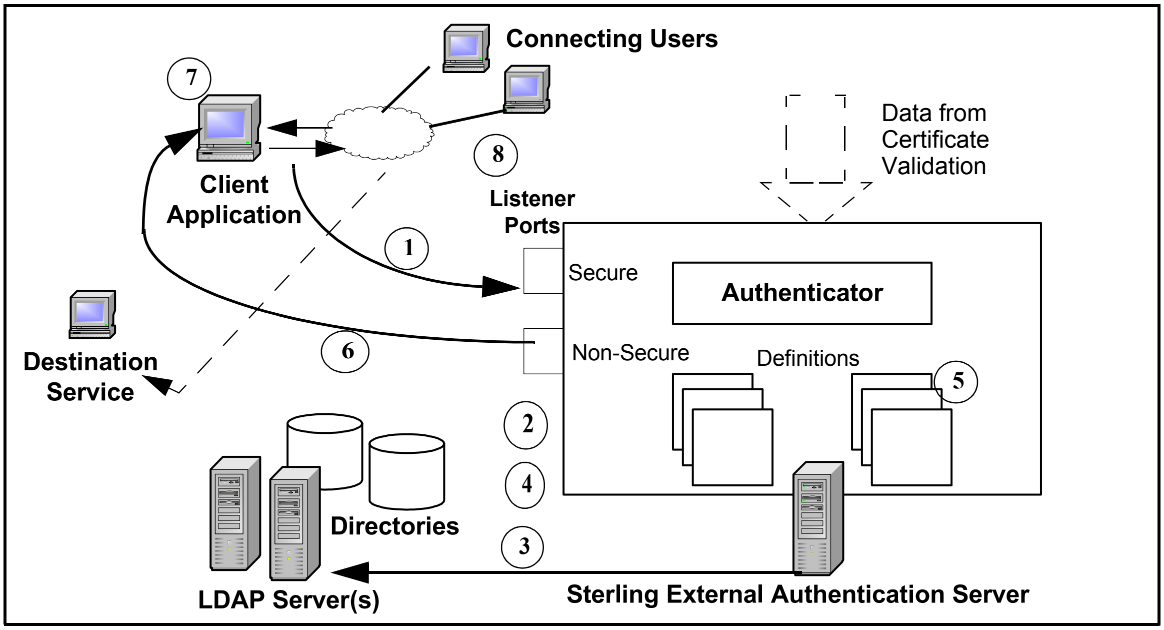


Following is how EA, the client, and LDAP interact when processing a CV definition:

Step	Description
1	A client sends a request and a certificate chain to EA. The client authenticates itself and specifies the certificate validation (CV) definition to use. If the connection is made, mutual authentication and encryption secure messages through the connection.
2	The CV definition performs validation steps. The definition can include LDAP connection information, certificate revocation list (CRL) data, attribute queries, and attribute assertions.
3	For definitions that include attribute queries and CRL definitions, EA connects to the LDAP server to download CRLs, verify certificate subject and group entries, or perform attribute queries.
4	EA verifies query results, end-user certificate information, and data specified in the CV definition.
5	EA notifies the client of the success or failure of a CV.

User Authentication and Authorization Steps

The following diagram illustrates the interaction between a client, EA, and directories accessed through LDAP servers. End users connect securely to the client that connects to EA.



Following is an explanation of the steps in the diagram:

1. A client sends a user ID and password to EA, authenticates itself to EA and specifies the definitions for authentication. If the connection is made, authentication and encryption secure messages through the connection.
2. EA uses the authentication definition to identify LDAP connection information, attribute queries and assertions, and how to authenticate or authorize the connection.
3. EA connects to the LDAP server and validates the user ID and password. It performs the tasks in the definitions, such as attribute queries and assertions. Attribute queries find a user ID entry, validate group membership, and look up login credentials.

4. EA determines if the user is authenticated or allowed access to a destination service. A user authenticated after a certificate validation can use the certificate validation for authentication.
5. An authentication definition can include application output to identify how query map attributes are sent to the client.
6. EA sends the client the results of the user authentication. If it is successful, the response can include credentials. For example, EA provides the user ID and password from a query in an application output definition as part of the response message.
7. A proxy application logs in to the destination service with the credentials obtained by EA.

Interaction with Sterling Secure Proxy (SSP) and Connect:Direct Secure+ Option

EA enhances the security of Sterling Secure Proxy (SSP) and Connect:Direct Secure+ Option (Secure+). For example, EA extracts data from the certificate chain and uses the CV definition to connect to an LDAP server. It can validate the certificate subject and determine that a digital certificate is not revoked. When EA receives an authentication or authorization request, it can determine that a user ID and password have access to a destination service or application.

The EA server processes requests from SSP to validate certificates and authenticate users, in LDAP or Active Directory and validates certificates for Secure+ client requests. The server accepts requests on a secure and nonsecure listener port. All SSL/TLS requests connect through the secure listener port. The nonsecure port is used for testing, or when the connection originates from a client in the trusted zone. The nonsecure port can be disabled after the secure port is set up.

In a typical scenario, SSP establishes a secure session to EA and validates the external client connecting to a Web application, destination service, or Connect:Direct node. Based on the authentication definitions configured in EA, the server initiates a secure connection to an LDAP server and queries the directory to verify the following information for the proxy connection:

- The certificate presented belongs to an organization listed in the LDAP directory, has not expired or been revoked, and has a valid signature.
- The certificate contains specific X.509 v3 extensions.
- The key meets minimum length requirements.
- The password and login ID of the user match the UID attribute specified on the LDAP server.
- Attribute queries or assertions in the definition can be validated from the LDAP directory.
- The IP address of the connection is valid for the proxy connection and the client connection to SSP is from an IP address that is valid for the organization.

If EA confirms that the credentials are valid, SSP completes the connection request; otherwise, the connection fails.

Secure+ can initiate a connection to EA to request extended CV functions. The Connect:Direct PNODE or SNODE negotiating the session can initiate a direct connection to EA, if EA is defined as a remote node in the Secure+ parameters file. See the *Connect:Direct Secure+ Option Implementation Guide* for your platform for instructions.

File Naming Guidelines

Security Terms

File Naming Guidelines

Definition names can be up to 255 alphanumeric characters, and include space, underscore (_), and period (.). They cannot begin or end with a space. EA discards names that begin or end with a space. Following are valid uses of special characters in definition names: Routing Names, corpnet.ldap.home_server234, Cert_Subject_Romuli8Query.

Security Terms

Following are security terms used in EA:

- Self-signed certificate—Digital document is signed and authenticated by its owner. Its authenticity is not validated by the digital signature and trusted key of a third-party certificate authority (CA). To use self-signed certificates, exchange certificates with all trading partners.
- Simple authentication—Authentication by sending the fully-qualified DN and clear-text password of the user. Simple authentication can be used on an encrypted channel.
- CA-Signed certificate—Digital document from a certificate authority (CA) binds a public key to the identity of the certificate owner for authentication. An identity certificate issued by a CA is digitally signed with its private key.
- Certificate Authority (CA)—An organization that issues digitally-signed X.509 certificates. The CA authenticates the certificate owner identity and services they are authorized to use, issues new certificates, renews certificates, and revokes certificates that are no longer authorized. The CA digital signature is assurance that anybody who trusts the certificate signed by the CA can also trust that the certificate is a representation of the certificate owner.
- Certificate Signing Request (CSR)—Message sent from an applicant to a CA to apply for a CA-signed certificate. Before creating a CSR, you first generate a key pair, keeping the private key secret. The CSR contains information that identifies the applicant (such as a distinguished name in the case of an X.509 certificate) and the public key chosen by the applicant.
- Certificate chain—An ordered list of certificates containing an end-user subscriber certificate and issuing authority certificates. EA uses each certificate in the chain to identify the subsequent certificate and checks the trust store for any missing certificates.
- Certificate Revocation List (CRL)—List of certificates that have been suspended or revoked before the expiration date. A CRL defines the CRL issuers name, date of issue, when the CRL will be reissued, serial numbers of revoked or suspended certificates, and times and reasons certificates were revoked or suspended.
- Distinguished Name (DN)—Unique name for a directory entry that includes the list of names of parent entries back to the root for the directory.
- Public key—Public part of a complementary public-private key pair. The asymmetric cipher of the public key encrypts data for the session key that is exchanged between server and client during negotiation for an SSL/TLS session. In EA, public keys are always associated with a subject name in the form of a certificate in a Java key store.
- Private key—Private part of a complementary public-private key pair. The asymmetric cipher of the private key is used to decrypt data that is encrypted with its public key. Data that is encrypted with a public key can only be decrypted using its private key. The private key is never transmitted. In EA, a public-private key pair is always created directly into a Java key store; the private key never leaves the key store.

- Session key—Asymmetric cipher used by the client and server to encrypt data. It is generated by the SSL software.
- Trusted root key—Digitally signed public key of the CA, used to validate the public key received during a SSL/TLS session is signed by the CA to verify the identity of the client.
- Keystore—File that contains the private keys and matching key certificates EA uses for SSL and TLS sessions. Each key/certificate pair in the keystore has an associated alias. The secure listener and connection definitions use the alias to reference the key/certificate.
- Trust store—The trust store includes the following digital certificates:
  - Trusted CA or self-signed certificates of the client that EA communicates with over the secure listener
  - Trusted CA or self-signed certificates of the secure servers EA communicates with, including HTTPS, LDAPS, and LDAP v3 Start TLS
  - Trusted CA certificates needed by the Certificate Validation Service when validating requests from client applications
- Principal—Name of user or service to authenticate the client or the name used to authenticate another server. For example, EA uses a principal to search for a user in an LDAP server.
- LDAP—Lightweight Directory Access Protocol. Open industry standard defines rules for messages used by clients and servers. It locally or remotely accesses and updates information in a directory. EA can be a client of an LDAP directory to perform actions such as authenticate user credentials, validate a certificate subject, or check the certificate for revocation.
- Certificate Validation (CV) Definitions and Authentication Definitions—Process requests to validate certificates and authenticate or authorize users. These files define how EA validates certificates and define how EA authenticates users and verifies authorization to access an application or destination service.

  Requests from clients reference the name of a CV or authentication definition. For example, SSP uses a profile name that must match the name of the CV definition; because Proxy_User_CertVal is the name of the profile in SSP, EA uses the CV definition Proxy_User_CertVal to process the CV request from SSP. An EA definition can process requests from multiple client applications.

Elements of CV Definitions

A certificate validation (CV) definition specifies how to validate a certificate from a user. It can include the following optional elements:

- Attribute query definition—Locates directory entries and returns attributes from the entries. The search must succeed for certificate validation to succeed.
- Attribute assertion definition—Statement that must be true for the validation to succeed. Attribute assertions allow you to specify conditions and compare details from the request, such as an IP address, to fixed data or to attributes returned from queries.
- Custom exit—Details for exiting from an EA or authentication definition to perform related tasks using a Java class running an operating system command.
- Certificate revocation list (CRL) definition—How to access the list of certificates that have been suspended or revoked before the scheduled expiration date. CRLs can be referenced and checked during certificate validation. A CRL defines the CRL issuer's name, when it was issued, when the CRL is scheduled for reissue, serial numbers of revoked or suspended certificates, and how often and why certificates are revoked or suspended. If a certificate is

found on a CRL, validation fails. Certificate revocation list definitions can be created independent of the CV definition and referenced in multiple CV definitions.

- Supported extensions—Processing instructions for the X.509 v3 extensions supported for EA.
- Custom extensions—Registers processing instructions for X.509 v3 extensions unknown to EA.

Authentication Definitions

An authentication definition specifies how EA authenticates a destination service user and how to use attributes. It specifies a user ID and password to use to authenticate and authorize the user. An authentication definition can include the following optional elements:

- Attribute query—An LDAP search to locate directory entries and return attributes from the entries. The attributes must be returned for authentication to succeed. Create the query by specifying parameters in a Uniform Resource Locator (URL) or on the Query Parameters screen.
- Attribute assertion—A statement that must evaluate as true for authentication to succeed. They allow additional conditions and compare details from the request, such as a user ID or destination service, to fixed data or to attributes returned from queries.
- Applications outputs—Uses a directory object to map the query return attributes to an output name known by the client. This looks up login credentials to log in to the destination service.
- Custom exit—How to exit a EA generic authentication definition to perform tasks using a Java class or a script or program executed by running an operating system command.

An authentication definition authenticates users by accessing an LDAP server, a Tivoli Access Manager authorization server, or a generic authentication configuration you customize. Within an authentication definition you can create any of the optional elements. See *Create LDAP Authentication Definitions* on page 141, *Create Generic Authentication Definitions* on page 137, or *Create Tivoli Access Manager (TAM) Authentication Definitions* on page 149.

Prerequisite Tasks for Establishing Secure Connections

Before you can establish secure communications sessions using SSL or TLS, your security administrator should determine whether the security policy requires the use of self-signed certificates, CA-issued certificates, or a both.

Refer to the following lists for a summary of the tasks related to using self-signed and CA-issued X.509 digital certificates. See *Create and Manage System Certificates* on page 45, for instructions on generating and storing certificates using the keytool utility.

Tasks Required to Use CA-Issued Certificates

To use certificates issued by a certificate authority, you must complete the following tasks:

- Generate your public-private key pair directly into the keystore of EA or the GUI.
- Generate the certificate signing request (CSR), which contains your public key, and submit it to your certificate authority (CA) for authentication.
- Import the certificate issued by the CA into the server or GUI keystore.
- Provide the digitally signed public key of the CA to your communication peers.

- Import the CA root certificate, or import copies of the X.509 digital certificates containing the public key and digital signature of all the entities that EA communicates with as server and as client in the server trust store, if you are using self-signed certificates.
- To establish a secure connection to EA from the GUI when it is running on a remote computer, complete the procedures listed here to use CA certificates for both the GUI and the server.

Tasks Required to Use Self-Signed Certificates

To use self-signed certificates, complete the following tasks:

- Generate your public-private key pair directly into the keystore of EA or the GUI.
- Export the EA self-signed certificate to a file and distribute a copy to all entities that EA communicates with as server and as client.
- Store copies of the X.509 certificates of the entities that EA communicates with as server and as client in the server trust store, or import the CA root certificate.
- To establish a secure connection to EA from the GUI when it is running on a remote computer, complete all the procedures listed here for using self-signed certificates for the GUI and the server.

System Requirements

EA (EA) version 2.3.00 has the following hardware and software requirements.

Component or Functionality	Hardware	Software	RAM	Disk
EA	Windows compatible systems	Microsoft Windows 2003 Server	512 MB	200 MB
	HP 9000 Platform	HP-UX versions 11.11 and 11.23	512 MB	200 MB
	IBM RISC System/6000 platform	AIX versions 5.3	512 MB	200 MB
	SUN SPARC systems	Solaris versions 9 and 10	512 MB	200 MB
	Intel Pentium system	Red Hat Advanced Server versions 4.0 and 5.0 SuSE SLES versions 9 and 10 Solaris versions 9 and 10	512 MB	200 MB
		◆ Open LDAP versions 2.2 and 2.3 ◆ Sun Microsystems SunONE 5.2 ◆ IBM Tivoli 6.x ◆ Microsoft Windows 2003 Domain Functional Level Active Directory ◆ Active Directory 2008		
EA GUI		Use one of the following: ◆ Internet browser using Java WebStart ◆ JRE version 1.6, installed with EA	256 MB	
Authentication using Tivoli Access Manager		◆ Red Hat Advanced Server 4.0 ◆ Tivoli Access Manager 5.1 ◆ IBM Access Manager Runtime for Java ◆ JRE version 1.4.2	30 MB per TAM authentication definition	
VMware ESX and VMware vSphere		Any native operating system supported by EA. Consider VMware-specific configuration, administration, and tuning issues. Your VMware administrator must address any issues. Sterling Commerce does not provide advice regarding VMware-specific issues.		

What’s New in This Release

EA version 2.3.00 has the following features and enhancements:

Version	Feature or Enhancement
Version 2.3	Adds single-sign on (SSO) support for the SFTP, FTP, and Connect:Direct protocols.
	Allows trading partners to manage their passwords with a self-service mechanism. This change password functionality supports the recommended SSP, EA, and SSO configuration and does not use a third party external portal to manage passwords.
	Provides support for the RSA SecurID token.
	Improved startup supports the ability to run EA as a background process without requiring that the passphrase be saved to disk.
	The stopSeas script provides another secure shutdown method for the EA server.

Support Requests Resolved for This Release

No support requests are resolved for EA version 2.3.00 since the last cumulative fix release. For the history of issues resolved prior to this release, navigate to the Product Updates & Downloads site for your product and platform using the instructions in *Obtaining Product Updates* in the Release Notes PDF and review the Fix List.

Special Considerations

Refer to the following notes before installing the product.

Configuration Considerations

Refer to the following notes when configuring EA:

- The System Settings dialog box, which allows you to configure the listeners, SSL keystore, and trusted certificates, uses a separate object for each tab. When you click **OK**, the objects are updated in the following order: Listeners, SSL Keystore, and Trusted Certificates.
- EA uses strong, but limited, cryptography. To use stronger encryption, replace the default jurisdiction policy files with the Unlimited Strength Jurisdiction Policy Files 5.0, available from the JCE provider.

Jurisdiction Policy File Use

TLS and SSL protocols are implemented in EA, both server and GUI components, using the standard Java™ 5.0 API, Java™ Secure Socket Extension (JSSE) and default provider package.

JSSE, in turn, utilizes the standard Java™ 5.0 API, Java™ Cryptography Extension (JCE) to implement the underlying crypto algorithms.

The cipher suites available for use in SSL and TLS connections are determined by the following JCE jurisdiction policy files shipped with EA:

- *install_dir*/jre/lib/security/local_policy.jar
- *install_dir*/jre/lib/security/US_export_policy.jar

where *install_dir* is the directory where EA is installed.

The jurisdiction policy files shipped with EA enable strong, but limited, cryptography. To use stronger encryption, US customers and eligible countries can replace the default jurisdiction policy files with the Unlimited Strength Jurisdiction Policy Files 5.0, available from the JCE provider.

To replace the default jurisdiction policy files:

1. Access the Sun Web site and select **Java 2 Standard Edition** from the **Download** menu.
2. Scroll to **Other Downloads** and click the link for Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0.
3. Click **Download**.
4. Copy the unlimited strength jurisdiction policy files to the following locations:
    - *install_dir*/jre/lib/security/local_policy.jar
    - *install_dir*/jre/lib/security/US_export_policy.jar

    where *install_dir* is the EA installation directory

Following are the cipher suites enabled by default and by the unlimited jurisdiction policy files:

Default SSL/TLS Cipher Suites	Cipher Suites Enabled by Unlimited Strength Jurisdiction Policy Files
SSL_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_SHA	SSL_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA SSL_ RSA_WITH_DES_CBC_SHA	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA SSL_ DHE_DES_WITH_DES_CBC_SHA	TLS_DHE_DSS_WITH_AES_256_CBC_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_DES_CBC_SHA

Default SSL/TLS Cipher Suites	Cipher Suites Enabled by Unlimited Strength Jurisdiction Policy Files
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	SSL_DHE_RSA_WITH_DES_CBC_SHA SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA SSL_RSA_WITH_NULL_MD5	SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_WITH_NULL_SHA SSL_DH_anon_WITH_RC4_128_MD5	SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA	SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA SSL_RSA_WITH_NULL_MD5
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA SSL_DH_anon_WITH_DES_CBC_SHA	SSL_RSA_WITH_NULL_SHA SSL_DH_anon_WITH_RC4_128_MD5
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	TLS_DH_anon_WITH_AES_128_CBC_SHA TLS_DH_anon_WITH_AES_256_CBC_SHA
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA TLS_KRB5_WITH_RC4_128_SHA	SSL_DH_anon_WITH_3DES_EDE_CBC_SHA SSL_DH_anon_WITH_DES_CBC_SHA
TLS_KRB5_WITH_RC4_128_MD5 TLS_KRB5_WITH_3DES_EDE_CBC_SHA	SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_KRB5_WITH_3DES_EDE_CBC_MD5 TLS_KRB5_WITH_DES_CBC_SHA	SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA TLS_KRB5_WITH_RC4_128_SHA
TLS_KRB5_WITH_DES_CBC_MD5 TLS_KRB5_EXPORT_WITH_RC4_40_SHA	TLS_KRB5_WITH_RC4_128_MD5 TLS_KRB5_WITH_3DES_EDE_CBC_SHA
TLS_KRB5_EXPORT_WITH_RC4_40_MD5	TLS_KRB5_WITH_3DES_EDE_CBC_MD5 TLS_KRB5_WITH_DES_CBC_SHA
TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA	TLS_KRB5_WITH_DES_CBC_MD5 TLS_KRB5_EXPORT_WITH_RC4_40_SHA
TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5	TLS_KRB5_EXPORT_WITH_RC4_40_MD5
	TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA
	TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5

Known Restrictions

EA version 2.3.00 has the following known restrictions:

- ✦ On an AIX computer, the AES128 and AES256 ciphers do not work with the SSL protocol. To enable these ciphers, use the TLS protocol.
- ✦ When you install two NIC cards for a remote perimeter server and the network interface uses different IP addresses for the two NIC cards, make sure the definition for the associated SSP engine for a perimeter server matches what was defined when the perimeter server is installed.

When configuring client software, use the correct IP address based on the definition for the external network interface.

Caution: Be careful about using host name in the external network interface.Make sure the host name does not identify the IP address specified during the network interface configuration. If it does, use the IP address only.

- Do not attempt to run the cryptotool.sh on UNIX or the cryptotool.cmd on Windows unless instructed to do so by Sterling Commerce support.

Review Resources

Before you install EA, review security configuration details relevant for EA. Refer to the following resources as you plan network and security to install and configure EA:

Configuration Resource	EA Usage
TCP Ports	Use available port numbers, in appropriate port ranges to set the secure and non-secure listener, and servlet port used to download the GUI.
Network Interface Addresses	Confirm the local bind address of a network interface for a connection.
LDAP Directory Information Tree	Apply related knowledge when selecting and specifying LDAP parameters for checking attributes in directory entries.
Requirements for data encryption	Set SSL/TLS-related parameters for connections between the server and GUI, between the EA server and client applications, and between EA and LDAP directory servers.
Ciphers for data encryption	Apply knowledge of cipher selection and related requirements when configuring data encryption parameters.
Authentication mechanism use requirements	Choose the appropriate Simple Authentication and Security Layer (SASL) mechanism from those supported in authentication definitions.
Use of self-signed certificates	Allow self-signed certificate use as appropriate.
Use of certificates signed by Certificate Authorities (CAs)	Support use of certificates signed by selected CAs.
Length of public keys	Set the public key minimum length in certificate validation definitions.

Install EA on UNIX or Linux

During installation, write down the passphrase you define. You may need it when you start EA.

To install EA on UNIX:

1. Navigate to the installation directory.
2. Navigate to the directory for your platform, as identified in the following table:

Platform	Directory
HP PA-RISC	HP-UX
IBM System p and IBM Power Systems	AIX
Sun SPARC	SolarisSPARC
Intel x86 Linux	LinuxINTEL
Intel x86 Solaris	SolarisINTEL

3. To start the installation, type the following command.

```
sh SEASInstall.bin
```

4. Accept the default installation directory or specify a different directory and press **Enter**.
5. Accept the nonsecure listener or specify a different port and press **Enter**. Default=61365.
6. Type a passphrase, six or more characters, and press **Enter**. Write it down.
7. To configure the servlet container:
   a. Accept the default value for the port number or specify a value.
   b. Accept the default or specify a value for the fully-qualified DNS name for the engine.
8. Review the installation and press **Enter**. When the installation is complete, the command prompt is displayed.

Start EA on UNIX or Linux

Use the following checklist to ensure that you complete the tasks necessary to start EA:

Installation Task	Procedure to Complete
Start the EA server	Use one of the following procedures: ◆ *Start the EA Server on UNIX Using a Stored Passphrase* on page 26 ◆ *Start the EA Server on UNIX and Require a Passphrase* on page 26

Installation Task	Procedure to Complete
Start the EA GUI	Use one of the following procedures: ◆ *Start the EA GUI From the Computer Where the EA GUI Is Installed* on page 27 ◆ *Start the EA GUI from a Remote Computer* on page 28
Change admin password	*Change the Admin Password* on page 41
Log Off	*Log Off EA on UNIX or Linux* on page 29
Shut down EA	*Shut Down EA on UNIX* on page 29

Start the EA Server on UNIX or Linux

Use one of the following methods to start EA:

Start EA automatically, without interaction from the user. The passphrase is read from an encrypted file.

Start EA and require that the user type a passphrase when prompted. The passphrase is masked and is not visible as the user types the characters.

With both methods, the server starts in the background. All log messages are sent to the bin/startSeas.out file. Determine how to start EA and complete the procedure for that method.

Start the EA Server on UNIX Using a Stored Passphrase

This is the default startup method. It does not require that a user type a passphrase at startup because it is stored in an encrypted file. The server starts in the background without user interaction.

Note: If this is an upgrade and the passphrase file does not exist in the previous installation, it will not be created during the upgrade.

To start the EA server using a stored passphrase:

1. Navigate to *install_dir*/bin, where *install_dir* is the EA installation. Type the following:

```
./startSeas.sh
```

2. Check the status of the server startup by viewing the bin/startSeas.out file. If the startup is successful, the file contains the following message:

```
   Sterling External Authentication Server is ready for Service.
```

Start the EA Server on UNIX and Require a Passphrase

This method requires that you type a passphrase. When entered, it is masked and not visible.

To start EA and require that a passphrase be provided at startup:

1. Delete the sb.enc file from the *install_dir*/conf/system directory.

2. Navigate to the *install_dir*/bin directory and type the following command:

```
./startSeas.sh
```

3. Type the passphrase and press **Enter**.
4. Check the status of the startup by viewing the bin/startSeas.out.out file. If the successful, the file contains the following message:

```
Sterling External Authentication Server is ready for Service
```

Restore the Stored Password File on UNIX

If you use the method, *Start the EA Server on UNIX and Require a Passphrase* on page 26, to start the EA server, it deletes the stored passphrase and requires that the user type a passphrase at startup.

To restore the default start up method, restore the saved passphrase. This procedure restores the passphrase to the conf/system/sb.enc file.

To restore the stored password file on UNIX:

1. From the *install_dir*/bin directory, type the following command:

```
enableBootstrap.sh
```

2. At the prompt, type the passphrase defined for EA and press **Enter.**

Complete the procedure *Start the EA Server on UNIX Using a Stored Passphrase* on page 26 to start EA, using the stored passphrase.

Start the EA GUI

Select the platform where you want to start the EA GUI:

Start the EA GUI On UNIX on page 27

Start the EA GUI on Windows on page 33

Start the EA GUI On UNIX

When you start the GUI and connect to the server the first time, you use the nonsecure port. To connect to the server using the secure listener port, set up the certificates on the server and on the GUI and enable the secure listener port in the server. Refer to *Create and Manage System Certificates* on page 45. Start the GUI from the computer where it is installed or from a remote connection.

Start the EA GUI From the Computer Where the EA GUI Is Installed

To start the GUI on the computer where EA is installed, X Windows must be running.

To start the EA GUI:

1. Navigate to the *install_dir*/bin directory. Type the following command:

```
./startGUI.sh
```

2. On the Login screen, provide the following information:
   - Host
   - Port
   - User
   - Password

Note: The default user is **admin** and password is **admin**. Use them the first time you logon. Then, change the password.

3. Click **Login**.

Start the EA GUI from a Remote Computer

You can run the EA GUI on any remote computer that can connect to the EA server.

To run the EA GUI from a remote computer:

1. Open an Internet browser.
2. In the **Address** field, type **http://*SEAS_host*:*port***, where *SEAS_host* is the host name of the EA server, and *port* is the port for the servlet container, defined at installation. Default=9080.
3. Click **Launch GUI**. The first time you run EA from a browser, you receive messages about the launch and any security issues.
4. Accept the certificate to start the GUI from the browser for the first time.
5. Provide the following information:
   - Host
   - Port
   - User
   - Password
   - SSL/TLS

Note: The default user is **admin** and the password is **admin**. Use these values the first time you logon. Then, change the password.

6. Click **Login**.

Log Off EA on UNIX or Linux

To log off of EA, select **Exit** from the **File** menu.

Shut Down EA on UNIX

If you close the EA GUI, the EA server continues to run. Keep the EA server running when client applications need to connect. To shut down the server, close all open GUI windows and complete the one of the following procedures.

To shut down the EA server from the EA GUI:

1. From the **File** menu, select **Shutdown Server**.
2. Type the passphrase created at installation and click **OK.**

To shut down the EA server from a command line:

1. From a command prompt, navigate to the *install_dir*/bin directory.
2. Type the following command.

```
./stopSeas.sh
```

3. When prompted, type the passphrase, defined at installation.
4. When prompted, type the administrator ID and password.

   A message is displayed indicating the server is shutting down.

Install EA on Windows

At installation, you define a six or more character passphrase that contains any combination of characters. Write it down because you may need it when you start the server.

To install EA on Windows:

1. Navigate to the directory where the EA file is downloaded and navigate to the Windows folder to locate the installation file.
2. Double-click the SEASInstall.exe file.
3. Read the introductory information and click **Next**.
4. Accept the installation directory or click **Choose** to select another directory. Click **Next**.
5. Accept the default for the listener or specify a different port. Click **Next**. Default=61365.
6. Type a passphrase, in the **Passphrase** and **re-enter passphrase** fields. Click **Next**.
7. To configure the servlet container:
   a. Accept the default value for the port of the servlet container or specify a value.
   b. Accept the default for the fully-qualified DNS name for the engine or specify a value.
   c. Click **Next**.
8. Review the installation details and click **Install**.
9. Click **Done**.

Start EA on Windows

Use the following checklist to ensure that you complete tasks to start EA.

Installation Task	Procedure to Complete
Start the EA server	Use one of the following procedures: ◆ *Start the EA Server on Windows Using an Encrypted Password* on page 32 ◆ *Start the EA Server on Windows And Require a Passphrase* on page 32
Start the EA GUI	Use one of the following procedures: ◆ *Start the GUI from the Local Windows Computer* on page 33 ◆ *Start the GUI from a Remote Computer* on page 33
Change the password for the admin user	*Change the Admin Password* on page 41

Start the EA Server on Windows

When you install the EA server, you define a passphrase. It is required to start the server. Start EA server as a Windows service or require that a passphrase be typed at startup.

Start the EA Server on Windows Using an Encrypted Password

This startup method is enabled when you install EA. The user is not required to type a passphrase at startup because it is stored in a file. The server starts in the background, as a Windows service without user interaction.

To start the EA server using a stored passphrase:

1. From **Control Panel**, double-click **Administrative Tools**.
2. Double-click **Services**.
3. Double-click the **Sterling External Authentication Server V2.3.00** service.
4. To configure the service to start automatically every time the computer is started, set **Startup type** to **Automatic**.
5. Under **Service status**, click **Start**.

Start the EA Server on Windows And Require a Passphrase

To start the EA server and require that a passphrase be provided:

1. Delete the sb.enc file from the *install_dir*/conf/system directory, where *install_dir* is the directory where EA is installed.
2. From a command prompt, navigate to *install_dir*/bin and type the following command:

```
startSeas.bat
```

3. When prompted, type the passphrase defined at installation.The following message is displayed when the startup is successful.

```
The Sterling External Authentication Server V2.3.00 service was started successfully.
```

The EA server runs as a Windows Service when the startup is complete.

Restore the Stored Password File on Windows

If you use the method, *Start the EA Server on Windows And Require a Passphrase* on page 32, it deletes the stored passphrase and requires that the user type a passphrase at startup.

To restore the default start up method, you must restore the saved passphrase.

To restore the passphrase to the conf\system\sb.enc file:

1. From the *install_dir*/bin directory, type the following command:

```
enableBootstrap.bat
```

2. At the prompt, type the passphrase defined for EA and press **Enter.**

Start the EA GUI on Windows

When you start the GUI the first time, you must use the nonsecure port. To prepare for secure connection, set up certificates on the server and GUI and enable the secure listener port. See *Create and Manage System Certificates* on page 45.

You can logon from the computer where EA is running or from a remote computer.

Start the GUI from the Local Windows Computer

To start the GUI on the computer where EA is running:

1. From the **Start** menu, click **Programs** > **Sterling External Authentication Server V2.4.00**>**Sterling External Authentication GUI**.
2. Provide the following information:
   - Host
   - Port
   - User
   - Password
   - SSL/TLS

   Note: The default user is **admin** and the default password is **admin**. Use the default values the first time you logon. Then, change the password to maintain security.

3. Click **Login**.

Start the GUI from a Remote Computer

You can download and run the GUI on any remote computer that can connect to the EA server.

To start the GUI on a remote computer:

1. Open an Internet browser.
2. In the **Address** field, type **http://*SEAS_host*:*port***, where *SEAS_host* is the EA server name, and *port* is the port for the servlet container, specified during installation. Default=9080.
3. Click **Launch GUI**. The first time you run EA from a browser, messages are displayed about the launch and any potential security issues.
4. Accept the certificate to start the GUI from the browser for the first time.
5. Provide the following information:
   - Host
   - Port
   - User
   - Password
   - SSL/TLS

Note: The default user is **admin** and the default password is **admin**. Use the default values the first time you logon. Then, change the password to maintain security.

6. Click **Login**.

Log Off EA on Windows

To log off EA, select **Exit** from the **File** menu.

Shut Down EA on Windows

If you close the EA GUI, the server continues to run. Keep the server running when applications need to connect.

Shutdown the EA Server from the EA GUI

To shut down the EA server from the EA GUI:

1. Select **File**>**Shutdown Server**.
2. Type the passphrase created at installation and click **OK**. The EA server shuts down and the GUI closes.

Shutdown the EA Server from a Command Line

To shut down the EA server from a command line:

1. From a Windows command prompt, navigate to the *install_dir*/bin directory.
2. Type the following command.
3. When prompted, type the passphrase, defined at installation.
4. When prompted, type the administrator ID and password.

   A message indicates the server is shutting down.

Upgrade EA

If you upgrade an installation, configuration files located in the conf directory and log files located in the logs directory are not overwritten. Configuration files that are new to version 2.3.00 are installed and encrypted with a passphrase. If you removed any files from an installation, such as removing the sb.enc file to require that a passphrase be provided at startup, these files will not be replaced during an upgrade.

To upgrade Sterling External Authentication Server to version 2.4:

1. Shut down the EA server and confirm that no application is accessing EA files.
2. Make a backup of the existing installation, to use only if the upgrade is unsuccessful.
3. Install Sterling External Authentication Server version 2.4.00.
4. Specify the directory where the existing version is installed.

   The installation program detects the existing installation and gives you the opportunity to install over the existing files or specify an alternate directory.
5. If the file conf/system/sb.enc does not exist in the existing installation, you are prompted for a passphrase. Specify the passphrase from the original installation.

   The original port of the non-secure listener and the Servlet container from the original installation are used.

   After you review information displayed in the Pre-Installation Summary, the upgrade updates any new and modified files.

   When the upgrade is complete, you may start the EA server.

Configure System Resources

After you install EA, use the procedures in this section to configure system resources. Define a system-wide connection to define the connection requirements. A connection requirement can be used with more than one definition, if the connection requirements are the same.

Organization of the LDAP Connection Configuration

The configuration instructs you how to create a basic setup. After you complete a scenario, test the connection to ensure it is correctly configured. Determine which features apply to your environment and configure only those.

How to Use the Worksheets

Before you configure a definition, gather the information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

After you complete a definition, test it to make sure EA can connect to SSP and the outbound server.

Prerequisites

No prerequisites are required before you configure an LDAP Connection definition.

System-Wide LDAP Connection Definition

Create a system-wide connection definition to associate with a CV, CRL, or Authentication definition. Creating a system-wide connection definition before you create the authentication definition saves time when the same connection is required for multiple uses. A system-wide connection definition ensures that connection details are made in one place and login credentials are entered only once.

Complete the LDAP connection worksheet to identify the information needed to configure a connection. Then, complete the procedure.

System-Wide LDAP Connection Worksheet

Before you configure the system-wide connection, gather the following information:

Field	Description	Value
Name	Name of the system-wide connection.	
Protocol	Protocol to use. For clear text connection, select ldap://. For SSL, select ldaps://.	
Host	Host name of the LDAP server.	
Port	Port number to use to connect to the LDAP server.	

Field	Description	Value
Authentication Method	Method to authenticate the principal against the directory: ◆ Simple—The password ◆ Digest-MD5—Transmits message digests over the network. ◆ CRAM-MD5—The CRAM-MD5 encrypted password. This method transmits message digests. ◆ GSSAPI—Use Kerberos V authentication. This is the native authentication used in Active Directory.	
Principal Name	Security principal used in the bind operation to LDAP. The name is displayed as a formula with details filled in based on a previous selection.	
Principal Password	Password for the security principal, obtained from the authentication request.	
Client Key Certificate Alias	Key certificate with LDAP when client authentication is enabled. Disabled unless the protocol is ldaps://, or Start TLS option=Yes.	
LDAP Version	LDAP protocol version. Select one of the following values from the list: ◆ 2—Use LDAP version 2. ◆ 3—Use LDAP version 3.	
Start TLS	Whether to request TLS encryption using LDAP v3.	
Referral Action	What to do when a request is referred by an LDAP server to another. ◆ Follow—Follow the referral to the referred directory. ◆ Ignore—Ignore the referral. ◆ Throw—Ignore the referral and generate an exception.	
Advanced options	Property names and values if the JNDI service provider requires special properties. Click to specify properties.	

Create a System-Wide LDAP Connection

To create a connection definition:

1. From the **Manage** menu, click **System Settings** and click the **Connection Definitions** tab.
2. Click + to add a new connection definition. Name the connection definition.
3. In the **Protocol** field, specify ldap:// to connect using the Lightweight Directory Access Protocol or ldaps:// to connect using the SSL/TLS.
4. Specify the connection parameters. Use the values you identified in the worksheet.
5. Click **Next** and click **Save**.
6. Click **Close**.

Define a System-Wide HTTP Connection Definition

Create a system-wide connection definition to perform attribute queries or download certificate revocation lists. When you create a CV definition, CRL definition, or AD definition in EA, you can then select a system-wide definition.

Creating system-wide connection definitions before you create the authentication definition saves time when the same connection is required for multiple uses. A system-wide connection definition ensures that connection details are made in one place and login credentials are entered only once.

Complete the HTTP connection worksheet to identify the information needed to configure a connection. Then, complete the procedure.

System-Wide HTTP Connection Worksheet

Before you configure the system-wide connection, gather the following information.

Field	Description	Value
Name	Name of the system-wide connection.	
Protocol	Protocol to use. For clear text connection, select ldap://. For SSL, select ldaps://.	
Host	Host name of the LDAP server.	
Port	Port number to connect to the LDAP server.	
Client Key Certificate Alias	The key certificate during SSL or TLS with the LDAP server when client authentication is enabled. Disabled unless the protocol is ldaps://, or the Start TLS option is set to Yes.	
Advanced options	Click button to define the local bind address of the JNDI service provider.	

To create a connection definition:

1. From the **Manage** menu, click **System Settings** and click the **Connection Definitions** tab.
2. Click + to add a new connection definition. Name the connection definition.
3. In the **Protocol** field, specify either http:// to connect using the HTTP protocol without SSL/TLS or https:// to connect using SSL/TLS.
4. When the protocol is https:// specify a client key certificate alias to identify a certificate to use from the system SSL key store.
5. Specify the connection parameters and click **Next**.
6. Click **Next** and click **Save**.
7. Click **Close**.

Modify System Resources

Use the following procedures to modify the admin password, non-secure listener port, servlet container, and logging options. Procedures are available to refresh the GUI lists, set listener connection settings, and configure the Kerberos API.

Configure a Listener Port

Modify the Non-Secure Listener Port

Disable the Non-Secure Listener Port

Change the Port Number of the Servlet Container

Change the Admin Password

Configure Logging Options

Refresh GUI Lists from the Server

Set Listener Connection Settings (Backlog and Timeout)

Configure the Kerberos API

Configure a Listener Port

Modify the Non-Secure Listener Port

Configure the Secure Connection Listener

Modify the Non-Secure Listener Port

The non-secure listener defines how a client connects to EA when SSL or TLS in not required. The non-secure listener port is configured during installation.

To change the non-secure listener port:

1. From the **Manage** menu, select **System Settings**.
2. Click the **Listeners** tab.
3. In the Non-Secure Listener section, specify the IP address and port.
4. To enable the port, click Enabled and click **OK**.

Disable the Non-Secure Listener Port

After you set up the secure listener port, you can disable the non-secure listener.

To disable the non-secure listener port:

1. From the **Manage** menu, select **System Settings**.
2. From the **Listener** tab, disable the Enabled box.
3. Click **OK**.

Change the Port Number of the Servlet Container

You may require a different port number for the servlet container.

To change the port number specified for the servlet container:

1. Open the XML document file, *install_dir*/conf/jetty/JettyConfigDef.xml.
2. Locate the XML tag: <port>*servlet_port*</port>, where *servlet_port* is the servlet port you specified during installation, such as 9080.
3. Change *servlet_port* to the new port you want to use for the servlet container.
4. Save the file.
5. Open the Java Network Launching Protocol definition file, *install_dir*/conf/jetty/docroot/webstart/EA_GUI.jnlp.
6. Locate the XML tag, <jnlp spec="0.2 1.0" codebase="http://*host_info*:*servlet_port*/webstart"href="EA_GUI.jnlp">, *host_info* is the name of the host used for the installation.
7. Change *servlet_port* to the new port you want to use for the servlet container.
8. Save the file.

Change the Admin Password

To secure the application after installation, change the admin password.

To change the password for the admin user:

1. From the **Manage** menu, select **Users**.
2. Select the user definition for admin and click 📝.
3. Type the new password in the **Password** and **Confirm Password** fields.
4. Click **OK**.

Configure Logging Options

EA supports multiple levels of logging to capture operational messages reported for certificate validation and authentication definitions. EA logging has the following default configuration:

Logging to the console is disabled for the server and the GUI.

INFO logging level captures errors, warnings, and informational messages.

The installation log called Sterling_External_Authentication_Server_V2.0.00_InstallLog.log file is saved in the *install_dir* directory.

The server log called seas.log is in stored in the *install_dir*/logs file.

The GUI log called seasgui.log is stored in the /*install_dir/*bin directory.

The default maximum log file size allowed before archiving is 1000 KB.

The maximum number of log files kept in the system is 20.

Configure the logging level and logging details by editing the log4j properties file in the *install_dir*/conf directory to change logging for the server. Edit the guilog4j.properties file to change logging for the GUI. You can also change the logging level for the server from the GUI; see *Set Listener Connection Settings (Backlog and Timeout)* on page 43 for information.

Change the Logging Level from the GUI

To change the level of detail captured in the EA log files, using the GUI:

1. From the **Manage** menu, select **System Settings**.
2. Click the **Globals** tab.
3. Select the logging level in the **Logging Level** field and click **OK**.

Turn Logging to the Console On or Off

To turn logging to the console on or off.

Note: Do not enable logging to the console if Tivoli Access Manager (TAM) authentication definitions are used. Logging data conflicts with interprocess communications.

1. Navigate to the *install_dir*/conf directory, where *install_dir* is where EA is installed:
2. Do one of the following:
   - Open the log4j.properties file to modify logging for the server.
   - Open the guilog4j.properties file to change logging for the GUI.
3. Identify the logging output parameters, as illustrated in the following log4j.properties file:

   ```
   #log4j.appender.stdout=org.apache.log4j.ConsoleAppender
   log4j.appender.stdout=org.apache.log4j.varia.NullAppender
   ```

   The first line ending in ConsoleAppender turns on the console output. The line ending in varia.NullAppender suppresses console output.
4. Comment out the logging option you do not want to use by adding the pound symbol (#) at the beginning of the line. By default, output to the console is turned off.

   Note: Either the ConsoleAppender or the NullAppender must be commented out.
5. Save the logging properties file.

Change the Log File Size

To change the maximum size a log file reaches before it is archived:

1. Navigate to the *install_dir*/conf directory, where *install_dir* is the EA installation.
2. Do one of the following:
   - Open the log4j.properties file to modify logging for the server.
   - Open the guilog4j.properties file to change logging for the GUI.
3. Define how large you want the log file to reach before archiving in the MaxFileSize parameter. Default=1000 KB and size is defined in kilobytes.

   ```
   log4j.appender.R.MaxFileSize=1000KB
   ```
4. Save the logging properties file.

Change the Number of Archive Log Files

To change the maximum number of log files to archive:

1. Navigate to the *install_dir*/conf directory, where *install_dir* is where EA is installed.
2. Do one of the following:
   - Open the log4j.properties file to modify logging for the server.
   - Open the guilog4j.properties file to change logging for the GUI.
3. Type how many archive log files to keep in the MaxBackupIndex parameter. Default=20.

```
log4j.appender.R.MaxBackupIndex=20
```

4. Save the logging properties file.

Change the Logging Level in a Logging Properties File

lo change the logging level to determine what server performance details are written in a log:

1. Navigate to the *install_dir*/conf directory.
2. Do one of the following:
   - Open the log4j.properties file to modify logging for the server.
   - Open the guilog4j.properties file to change logging for the GUI.
3. Define the logging level to report in the LEVEL parameter.

```
log4jrootLogger=LEVEL,R,stdout
```

4. Save the logging properties file.

Refresh GUI Lists from the Server

More than one administrator can use the same EA GUI to configure EA. When multiple administrators change EA definitions, you may need to update the windows that list certificate revocation lists, certificate validations, authentications, user definitions, and role definitions.

To refresh lists with updated configuration information from the server:

From the **Manage** menu, click **Refresh Lists**. When the progress message dialog box closes, the GUI lists EA configuration definitions added since the last refresh.

Set Listener Connection Settings (Backlog and Timeout)

Leave the listener connection fields blank to accept the default settings. To change the listener connection settings, specify parameters to control the backlog of connections, set the timeout for accepting an inbound connection, and set the timeout for outbound connections and read operations.

To change listener connection settings:

1. From the **Manage** menu, click **System Settings**.
2. Click the **Globals** tab.

3. To customize the listener settings, set the following connection parameters:
    - Listen Backlog
    - Accept Timeout
    - SSL Handshake Timeout
    - Session Idle Timeout
    - Connect Timeout
    - Read Timeout
4. Click **OK**.

Configure the Kerberos API

EA uses Kerberos to allow users to change a password in Active Directory. Configure Kerberos using EA system settings. In Active Directory, realm names are domain names.

Note: Kerberos cannot be used on AIX.

To configure EA to identify each Kerberos realm:

1. From System Settings, click **Kerberos Configuration**.
2. Click +.
3. Define the realm by providing information in the **Name** and **Kdsc** fields.
4. Click **OK**.

Create and Manage System Certificates

Before you configure SSL or TLS connections, you create, exchange, and store certificates for EA and the its clients. Based on how you deploy EA, you may have to distribute the public key to clients and store certificates from clients, LDAP servers, and end users.

This section explains how to generate and store self-signed and CA-signed certificates for the server, import certificates, configure access to the keystore and trust store, and configure a secure connection between a remote GUI and server. Identify the type of certificate you generate and the connection you secure, and complete the procedures in one of the following sections.

Generate a Self-Signed Certificate to Secure the EA Connection

Complete the following procedures to configure a self-signed certificate to secure the EA server:

Generate a Self-Signed Certificate to Authenticate EA to a Client on page 46

Export a Self-Signed Certificate from the Keystore for the EA Server on page 49

Import a Certificate into the EA Server Trust Store on page 53

Configure EA to Access the SSL Keystore on page 54

Configure EA to Access the SSL Trust Store on page 54

Configure the Secure Connection Listener on page 55

Generate a CA Certificate to Secure the Connection to EA

Complete the following procedures to generate CA certificates for the EA server:

Generate a Self-Signed Certificate to Authenticate EA to a Client on page 46

Create a PKCS#10 Certificate Signing Request for the Server on page 50

Obtain the root certificate from the CA. Distribute it to the GUI, clients, and servers that require client authentication.

Import the CA Certificate into the Server Keystore on page 51

Import a Certificate into the EA Server Trust Store on page 53

Configure EA to Access the SSL Keystore on page 54

Configure EA to Access the SSL Trust Store on page 54

Configure the Secure Connection Listener on page 55

Generate a Self-Signed Certificate for the Connection between the GUI and EA Server

Complete the following procedures to set up self-signed certificates between the server and GUI:

Generate a Self-Signed Certificate to Authenticate EA to a Client on page 46

Generate a Self-Signed Certificate for the GUI on page 47

Export a Self-Signed Certificate for the GUI on page 49

Import a Certificate into the EA Server Trust Store on page 53

Import the Server Certificate into the GUI Trust Store on page 53

Generate a CA Certificate to Secure the Connection between the GUI and EA

Complete the following procedures to configure CA certificates between EA and the GUI:

Generate a Self-Signed Certificate to Authenticate EA to a Client on page 46

Create a PKCS#10 Certificate Signing Request for the Server on page 50

Import the CA Certificate into the Server Keystore on page 51

Generate a Self-Signed Certificate for the GUI on page 47

Create a PKCS#10 CSR for the GUI on page 51

Import the CA Certificate to the GUI Keystore on page 52

Import a Certificate into the EA Server Trust Store on page 53

Import the Server Certificate into the GUI Trust Store on page 53

Generate a Self-Signed Certificate

To configure EA for secure communications, first generate a self-signed certificate. A self-signed certificate is required whether you use a self-signed or CA certificate to secure the connection between EA and clients or between the GUI and EA.

To use a self-signed certificate to authenticate EA to a client, first generate the certificate for EA. Then, export the certificate and send it to the client. You are responsible for updating and maintaining self-signed certificates.

To use a CA certificate to authenticate EA to a client, generate the self-signed certificate. Then, use this information to generate a certificate signing request (CSR) for a CA certificate. When you receive the certificate, import it into the EA server keystore.

To use a self-signed certificate to authenticate the GUI to the EA server, first generate the self-signed-certificate for the GUI. Then, export the certificate and send it to EA.

To use a CA certificate to authenticate the GUI, first generate the self-signed certificate at the GUI. Then, use the information to generate a certificate signing request (CSR) for a CA certificate. After you obtain a CA certificate, import this information into the GUI keystore.

Generate a Self-Signed Certificate to Authenticate EA to a Client

To use a CA certificate to authenticate EA to a client or the GUI, generate a self-signed-certificate for EA and use the information to create a Certificate Signing Request (CSR).

To generate a self-signed key certificate for the EA server and add it to the EA keystore:

1. At the EA server, type the following command from the *install_dir*/jre/bin directory where *install_dir* is the installation directory path, and press **Enter**.

```
keytool -genkey -alias alias_name         alg_type          keysize
validity_in_days           keystore_path -         password
```

Refer to *Parameters to Generate a Self-Signed Certificate* on page 48 for the parameters.

Following is a sample command used to create a server key certificate:

```
$ keytool  -genkey -alias SEASkeycert -keyalg RSA -keysize 1024 -validity 360
-keystore C:\           \conf\system\keystore -storepass password
```

Following are sample commands to create a key certificate. Each uses the -dname option to control the attributes used to define the subject distinguished name:

```
$ keytool  -genkey -alias SEASkeycert -keyalg RSA -keysize 1024 -validity 360
-keystore c:\           \conf\system\keystore -storepass password -dname
“CN=SEAServer, DC=companyname, DC=com”
```

```
$ keytool  -genkey -alias SEASkeycert -keyalg RSA -keysize 1024 -validity 360
-keystore c:\           \conf\system\keystore -storepass password -dname “C=US,
O=companyname, CN=SEAServer”
```

2. If you did not use the -dname option to define the CN, provide the following information:
   - First and last name

   Note: Information you provided in these fields creates the CN attribute in the subject DN.

   - Organizational unit
   - Organization
   - City or locality
   - State or Province (use UPPER CASE characters)
   - Two-letter country code (use UPPER CASE characters)
3. Verify the information and press **Enter**.
4. At the prompt to provide a key password, do not provide a password. Press **Enter**.

Caution: The key certificate and keystore passwords must be the same for EA to function properly.

5. Do one of the following:
   - For CA certificates, continue to Create a PKCS#10 CSR for the GUI.
   - For self-signed certificates, refer to Export a Self-Signed Certificate from the Keystore for the EA Server.

Generate a Self-Signed Certificate for the GUI

To establish secure communications between the GUI and EA, create a key certificate on the GUI.

To create a self-signed key certificate at the GUI:

1. On the GUI computer, type the following command and press **Enter**:

```
keytool -genkey -alias            -keyalg          -keysize         -validity
                 -keystore                storepass
```

The follow example illustrates how to create a key certificate:

```
$ keytool  -genkey -alias SEASGUIkeycert -keyalg RSA -keysize 1024 -validity 360
-keystore c:\keystore\mykeystore -storepass password
```

The following examples illustrate how to create a key certificate using the -dname option to define subject distinguished name:

```
$ keytool  -genkey -alias SEASGUIkeycert -keyalg RSA -keysize 1024 -validity 360
-keystore c:\keystore\mykeystore -storepass password -dname “CN=SEASGUI,
DC=companyname, DC=com”
```

```
$ keytool  -genkey -alias SEASGUIkeycert -keyalg RSA -keysize 1024 -validity 360
-keystore c:\keystore\mykeystore -storepass password -dname “C=US, O=companyname,
CN=SEASGUI”
```

2. If you do not use the -dname option to define the CN attribute, provide the following information:

    - First and last name

    Note: Information you provided in these field creates the CN attribute in the subject DN.

    - Organizational unit
    - Organization
    - City or locality
    - State or Province (use UPPER CASE characters)
    - Two-letter country code (use UPPER CASE characters)

3. Verify the information you provided and press **Enter**.
4. At the prompt to provide a password, do not provide a password. Press **Enter**.

Caution: The key certificate and keystore passwords must be the same for EA to function properly.

5. Do one of the following:

    - For CA certificates, complete the procedure Create a PKCS#10 CSR for the GUI.
    - For a self-signed certificate, export a copy of the file. Refer to Export a Self-Signed Certificate for the GUI.

Parameters to Generate a Self-Signed Certificate

Following are the descriptions of the parameters to generate a self-signed certificate for the GUI and the server:

Parameter	Description
keytool	Invokes the Keytool utility. Type this command with no options to view help.
-genkey	Instructs the Keytool utility to generate a certificate and a private key.
-alias *alias_name*	Certificate name. Identifies the certificate in the keystore.

Parameter	Description
-keyalg *alg_type*	Algorithm type to create the key. This value must be an RSA algorithm.
-keystore *keystore_path*	Path and file name of the keystore file. If you omit this parameter, the keystore is created in your home directory with the file name **.keystore**.
-keysize *keysize*	Size of the key to create. Maximum key size is 2048.
-validity *validity_in_days*	Number of days the certificate is valid.
-storepass *password*	Password of the keystore file.
-dname	Controls attributes that specify the distinguished name in the self-signed certificate or CSR. For example, use domain attributes instead of geographic attributes.

Export a Self-Signed Certificate for the EA Server or the GUI

After you create a self-signed certificate at the EA server, you send this information to the client with which you are communicating. Complete the procedure, *Export a Self-Signed Certificate from the Keystore for the EA Server* on page 49, to export the information.

After you create a self-signed certificate at the GUI, export the certificate and send it to EA. Complete the procedure *Export a Self-Signed Certificate for the GUI* on page 49.

Export a Self-Signed Certificate from the Keystore for the EA Server

Send the self-signed certificate to the entity with which you are communicating.

To export a self-signed certificate generated for the EA server:

1. From the *install_dir*/jre/bin directory on the EA server, type the following command, where *install_dir* is the installation directory. Press **Enter**:

```
keytool -export -alias alias_name -keystore keystore_path -storepass password
-rfc -file cert_file_name.xxx
```

   Refer to *Commands to Export a Self-Signed Certificate* on page 50 for the parameters.

2. Import the certificate into the EA trust store. Refer to *Import a Certificate into the EA Server Trust Store* on page 53.

Export a Self-Signed Certificate for the GUI

If you use a self-signed certificate, export it to a file and import it to the server trust store.

To export the certificate:

1. On the GUI computer, type the following command and press **Enter**.

```
keytool -export -alias alias_name -keystore keystore_path -storepass password
-rfc -file cert_file_name.xxx
```

2. Continue to *Import the Server Certificate into the GUI Trust Store* on page 53.

Commands to Export a Self-Signed Certificate

Following are the parameter descriptions for the export command:

Parameter	Description
keytool	Invokes the Keytool utility.
-export	Exports a copy of the certificate.
-alias *alias_name*	Name of the certificate. It is used to identify the certificate in the keystore.
-keystore *keystore_path*	Path and file name of the keystore. If you do not define this parameter, it is created in your home directory with the file name **.keystore**.
-storepass *password*	Password of the keystore file.
-rfc	Exports the certificate in PEM format; if you do not include this parameter, the certificate is exported in DER format.
-file cert_file_name.xxx	Path and file name of the certificate to export.

Create a CSR

After you create a self-signed certificate for the server or GUI, create a CSR with information from the key and certificate. Submit the CSR to a Certificate Authority (CA) and request a certificate authenticated and signed by the CA. This procedure does not apply to self-signed certificates.

Create a PKCS#10 Certificate Signing Request for the Server

Create a PKCS#10 CSR for the GUI

Parameters to Create a CSR

Create a PKCS#10 Certificate Signing Request for the Server

To create a CSR for the EA certificate:

1. At the EA server, navigate to the *install_dir*/jre/bin directory, where *install_dir* is the EA installation. Type the following command and press **Enter**.

```
keytool -certreq -keystore               -alias            -file CSR_file
           password
```

   Refer to *Parameters to Create a CSR* on page 51 for a description of the parameters.

   Following illustrates how to generate a PKCS#10 CSR for the *SEASkeycert* certificate:

```
                                install_dir
SEASkeycert
```

2. Submit the output file to the CA to request a server certificate.

   When you receive the certificate from the CA, perform the procedure *Import the CA Certificate into the Server Keystore* on page 51.

Create a PKCS#10 CSR for the GUI

Create a CSR that contains key and certificate information from a self-signed certificate. Then, submit it to a CA to request a CA certificate.

To create a CSR for the GUI:

1. On the GUI computer, type the following command and press **Enter**:

```
                          keystore_path        alias_name       CSR_file
           password
```

The following command illustrates how to generate a PKCS#10 CSR for the GUI certificate:

```
                                                           GUIkeycert
```

2. Submit the output file to the CA to request a certificate for the GUI.

Parameters to Create a CSR

Following are the parameters used to create a CSR:

Parameter	Description
keytool	Invokes the Keytool utility.
-certreq	Generates a CSR.
-keystore *keystore_path*	The path to the keystore that contains the certificate to create the CSR for.
-alias *alias_name*	The alias name of the certificate to create the CSR for.
-file *CSR_file*	The path and file name of the CSR to create.
-storepass *password*	The password of the keystore.

Import the CA Certificate Keystore

The keystore stores the private-public key pair and associated CA certificate. Two keystores are maintained: the keystore at the EA server and the GUI. Import certificates used by EA to communicate with clients in the EA keystore. Import certificates used by the GUI to communicate with EA in the GUI keystore.

Import the CA Certificate into the Server Keystore

Import the CA Certificate to the GUI Keystore

Parameters to Import the CA Certificate into the Keystore

Import the CA Certificate into the Server Keystore

To replace a self-signed certificate with the CA certificate for EA:

1. Navigate to the *install_dir*/jre/bin directory on the EA server.

2. Type the following command and press **Enter**.

```
                          keystore_path        alias_name -         password
-     certificate
```

Refer to *Parameters to Import the CA Certificate into the Keystore* on page 52 for a description of the parameters.

Following is a sample command to import a CA certificate to the server keystore:

```
                               install_dir
```

3. When prompted to trust the certificate, type **yes** and press **Enter**.
4. Obtain a copy of the root certificate of the CA. Distribute it to EA servers that require client authentication, the remote computer running the EA GUI, and clients.

Import the CA Certificate to the GUI Keystore

The following procedure replaces the self-signed certificate created in *Generate a Self-Signed Certificate for the GUI* on page 47 with the CA certificate for the GUI.

To import the CA certificate into the GUI keystore:

1. At the GUI, type the following command and press **Enter**:

```
                          keystore_path        alias_name -         password
-     certificate
```

Refer to *Parameters to Import the CA Certificate into the Keystore* on page 52 for a description of the parameters.

The following illustrates how to import a CA certificate to the GUI keystore:

```

```

2. When prompted, Trust this certificate?, type **yes** and press **Enter**.
3. Obtain a copy of the root certificate of the CA and import it into the trust store of EA. Refer to *Import the Server Certificate into the GUI Trust Store* on page 53.

Parameters to Import the CA Certificate into the Keystore

Following are the parameters used to import a CA certificate into a keystore:

Parameter	Description
keytool	Invokes the Keytool utility.
-import	Instructs Keytool to import a certificate to the keystore.

Parameter	Description
-keystore *keystore_path*	Path and file name of the keystore file.
-alias *alias_name*	Alias name to identify the certificate in the keystore. Use the same alias as used to create the certificate in *Generate a Self-Signed Certificate for the GUI* on page 47.
-storepass *password*	Password of the keystore file.
-file *certificate*	Location of the CA certificate to import.

Import Certificates into the Trust Store

You import certificates into the trust store based on the connection you are securing. To secure the connection between EA and a client, the EA trust store must contain the root certificate for each server and client that EA connects to. To secure the connection between the GUI and EA, the GUI trust store must contain a copy of the public key of EA.

Import a Certificate into the EA Server Trust Store

Import the Server Certificate into the GUI Trust Store

Parameters Used to Import the Certificate into the Trust Store

Import a Certificate into the EA Server Trust Store

For each server that EA connects to, obtain the root certificate and import it into the trust store.

To import the CA root or the public key to the EA server trust store:

1. Move to the *install_dir*/jre/bin directory on EA, where *install_dir* is the EA installation.
2. Type the following command, and press **Enter**:

```
                                store_path -          password -     certificate
```

   Following illustrates how to import the server certificate to the EA trust store:

```
                                 install_dir
mypassword -file c:\TrustCertificate\cert.txt
```

3. When prompted, Trust this certificate?, type **yes** and press **Enter**.

Import the Server Certificate into the GUI Trust Store

To import the server CA root or the public key to the GUI trust store:

1. On the GUI computer, type the following command and press **Enter**:

```
keytool -import -keystore truststore_path -storepass password -file certificate
```

The following illustrates how to import the server certificate to the GUI trust store:

```
$ keytool -import -keystore c:\truststore\mytruststore -storepass mypassword
-file c:\TrustCertificate\cert.txt
```

2. When prompted, Trust this certificate?, type **yes** and press **Enter**.

Parameters Used to Import the Certificate into the Trust Store

Following are the parameters used to import a CA certificate into a trust store:

Parameter	Description
keytool	Invokes the Keytool utility.
-import	Instructs Keytool to import a certificate to the trust store.
-keystore *truststore_path*	Path and file name of the trust store file.
-storepass *password*	Password of the trust store file. Default=changeit.
-file *certificate*	Location of the public certificate to import.

Configure EA to Access the SSL Keystore

You must have a self-signed or CA certificate in the server keystore before completing this procedure. See *Generate a Self-Signed Certificate to Authenticate EA to a Client* on page 46 or *Import the CA Certificate Keystore* on page 51.

The SSL keystore file stores the certificate used to secure LDAP servers and to perform TLS/SSL negotiations with clients.

To configure EA to access the keystore:

1. From the **Manage** menu, select **System Settings**.
2. Click the **SSL** tab.
3. Specify the following information:
    - Protocol
    - Keystore File
    - Keystore Password
4. Click **OK**.

Configure EA to Access the SSL Trust Store

The trust store file contains the CA and self-signed certificates that authenticate secure connections to EA from clients and from EA to LDAP servers. It also contains certificates to validate signatures on CRLs and certificates.

This procedure assumes that you have imported client and server certificates to the server trust store. See *Import a Certificate into the EA Server Trust Store* on page 53.

To enable the EA trust store:

1. From the **Manage** menu, select **System Settings**.
2. Click the **Trusted Certificate** tab.
3. Specify the following parameters:
   - Trust Store File
   - Trust Store Password
4. Click **OK**.

Configure the Secure Connection Listener

To enable a client application to connect securely to EA:

1. From the **Manage** menu, select **System Settings**.
2. Click the **Listeners** tab.
3. In the Secure Listener section, specify the following parameters:
   - IP Address
   - Port
   - Keystore alias
   - Enabled
4. Click **OK**.

Configure SSL or TLS Between the GUI and the EA Server

Complete this procedure to configure the GUI to use SSL/TLS to connect to EA. Before you complete this procedure, have a certificate at the GUI and EA, import the GUI certificate into the EA trust store and import the root certificate of the EA server into the GUI trust store.

To configure SSL or TLS between the GUI and EA:

1. From the Login screen, click **Config**.
2. Specify the Keystore File and password. Click **Next**.
3. On the Create SSL/TLS Trust Store Info screen, specify the trust store file and password. Click **Next**.
4. From the Confirm screen, click **Save**.
5. Click **Close** to return to the Login screen.

Create and Manage Certificate Revocation Lists (CRL) Definitions

Create Certificate Revocation List (CRL) definitions to download published CRLs. Published CRLs validate certificates and determine if a certificate has been revoked. During certificate validation, EA checks CRLs in the CV definition and determines if a certificate is revoked.

Organization of the CRL Configuration

The configuration instructs you how to create a basic setup. After you complete a scenario, test the connection to ensure it is correctly configured. Determine which features apply to your environment and configure only those.

How to Use the Worksheets

Before you configure a definition, gather the information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

After you complete the definition, test it to make sure EA can connect to SSP and the outbound server. Continue to the next section and add another security feature.

Prerequisites

Before you configure a CRL definition, configure a system-wide definitions to connect to the LDAP server. Refer to *System-Wide LDAP Connection Definition* on page 37 or *Define a System-Wide HTTP Connection Definition* on page 39.

Basic CRL Definition Worksheet

Before you configure a CRL definition, gather the following information. Use this worksheet to configure the basic CRL definition.

Parameter	Description	Value
CRL Definition Name	Name assigned to the CRL definition.	
Clock Tolerance	Difference allowed between EA and the CRL clock, in seconds	
Use defined connection	Select the connection definition you created.	
Base DN	Starting point in the directory to begin the search.	

Create a Basic CRL Definition

Create a CRL definition to allow EA to determine if a certificate has been revoked early. Certificate authorities issue CRLs periodically and publish them to HTTP or LDAP servers so that they can be referenced for up-to-date information about revoked certificates. To allow EA to access this information and validate certificates received against the CRL list, create a CRL definition.

To create a CRL definition:

1. From the **Manage** menu, click **CRL Definitions**.
2. From the CRL Definitions screen, click +.
3. On the General screen, specify the CRL parameters. Use the worksheet to fill out the fields.
4. Click **Next** twice.
5. Enable use defined connection and select the system-wide connection you defined. Click **Next**.
6. Type the value of the base DN.
7. Click **Next**.
8. On the Confirm screen, verify the parameters and click **Save** and **Close**.

Advanced CRL Definition

After you configure and test a basic CRL definition, you can add one or more advanced functions. Advanced features include: modify the cache method, reject an expired CRL, verify the signature of a CRL, and specify a query as a URL.

Advanced CRL Definition Worksheet

Before you configure advanced CRL features, gather the information to configure advanced options:

Parameter	Description	Value
Cache CRL	Modify the cache method to one of the following: ◆ Refresh cache at interval—How often to refresh the CRL, in minutes ◆ Refresh CRL on every check	
Reject expired CRL	Reject an expired CRL. A revoked certificate causes a certificate validation to fail	
Verify Signature	Verify the signature of the CRL. The certificate of the CRL issuer must be in the system trust store or a certificate in the certificate chain in the validation request	
Specify query parameters	Define the connection to the server as you specify query parameters	
Specify query as URL	Specify a URL to query where the CRL is stored	

Add Features to a CRL Definition

To add advanced features to a CRL definition:

1. From the **Manage** menu, click **CRL Definitions**.

2. Double-click the CRL Definitions to modify.
3. On the General screen, modify the advanced fields and click **Next**. Use the worksheet to fill out the fields.
4. On the Query General screen, select one of the options to identify how to query for the list:
   - Specify query parameters
   - Specify query as URL
5. On the Confirm screen, verify the parameters and click **Save** and **Close**.

Edit a CRL Definition

Edit a CRL definition from a certificate validation it is associated with.

To edit a CRL:

1. From the Certificate Validation Definitions window, double-click the definition to modify.
2. Click the **Referenced CRLs** tab. Select the CRL definition to modify or copy and click ✎.
3. Modify parameters as required.
4. Click the **Summary** tab and review all parameters. Click **OK**.

Copy a CRL Definition

Copy a CRL definition using the **Manage** menu or when you are in a CV definition. To copy a CRL definition from the CV definition, it cannot be referenced.

To copy a CRL:

1. From the Certificate Validation Definitions window, double-click the CV to modify.
2. Click the **Referenced CRLs** tab. Highlight the CRL definition to modify and click ✎.
3. Rename the CRL.
4. Change the parameter settings as required.
5. On the Confirm screen, verify the settings and click **Save**.

Delete a CRL Definition

Delete a CRL from the **Manage** menu. It cannot be deleted from a CV definition.

To delete a CRL:

1. From the **Manage** menu, click **CRL Definitions**.
2. Select the CRL to delete and click −.
3. Click **OK**.

Configure Active Directory to Use with EA

You can store certificates, SSH keys, users, or IP addresses in Active Directory and use EA to access the information. EA then provides it to Sterling Secure Proxy for user authentication.

The Active Directory Schema

The Active Directory schema defines the objects allowed in a directory. Use the schema file to extend Active Directory. Complete the procedures in this section to extend the schema.

Complete the following procedures to extend Active Directory:

Extend Schema for Active Directory on page 61

Add an Active Directory Schema Node on page 61

If you use the SSH protocol, *Add the sshPublicKey Attribute to the User Class* on page 62

Add superior to the loginCredentials Class on page 62

Extend Schema for Active Directory

To add a directory object for Lookup Login Credentials, extend the schema.

To extend the Active Directory schema:

1. Log into the AD domain. Use an administrator account that is a member of the Schema Admins group.
2. Edit the file called seas_ad.ldf and replace occurrences of DC=example,DC=com with your AD domain name. For example, if your AD domain is acme.local, replace DC=example,DC=com with DC=acme,DC=local.
3. Save the file.
4. Make a backup of Active Directory.
5. Run the following command:

```
ldifde -i -f seas_ad.ldf
```

Note: If you get the error, access denied, the account you logged in may not be an administrator in the Schema Admins group. If you meet these requirements and you get the error, run the command using the administrator account for the domain controller.

Add an Active Directory Schema Node

When you add attributes to AD, an Active Directory schema node must exist under the Console Root. If the node does not exist, add it as follows:

1. Select **Start>Run**.
2. On the **Open** field, type regsvr32 schmmgmt.dll.
3. Click **OK**.
4. Select **File>Add/Remove Snap-in**.

5. Click **Add**.
6. Select Active Directory Schema and click **Add**.
7. Click **OK**.

Add the sshPublicKey Attribute to the User Class

If you use the SSH protocol, add the sshPublicKey attribute. Before you complete this procedure, make sure an active schema node exists. Refer to *Add an Active Directory Schema Node* on page 61.

To add the sshPublicKey attribute:

1. Open Microsoft Management Console.
2. Right-click **Active Directory Schema** and click **Reload the schema**.
3. Expand the Active Directory Schema and click **Classes**.
4. Right-click **user class** on the list and click **Properties**.

Note: If you use a different class than the system user class, select it.

5. On the Attributes tab, click **Add**.
6. On the Select Schema Object dialog, select sshPublicKey from the list. Click **OK**.
7. Click **Apply**. Click **OK** to close the Properties dialog.

Add superior to the loginCredentials Class

To complete the Active Directory schema setup, add a superior property to the loginCredentials class. Make sure an active schema node exists.

To add superior to the loginCredentials class:

1. Open the MMC.
2. Right-click **Active Directory Schema** and click **Reload the schema**.
3. Expand Active Directory Schema.
4. Click **Classes**.
5. Right-click **loginCredentials class** on the list and select Properties.
6. Click the Relationship tab and select **Add Superior**.
7. Select the user class and click **OK**.

Note: If you use a different class than the system user class, select it.

8. Click **Apply**. Click **OK** to close the Properties dialog.

Set Up Mapped Credentials in Active Directory

If you want to log in to a backend server using different credentials than those presented by trading partners, you must first set up mapped credentials in AD. Complete the following procedures to set up mapped credentials:

If necessary, *Add the ADSI Edit Node and the Domain Node* on page 63

Assign Mapped Credentials in AD on page 63

Create a User Authentication Profile in EA on page 64

Add the ADSI Edit Node and the Domain Node

Before you assign mapped credentials, you must add an ADSI Edit node and a domain node. To determine if an ADSI Edit node exists, expand the Console Root. Look for Domain node under ADSI Edit. If the node does not exist, use this procedure to add it.

To add the ADSI edit node in AD:

1. Select **File>Add/Remove Snap-in** and click **Add**.
2. Select **ADSI Edit** and click **Add**.
3. Click **OK**.

To add the domain node:

1. Right click **ADSI Edit** and click **Connect to**.
2. On the Connection Settings dialog, enable **Select a well known naming context** and select Domain.
3. Click **OK**.

Assign Mapped Credentials in AD

Before you assign mapped credentials, be sure to extend the schema. Refer to *Extend Schema for Active Directory* on page 61. Additionally, the ADSI Edit node and the Domain node must exist.

To assign map credentials in AD:

1. Open the MMC.
2. Expand ADSI Edit.
3. Expand the Domain node. Right-click **Domain** and select **Update Schema Now**.
4. Expand the node for your AD domain.
5. Expand the Users container.

Note: If users are stored in a different container, such as OU=External Partners, navigate to that container and expand it.

6. Right click the user to modify and select **New>Object**.
7. Select **loginCredentials** from the class list and click **Next**.
8. Type a name for the object and click **Next**.

   For example, name the object Credentials for *XXXX*, where *XXXX* is the destination service.

9. Click **More Attributes**.
10. Select loginTarget from the **Select a property to view** field.
11. On the **Edit attribute** field, type the destination service name for the server.

    This value must match the destination service field in the Sterling Secure Proxy netmap.
12. Click **Set** and click **OK**.
13. To return a routing key when you log in to the destination service, do the following:
    a. Click **More Attributes**.
    b. Select routingKeyName from the **Select a property to view** field.
    c. In the **Edit attribute** field, type the routing key label for the public/private SSH key pair used to log in to the server.
    d. Click **Set** and click **OK**.
14. To return a mapped user ID to log in to the destination service:
    a. Click **More Attributes**.
    b. Select loginId from the **Select a property to view** field.
    c. On the **Edit attribute** field, type the user ID used to log in to the server.
    d. Click **Set** and **OK**.
15. To return a mapped password to login to the destination service:
    a. Click **More Attributes**.
    b. Select loginPwd from the **Select a property to view** field.
    c. On the **Edit attribute** field, type the password used to log in to the server.

    Note: Type the password in hexadecimal, with each character in the form 0xHH, separated by spaces. For example, if password=password, enter: 0x70 0x61 0x73 0x73 0x77 0x6f 0x72 0x64. To find converters, search for text to hexadecimal converter on Google.

    d. Click **Set** and click **OK**.
16. Click **Finish**.

Create a User Authentication Profile in EA

Use the table below to identify the values to assign in EA to create a user authentication. Refer to *Create a Basic LDAP Authentication Definition* on page 141.

EA Field	Value
Profile name	Name for the profile
Host	Host name or IP address of the Active Directory server
Port	Port number to connect to the Active Directory server.
LDAP principal to bind	Specify User DN Replace base DN with the distinguished name where users are stored, for example, CN=Users,DC=example,DC=com.

EA Field	Value
Application Feature	Lookup login credentials (Sterling) to configure user ID, password, or routing key mapping.
User authenticated user connection	Connection definition for the Active Directory server.
Specify query parameters	Enable to allow you to define the query parameters.
Return Attributes	Mapped credentials to return. By default, loginId, loginPwd, and routingKeyName are returned. Delete any attributes you don't want to map.

Configure SSH Public Keys

Assign SSH public keys to users in AD for trading partners who log in using SFTP and use EA authentication. The trading partner must use either key or password and key as the authentication.

Complete the following procedures to configure SSH public keys in AD and EA:

Assign SSH Public Keys

Create an SSH Key Authentication Profile

Assign SSH Public Keys

Complete this procedure to add the SSH public keys to the AD database.

Before you assign SSH public keys, extend the schema. Refer to *Extend Schema for Active Directory* on page 61. The ADSI Edit node and the Domain node must exist. To check for the ADSI Edit node definition, open AD and expand Console Root. Look for Domain node under ADSI Edit. If they do not exist, add them. Refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To assign SSH pubic keys:

1. Open MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.
4. Right-click the Domain node and click **Update Schema Now**.
5. Expand the node for your AD domain.
6. Expand the Users container.

Note: If users are stored in a different container, navigate to that container and expand it.

7. Right-click the user to modify and select **Properties**.
8. Select sshPublicKey on the list and click **Edit**.
9. Open the SSH public key file.
10. Copy the base64 key and paste it into a new text document.

    The base64 key is the lines between the BEGIN SSH2 PUBLIC KEY and END SSH2 PUBLIC KEY markers, excluding lines that start with keywords like Comment.

11. Remove newlines from the text, leaving a single long line of base64 text.
12. Copy the single line of base64 text.
13. In MMC, paste the single line into the **Value to add** field and click **Add**.
14. Repeat step 8through 12 for any other public keys. Click **OK** when all keys have been added.
15. Click **Apply** to save changes, then click **OK** to close the Properties dialog.

Create an SSH Key Authentication Profile

Create an SSH key authentication profile in EA to use the keys you configured to authenticate users. Refer to the procedure, *Create an SSH Key Authentication Definition* on page 129. Use the following table to identify the values to assign to the EA fields.

EA Field	Value
Authentication type	SSHKEY
Profile name	Name for the profile
Name	Automatically populated with sshPublicKeyQuery
Connection method	Use globally defined connection
Global connection definition	Definition you created for the LDAP server
Specify Query Parameters	Allows you to specify query parameters
Base DN	Distinguished name for the user container. For example, CN=Users,DC=example,DC=com.
Application Feature	Lookup login credentials (Sterling) to configure user ID, password, or routing key mapping.
Use globally defined connection	Connection definition for the Active Directory server
Specify query parameters	Allow you to define the query parameters
Return Attributes	Mapped credentials. By default, loginId, loginPwd, and routingKeyName are returned. Delete attributes you don't want to map.

Validate Certificates Stored in Active Directory

To validate users with certificates stored in Active Directory (AD), configure AD and EA to look up certificates through an LDAP query. Add certificates to AD in one of the following ways:

1. Publish third-party certificates to the Active Directory Enterprise Trust

   When a certificate is published to the Active Directory Enterprise Trust, it is added to the multi-value cACertificate attribute of the following object:

```
CN=NTAuthCertificates,CN=Public Key Services,CN=Services,CN=Configuration,
DC=<domain>,DC=<com>
```

2. Issue a certificate to a user through the domain's Certificate Service web site, http://<dcname>/certsrv/.

   When a Certificate Service web site issues a certificate to user, the data is stored in the userCertificate attribute on the AD user's record. In addition, the subject of the issued certificate is set to the distinguished user name.

For each method used to store certificates in AD, you must define a certificate validation profile to validate certificates based on where they are stored. In Sterling Secure Proxy, you must define separate inbound nodes or adapters for clients that have certificates published in the AD Enterprise Trust and for clients with certificates issued through the Certificate Service web site.

You can add certificates to AD by publishing third-party certificates to the AD Enterprise Trust. Complete the following procedures to prepare the AD Enterprise Trust for certificate validation:

Publish Certificates to the Active Directory Enterprise Trust

Use the following procedures to publish certificates to the Active Directory Enterprise Trust:

Publish Certificates

View Certificates Published to Active Directory Enterprise Trust

Create an EA Certificate Profile to Validate Certificates in the AD Enterprise Trust

Assign Users to a Partner Certificate in the AD Enterprise Trust

Create a Profile on EA to Authenticate a User ID with Certificate in Enterprise Trust

Publish Certificates

To publish certificates to the AD Enterprise Trust:

1. Log in to the AD domain controller. Use an administrator account.
2. Run the following command:

```
certutil -dspublish -f          NTAuthCA
```

   where *filename* is the path to the certificate in text Base-64 or binary DER format.

Refer to http://support.microsoft.com/kb/295663 (Article ID: 295663; How to import third-party certification authority (CA) certificates into the Enterprise NTAuth store) for more information.

View Certificates Published to Active Directory Enterprise Trust

You can view certificates published to the Active Directory Enterprise Trust.

To view certificates:

1. Log in to the AD domain controller. Use an administrator account.
2. Open the MMC.
3. Look for Certificates (Local Computer) under Console Root. If no certificate is displayed, add it as follows:
   a. Select **File>Add/Remove Snap-in**.
   b. Click **Add**.

c. Select **Certificates**.

d. Click **Add**.

e. Enable **Computer Account** and click **Next**.

f. Enable **Local computer**.

g. Click **Finish**.

h. Select **Enterprise PKI**.

i. Click **Add**.

j. Click **Close**.

4. Expand Certificates (Local Computer).
5. Expand Enterprise Trust.
6. Select **Certificates**. The certificates are displayed in the list to the right of the screen.

Create an EA Certificate Profile to Validate Certificates in the AD Enterprise Trust

After you add certificates to the AD Enterprise Trust, create an EA profile to use certificates stored in the AD Enterprise Trust. Before you create the profile, be sure to define a connection definition for the AD server. Use the following table to determine the values to assign in EA. Refer to *Configure EA to Validate Certificates* on page 105.

EA Field	Value
Name	Name for the CV definition, for example, AD_CertVal_EnterpriseTrust
Define query to verify certificate subject	Verify the subject of a certificate using an attribute query
Use defined connection	Select the connection definition for your AD server
Verify certificate matches certificate in directory	Compare the certificate to the one stored in the directory entry. Type cACertificate in the **Certificate Attribute** field.
Base DN	The starting point in the directory to begin the search. Type the following: CN=NTAuthCertificates,CN=Public Key Services,CN=Services,CN=Configuration,DC=domain,DC=com where d*omain* and *com* are the values for your domain.
Return Attributes	dn, cACertificate to define attributes to return from entries that match
Scope	Select base to search only base DN entries

Assign Users to a Partner Certificate in the AD Enterprise Trust

After you add a partner certificate to AD, you can assign users to the certificate.

To assign users to a certificate:

1. Open the MMC.
2. Expand Active Directory users and computers.

3. Expand the node for your AD domain.
4. Expand the user container.

Note: If users are stored in different containers, such as OU=External Partners, navigate to that container and expand it instead.

5. Double-click the user to assign the certificate to.
6. Select the Published Certificates tab.
7. Click **Add from Store**.
8. Select the partner certificate from the list.

Note: If the certificate list gets too large, select Add from File, then select the file for the partner certificate. Be careful to select the correct certificate file. Make sure that the certificate is published to the Active Directory Enterprise Trust, or publish it immediately.

9. Click **OK**.
10. Click **Apply** to save changes.
11. Click **OK** to close the Properties dialog.
12. Repeat steps 5 to 11 for any other users you want to assign to the partner certificate.

Create a Profile on EA to Authenticate a User ID with Certificate in Enterprise Trust

The previous procedures add the partner certificate data to the userCertificate attribute of the Active Directory user object. You can now require that partner users present a specific certificate for authentication by creating a user authentication profile in EA that compares the certificate from the client against the certificate assigned to the user in Active Directory.

In order to use this user authentication profile, a certificate validation request must be issued before the user authentication request, and both requests must be issued on the same conversation. Sterling Secure Proxy uses the policy definition when selecting the External Authentication check boxes for both certificate validation and user authentication, and specifying the corresponding profile names.

Use the definitions in the following table to complete the procedure. Refer to *Create LDAP Authentication Definitions* on page 141.

EA Field	Value to Assign
Name	Name for the profile, such as AD_UserAuth_EnterpriseTrust
Protocol	Appropriate protocol for accessing Active Directory
Host	Host name or IP address of the AD domain controller
Port	LDAP listen port number for the AD domain controller
LDAP principal to bind	Specify user DN Replace the base DN with the distinguished name of the container where users are stored.

EA Field	Value to Assign
Attribute Query Definition fields	
Name	Name of the attribute query definition
Connection Specification	Use authenticated user’s connection
Base DN	{principal}
Return Attributes	userCertificate
Scope	Base
Attribute Assertion Definitions fields	
Name	Name for the assertion such as VerifyUserCertificate
Assertion	“{attr[GetUserCertificate].userCertificate} == {attr[VerifyCertSubject].cACertificate} This assertion compares the values of the userCertificate attribute returned by the GetUserCertificate query against the values of the cACertificate attribute returned by the VerifyCertSubject query from the prior certificate validation request for the client certificate. If one of the values of the userCertificate attribute matches one of the attributes of the cACertificate attributes returned by the VerifyCertSubject query, the assertion succeeds.

Validate Certificates Issued by Windows Certificate Service Web Site

When a user is issued a certificate through the Certificate Service web site, the certificate data is stored in the userCertificate attribute on the AD user's record. In addition, the subject of the issued certificate is set to the distinguished user name.

View Certificates Issued by Windows Certificate Service Web Site

To view certificates:

1. Log in to the domain controller. Use an administrator account.
2. Open the MMC.
3. Look for Certification Authority (Local) under Console Root. If it is not found, add it as follows:
   a. Select **File>Add/Remove Snap-in**.
   b. Click **Add**.
   c. Select **Certification Authority**.
   d. Click **Add**.
   e. Enable **Local computer**.
   f. Click **Finish**.
   g. Click **Close**.
4. Expand Certification Authority (Local).
5. Expand the node for the CA.

6. Select **Issued Certificates**. The certificates are displayed to the right.

Create a Profile in EA to Validate Certificates Issued by a Certificate Service Web Site

Create a profile to validate certificates issued by a certificate service web site. Refer to Create and *Configure EA to Validate Certificates* on page 105.

Create a Profile in EA to Authenticate a User ID with a Certificate in the AD User Record

Create an authentication definition to create a query to return the userCertificate attribute of the user record and an assertion to verify that the certificate sent by the client matches one of the certificates assigned to the user.

The assertion compares the value(s) of the userCertificate attribute returned by the GetUserCertificate query against the value(s) of the userCertificate attribute returned by the VerifyCertSubject query from the prior certificate validation request for the client certificate. If one of the values of the userCertificate attribute matches one of the attributes of the userCertificate attributes returned by the VerifyCertSubject query, the assertion succeeds.

Refer to *Create LDAP Authentication Definitions* on page 141 for instructions. Use the values in the following table to determine the values to assign in EA.

EA Field	Value
Name	Name for the profile, such as AD_UserAuth_EnterpriseTrust
Protocol	The appropriate protocol for accessing Active Directory
Host	Host name or IP address of the AD domain controller
Port	LDAP listen port number for the AD domain controller
LDAP principal to bind	Specify user DN Replace base DN with the distinguished name of the container where users are stored, such as CN=Users,DC=acme,DC=com.
Attribute Query Definition fields	
Name	Name of the attribute query definition
Connection Specification	Use authenticated user’s connection
Base DN	{principal}
Return Attributes	userCertificate
Scope	Base
Attribute Assertion Definitions fields	
Name	Name for the assertion such as VerifyUserCertificate
Assertion	“{attr[GetUserCertificate].userCertificate} == {attr[VerifyCertSubject].userCertificate} Compares the userCertificate attribute returned by the GetUserCertificate query against the userCertificate attribute returned by the VerifyCertSubject query from the prior certificate validation request for the client certificate. If a value in the userCertificate attribute matches an attribute of the userCertificate returned by the VerifyCertSubject query, the assertion succeeds.

Ensure that Only Authorized Users Can Access a Service

You can configure EA to receive the name of the service that the user requested during user authentication and SSH authentication requests. You write a query to enforce that only users authorized to use a service can log in to that service.

Consider the following sample directory structure:

```
DC=example,DC=com
  CN=SEAS
   CN=Service Groups
     CN=HTTP-Service Users
      ou=HTTP-Service
      uniqueMember=cn=partneruser01,cn=Partner Users,dc=example,dc=com
      uniqueMember=cn=partneruser02,cn=Partner Users,dc=example,dc=com
     CN=FTP-Service and SSH-Service Users
      ou=FTP-Service
      ou=SSH-Service
      uniqueMember=cn=partneruser03,cn=Partner Users,dc=example,dc=com
      uniqueMember=cn=partneruser04,cn=Partner Users,dc=example,dc=com
```

In the directory structure, a container called Service Groups contains a list of groups corresponding to services. Each group contains the distinguished names of the user records that are members of the group. The ou attribute of the group specifies the service name corresponding to the group.

The group HTTP-Service Users contains users that are allowed to access the service called HTTP-Service. The group FTP-Service and SSH-Service Users contains users allowed to access the services called FTP-Service and SSH-Service.

With this directory structure, you can define queries in EA user authentication profiles that make sure that members of a group for a service can log in to that service.

Complete the following procedures to configure access to a service by authorized users only:

Create a Container for Service Groups on page 72

Add a User ID to a User Group for a Service in Active Directory on page 73

Add a Query to an EA Authentication Profile to Validate the User ID and Service on page 74

Create a Container for Service Groups

Create a container for service groups to define a list of groups corresponding to services.

Before you create a container for service groups, the ADSI Edit node and Domain node must exist. To check for the ADSI Edit node definition, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, add them. Refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To create a container for service groups:

1. Open the MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.

4. Expand the node for your AD domain.
5. Find the parent container where you want to create the container (Ex: SEAS). If it does not exist, create it as follows:
    a. Right-click the node for your AD domain and select **New>Object**.
    b. On the Create Object dialog, select the container class from the list.
    c. Click **Next**.
    d. Type the name for the parent container, such as SEAS.
    e. Click **Next**.
    f. Click **Finish**.
6. Right-click the parent container and select **New>Object**.
7. On the Create Object dialog, select the container class from the list.
8. Click **Next**.
9. Type the name for the container, such as, Service Groups.
10. Click **Finish**.

Add a User ID to a User Group for a Service in Active Directory

Before you add a user ID to a group make sure the ADSI Edit node and the Domain node exist. To check for the ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If they do not exist, add them. Refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To add a user ID to a user group for an AD service:

1. Open MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.
4. Expand the node for your AD domain.
5. Right-click the user you want to add to a group and select **Properties**.
6. Select distinguishedName on the attribute list.
7. Click **Edit**.
8. Copy the value for the distinguished name.
9. Click **Cancel** twice.
10. Navigate to the container for the service groups and expand the node.
11. On the list to the right, find the group entry of the services where you want to add the user, for example, CN=HTTP-Service Users. If it does not exist, add it as follows:
    a. Right-click the user groups container and select **New>Object**.
    b. Select the groupOfUniqueNames class from the list and click **Next**.
    c. Type a name for the group, such as, HTTP-Service Users.
    d. Paste the distinguished name of the user record, for example, cn=partneruser01,cn=Partner Users,dc=example,dc=com.
    e. Click **Next**.

f. Click **More Attributes**.

g. Select ou from the **Select a property to view** field.

h. Type the service name, as it is configured on the **destination service name** field on the SSP netmap, for example, HTTP-Service, and click **Add**.

Repeat this step for other services to assign to the group.

i. Click **OK**.

j. Click **Finish**.

12. If a group entry for the service already exists, add the user to the group as follows:

a. Double-click the group entry for the service.

b. Select uniqueMember on the list of attributes and click **Edit**.

c. Paste the distinguished name of the user record in the **Value to add** field.

d. Click **Add**.

e. Click **OK**.

f. Click **Apply** to save changes. Click **OK** to close the properties dialog.

Add a Query to an EA Authentication Profile to Validate the User ID and Service

Add a query to an EA authentication profile to validate the user ID and service. This procedure assumes that you have already created an authentication definition. Refer to *Query Parameters Worksheet* on page 153 for instructions.

Use the following table to determine the values to assign in EA:

EA Field	Value
Name	Name of the attribute query definition
Connection Specification	Use authenticated user’s connection
Specify query parameters	To allow you to define the query parameters
Base DN	Type the distinguished name where service groups are stored, for example, CN=Service Groups,DC=SEAS,DC=example,DC=com
Return Attributes	dn
Scope	Select One Level

Match Attributes	Name=ou Value=destinationService
	Name=uniqueMember Value={principal}

Check for Allowed IP Addresses

EA receives the incoming IP address of partners during certificate, user, and SSH key authentication requests. You can define queries on profiles to make sure incoming IP address is an allowed IP addresses and allow the request only if the IP address is allowed.

Consider the following sample directory structure:

```
DC=example,DC=com
  CN=SEAS
    CN=Allowed Hosts
      CN=partnerhost01
        ipNetworkNumber=xxx.xxx.xxx.xxx
      CN=partnerhost02
        ipNetworkNumber=yyy.yyy.yyy.yyy
      CN=zzz.zzz.zzz.zzz
        ipNetworkNumber=zzz.zzz.zzz.zzz
```

In the directory structure, a container called Allowed Hosts identifies the IP addresses that can access the system. Each entry defines an IP address. An LDAP query is then defined to check for allowed IP addresses. After a valid IP address is found, then verify that the client certificate, public SSH key, or user ID is associated with the IP address.

You can configure user IDs and identify IP addresses to use for log in. The easiest way to configure IP addresses allowed is to add the ipHost class as an auxiliary to the user class. This adds the multi-value ipHostNumber attribute to the user class. The IP addresses assigned to the user can be added to this attribute.

Complete the following procedures to configure EA to validate IP addresses from inbound nodes and determine if they are allowed.

Create a Container for Allowed Hosts in AD on page 75

Add an IP Address to an Allowed Hosts Container in AD on page 76

Add a Query to an EA Profile to Check for Allowed IP Addresses on page 76

Add ipHost as an Auxiliary Class to the User Class on page 77

Assign an IP Address to a User on page 77

Add a Query to the User Authentication Profile to Validate an IP Address and User ID on page 78

Create a Container for Allowed Hosts in AD

Use this procedure to create a container for allowed hosts.

Before you create a container, make sure the ADSI Edit and Domain nodes exist. To check for ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To create a container for allowed hosts in AD:

1. Open MMC.
2. Expand the ADSI Edit container.
3. Expand the Domain node.

4. Find the parent node where the container for allowed hosts will be created. If it does not exist, create it as follows:
   a. Right click the node for your AD domain and select **New>Object**.
   b. On the Create Object dialog, select the container class from the list.
   c. Click **Next**.
   d. Type the name for the parent container.
   e. Click **Next**.
   f. Click **Finish**.
5. Right click the parent container and select **New>Object**.
6. On the Create Object dialog, select the container class from the list and click **Next**.
7. Type the name for the hosts container, for example, Allowed Hosts.
8. Click **Finish**.

Add an IP Address to an Allowed Hosts Container in AD

After you create the host container, add the IP addresses to the container to identify the IP addresses that are allowed. Before you add IP addresses, the ADSI Edit node and the Domain node must exist. To check for the ADSI Edit node definition, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To add an IP address to an allowed host:

1. Open MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.
4. Right click the container for the hosts, for example Allowed Hosts, and select **New>Object**.
5. On the Create Object dialog, select ipNetwork class.
6. Click **Next**.
7. Type a name for the host record. Use a value that identifies the record, such as DNS host name or IP address.
8. Click **Next**.
9. Type the IP address for the partner host.
10. Click **Next**.
11. Click **Finish**.

Add a Query to an EA Profile to Check for Allowed IP Addresses

Define a query to look up the incoming IP address on the Allowed Hosts container. If the IP address is found, the query is successful and the dn attribute of the host record is returned. If the IP address is not found, the query fails and the certificate validation, user authentication, or SSH key authentication request fails.

Complete the procedure, *Query Parameters Worksheet* on page 153, to add the query. Use the values in the following table to configure EA. Create an authentication definition before you define the query definition.

EA Field	Value
Name	Name of the attribute query definition.
Connection Specification	Use globally defined connection Select the connection definition for the AD server.
Specify query parameters	Enable this option to allow you to define the query parameters.
Base DN	Distinguished name where service groups are stored, for example, CN=Allowed Hosts,DC=SEAS,DC=example,DC=com.
Return Attributes	dn
Scope	One Level
Match Attributes	Name=ipNetworkNumber Value=IpAddress

Add ipHost as an Auxiliary Class to the User Class

Before you add IP address as an auxiliary class, be sure an Active Directory schema exists. Refer to *Add an Active Directory Schema Node* on page 61.

To add IP addresses allowed to the ipHost class:

1. Log in to the AD domain as administrator and as a member of the Schema Admins group.
2. Open MMC.
3. Click **Classes**.
4. Right click user the list and select **Properties**.
5. Select the Relationship tab.
6. Click **Add Class** next to Auxiliary classes.
7. Select ipHost on the list and click **OK**.
8. Click **Apply** to save the changes and click **OK** to close the Properties dialog.
9. Right-click the Active Directory Schema node and select **Reload the Schema**.

Assign an IP Address to a User

Before you add IP addresses, the ADSI Edit node and the Domain node must exist. To check for the ADSI Edit node definition, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To assign an IP address to a user:

1. Open the MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.

4. If you added the ipHost class as an auxiliary class to the user class, right-click the Domain node and select **Update Schema Now**.
5. Expand the node for your AD domain.
6. Expand the Users container.

Note: If you store users in a different container, (ex: OU=External Partners), navigate to that container, and expand it instead.

7. Right click the user that you want to modify and select **Properties**.
8. Select the ipHostNumber attribute and type **Edit**.
9. In the **Value to add** field, type the IP address and click **Add**.
10. Repeat step 9 for other IP addresses. Click **OK** when all IP addresses have been added.
11. Click **Apply** to save the changes and click **OK** to close the Properties dialog.

Add a Query to the User Authentication Profile to Validate an IP Address and User ID

This procedure assumes that you have already created a user authentication definition. Complete the procedure, *Query Parameters Worksheet* on page 153 to create an assertion to compare the incoming IP address against the IP addresses assigned to the user. The assertion examines the ipHostNumber attribute of the user record. If it is equal to any of the values, it returns true. If it is not, it compares the incoming IP address against the value(s) in the ipHostNumber attribute. If the IP address is found in any of the values stored in the ipHostNumber attribute, the assertion succeeds. Otherwise, the assertion fails, and the user validation request also fails.

Use the values in the following table to determine the values to assign in EA:

EA Field	Value
Name	Name of the attribute query definition
Connection Specification	Use authenticated user’s connection
Specify query parameters	To define the query parameters
Base DN	{principal}
Return Attributes	dn, ipHostNumber
Scope	Base
Attribute Assertions Definitions	
Name	Name for the assertion
Assertions	{attr[FindUserIPAddrs].ipHostNumber} == any || {attr[FindUserIPAddrs].ipHostNumber} == {ipAddress}

Verify a Certificate Is Associated with an IP Address

After an incoming IP address is validated, you can further validate the user by verifying that the certificate presented by the client is associated with the IP address.

Consider the following sample directory structure:

```
DC=example,DC=com
   CN=SEAS
     CN=Allowed Hosts
        CN=acmehost01
           ipNetworkNumber=xxx.xxx.xxx.xxx
        CN=acmehost02
           ipNetworkNumber=yyy.yyy.yyy.yyy
        CN=internalhost01
           ipNetworkNumber=aaa.aaa.aaa.aaa

     CN=Host Groups
        CN=Acme Inc Hosts
          o=Acme Inc
          uniqueMember=cn=acmehost01,cn=Allowed Hosts,cn=SEAS,dc=example,dc=com
          uniqueMember=cn=acmehost02,cn=Allowed Hosts,cn=SEAS,dc=example,dc=com
        CN=No Org Hosts
          o=none
          uniqueMember=internalhost01,cn=Hosts,cn=SEAS,dc=example,dc=com
```

In the directory, a container called Allowed Hosts contains IP addresses that access the system.

Another container called Host Groups contains groups corresponding to partner organizations. The groups contain the distinguished name of the host records (IP addresses) that are valid for that organization. The organization name is specified in the o attribute.

The group called Acme Inc Hosts contains host IP addresses that present certificates with O=Acme Inc in the subject. The group No Org Hosts contains host IP addresses that present certificates with no organization.

With this directory structure, you can add a query to the certificate validation profiles to enforce the rule that only the IP addresses assigned to a certificate's organization can log in.

Complete the following procedures to configure the ability to verify that a certificate is associated with an IP address:

Create a Container for Host Groups in AD

Use this procedure to create a container for host groups that contains a list of groups corresponding to partner organizations. After you create the container, use the procedure *Add an IP Address to a Host Group in Active Directory* on page 80

Before you complete this procedure, the ADSI Edit node and the Domain node must exist. To check for the ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To create a container for host groups:

1. Open the MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.

4. Find the parent node under which you would like to create the hosts container, for example, SEAS. If the node is not found, create it as follows:
    a. Right click the node for your AD domain and select **New>Object**.
    b. On the Create Object dialog, select the container class from the list.
    c. Click **Next**.
    d. Type the name for the parent container, for example SEAS.
    e. Click **Next**.
    f. Click **Finish**.
5. Right click the parent container and select **New>Object**.
6. On the Create Object dialog, select the container class from the list.
7. Click **Next**.
8. Type the name for the host groups container, such as Host Groups.
9. Click **Finish**.

Add an IP Address to a Host Group in Active Directory

After you create a host container, you add IP addresses that are valid for that particular organization.

Before completing this procedure, the ADSI Edit and Domain nodes must exist. To check for the ADSI Edit node definition, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To add an IP address to the host group you created:

1. Open the MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.
4. Expand the container for the hosts.
5. Find the record for the IP address in the hosts container. If it is not there, add it as follows:
    a. Right click the hosts container and select **New>Object**.
    b. On the Create Object dialog, select ipNetwork from the list.
    c. Click **Next**.
    d. Type a name for the host record. Use any name that identifies the record, like the DNS host name or the IP address.
    e. Click **Next**.
    f. Type the IP address for the partner host.
    g. Click **Next**.
    h. Click **Finish**.
6. Right click the record for the IP address and click **Properties**.
7. Select distinguishedName from the attribute list and click **Edit**.
8. Copy the value for the distinguishedName attribute to the clipboard.
9. Click **Cancel** twice.

10. Navigate to the container for the host groups and expand the node.
11. On the list to right, find the group entry corresponding to the organization of the host you are adding, for example, CN=Acme Inc Hosts. If it does not exist, add it as follows:
    a. Right-click the host groups container and select **New>Object**.
    b. Select the groupOfUniqueNames class from the list and click **Next**.
    c. Type a name for the group, for example, <organization> Hosts. If the certificate has no organization, type a name to identify this, for example, No Org Hosts.
    d. Paste the distinguished name of the partner host record that you copied in step 8.
    e. Click **Next**.
    f. Click **More Attributes**.
    g. Select o (organization) from the **Select a property to view** field.
    h. Type the organization name, as it appears on the subject of the partner certificate, for example, Acme Inc. If the certificate has no organization, type none.
    i. Click **Add**.
    j. Click **OK**.
    k. Click **Finish**.

Add a Query to Validate an IP Address and Certificate

This procedure assumes that you have already created a certificate authentication definition in EA. Complete the procedure, *Query Parameters Worksheet* on page 153 to create an assertion and compare the incoming IP address against the IP addresses assigned to the user.

The FindHostGroup query you define looks up the host group corresponding to the certificate's organization and including the incoming IP address as a member. If the group is not found, the certificate validation request fails.

Use the values in the following tables to determine the values to assign in EA. Use the first set of values to define the query to look up an incoming IP address and the second values to create a query to find the host group for the certificate's organization. When you add the queries you defined, place the first query called FindHostDN first in the order and FindHostDN second in the list.

EA Field	Value to Assign
Name	Name of the attribute query definition, for example, FindHostDN.
Connection Specification	Use globally defined connection Select the connection definition for the AD server.
Specify query parameters	To define the query parameters
Base DN	Distinguished name for the hosts container, for example, CN=Allowed Hosts,CN=SEAS,DC=example,DC=com.
Return Attributes	dn, flags
Scope	One Level

EA Field	Value to Assign
Match Attributes	Name=ipNetworkNumber Value=IpAddress
Attribute Assertions Definitions	
Name	Name for the assertion
Assertions	{attr[FindUserIPAddrs].ipHostNumber} == any || {attr[FindUserIPAddrs].ipHostNumber} == {ipAddress}

EA Field	Value to Assign
Name	Name of the attribute query definition, for example, FindHostGroup
Connection Specification	Use globally defined connection Select the connection definition for the AD server
Specify query parameters	To define the query parameters
Base DN	Distinguished name for the hosts group, for example, CN=Host Groups,CN=SEAS,DC=example,DC=com).
Return Attributes	dn, uniqueMember
Scope	One Level

Match Attributes	Name=o Value=I{subject.o, none}
	Name uniqueMember Value= {attr[FindHostDN].dn}
	Note: If a certificate subject does not define an organization, set the value to **None**. To group hosts for certificates with no organizations, create a host group called No Org Hosts and set the o attribute to none.

Define an Authentication to Check for a Valid IP Address and Associated SSH Key

After an incoming IP address has been validated, it can be further checked by verifying that the public SSH key presented by the client is associated with that IP address.

Consider the following sample directory structure:

```
DC=example,DC=com
  CN=SEAS
      CN=Allowed Hosts
         CN=sshhost01
            ipNetworkNumber=aaa.aaa.aaa.aaa
         CN=sshhost02
            ipNetworkNumber=bbb.bbb.bbb.bbb
         CN=sshhost03
            ipNetworkNumber=ccc.ccc.ccc.ccc

      CN=SSH Public Key Groups
         CN=Keys for sshhost01
           sshPublicKey=<...>
           sshPublicKey=<...>
           sshPublicKey=<...>
             :
           uniqueMember=cn=sshhost01,cn=Hosts,cn=SEAS,dc=example,dc=com
         CN=Keys for sshhost02 and sshhost03
           sshPublicKey=<...>
           sshPublicKey=<...>
           sshPublicKey=<...>
             :
           uniqueMember=cn=sshhost02,cn=Hosts,cn=SEAS,dc=example,dc=com
           uniqueMember=cn=sshhost03,cn=Hosts,cn=SEAS,dc=example,dc=com
```

In the directory structure, a container called Allowed Hosts contains IP addresses with access to the system and SSH Public Key Groups contains a groups corresponding to SSH public keys. The groups contain the distinguished name of the host records (IP addresses) that are valid for the SSH public keys. With such a directory structure, you can add a query to the SSH key authentication profile to enforce only the IP addresses specifically assigned to an SSH public key can log in.

Complete the following procedures section to configure this authentication method:

Create a Container for SSH Public Keys in AD on page 83

Add an ldapPublicKey to the groupOfUniqueNames Class as an Auxiliary Class on page 84

Add an IP Address and SSH Public Key to a Group in Active Directory on page 85

Add a Query to an SSH Key Authentication Profile to Validate the IP address and SSH Public Key on page 86

Create a Container for SSH Public Keys in AD

Create a container for SSH Public Key that contains groups corresponding to SSH public keys.

Before you complete this procedure, the ADSI Edit node and Domain node must exist. To check for the ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To create a container to store SSH public keys in AD:

1. Open the MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.
4. Expand the node for your AD domain.

5. Find the parent container where you create the container for the keys, for example, SEAS. If the container has not been created, create it as follows:
   a. Right-click the node for your AD domain and select **New>Object**.
   b. On the Create Object dialog, select the container class from the list.
   c. Click **Next**.
   d. Type a parent container name.
   e. Click **Next**.
   f. Click **Finish**.
6. Find the parent node where you want to create the host container. This container is created for directory organization purposes only. If the node is not found, create it as follows:
   a. Right click the node for your AD domain and select **New>Object**.
   b. On the Create Object dialog, select the container class from the list
   c. Click **Next**.
   d. Type the name for the parent container (Ex: SEAS).
   e. Click **Next**.
   f. Click **Finish**.
7. On the Create Object dialog, select the container class from the list and click **Next**.
8. Type a name for the container, for example, SSH Public Keys.
9. Click **Finish**.

Add an ldapPublicKey to the groupOfUniqueNames Class as an Auxiliary Class

To use the sshPublicKey attribute in an object, add the ldapPublicKey class as an auxiliary to the object's class. Because you use the groupOfUniqueNames class to store SSH public key group, you add the ldapPublicKey class as an auxiliary to the groupOfUniqueNames class.

Make sure an Active Schema node exists. To search for a node, open MMC. Look for the Active Directory Schema node under Console Root. If it does not exist, create it. Refer to *Add an Active Directory Schema Node* on page 61.

To add an ldapPublicKey to the groupOfUniqueNames:

1. Log in to the AD domain with an administrator account that is a member of the Schema Admins group.
2. Open the MMC.
3. Click **Classes**.
4. Right click the groupOfUniqueNames class on the list and select **Properties**.
5. Select the Relationship tab.
6. Press **Add Class** next to Auxiliary classes.
7. Select the ldapPublicKey class on the list and click **OK**.
8. Click **Apply** to save the changes and click **OK** to close the Properties dialog.
9. Right-click Active Directory Schema node and select **Reload the Schema**.

Add an IP Address and SSH Public Key to a Group in Active Directory

After you add the SSH Public Key container, you then add the distinguished name of the host records (IP addresses) that are valid for the SSH public keys in by the group.

Before you complete this procedure, the ADSI Edit node and the Domain node must exist. To check for the ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To add an IP address and SSH public key to a group:

1. Open MMC.
2. Expand ADSI Edit.
3. Expand the Domain node.
4. Right click the Domain node and select **Update Schema Now**.
5. Expand the container for the hosts, for example Allowed Hosts, and find the record for the IP address in the hosts container. If it does not exist, add it as follows:
   a. Right click the hosts container and select **New>Object**.
   b. On the Create Object dialog, select the ipNetwork class from the list.
   c. Click **Next**.
   d. Type a name for the host record, for example, the DNS host name or IP address.
   e. Click **Next**.
   f. Type the IP address for the partner host.
   g. Click **Next**.
   h. Click **Finish**.
6. Right click the record corresponding to the IP address and click **Properties**.
7. Select distinguishedName from the attribute list and click **Edit**.
8. Copy the value for the distinguishedName attribute and click **Cancel** twice.
9. Navigate to the SSH public key groups container and expand the node.
10. On the list to the right, find the group entry corresponding to the host you are adding. If the group does not exist, add it as follows:
    a. Right click the SSH public key groups container and select **New>Object**.
    b. Select the groupOfUniqueNames class from the list and click **Next**.
    c. Type a name for the group, for example, Keys for <host>.
    d. Paste the distinguished name of the partner host record that you copied in step 8.
    e. Click **Next** and click **Finish**.
11. Double-click the group entry.
12. Select sshPublicKey on the list of attributes and click **Edit**.
13. Select sshPublicKey on the **Select a property to view** field.

Note: If the sshPublicKey attribute is not listed, you either did not add the ldapPublicKey class as an auxiliary to the groupOfUniqueNames class, or you did not reload an updated schema.

14. Open the SSH public key file and copy the base64 key to the clipboard.

    The base64 key are the lines between the BEGIN SSH2 PUBLIC KEY and END SSH2 PUBLIC KEY markers, excluding lines that start with keywords, like Comment.
15. Paste the copied text into a new text document.
16. Remove any newlines from the pasted text. This creates a single long line of base64 text.
17. Copy the single line of base64 text and paste it into the **Value to add** field.
18. Click **Add**.
19. Click **OK**.
20. Click **Apply**.
21. Click **OK** to save changes.

Add a Query to an SSH Key Authentication Profile to Validate the IP address and SSH Public Key

Complete the procedure, *Query Parameters Worksheet* on page 153 to add a query and validate the IP address and SSH public key. The FindSSHKeyGroup query looks up the SSH pubic key group with the key sent by the client including the IP address. If the group is not found, key authentication request fails.

Use following table to determine the values to define in the Attribute Query:

EA Field	Value to Assign
Name	Name for the attribute query definition.
Connection Specification	Use globally defined connection Select the connection definition for the AD server
Specify query parameters	To define the query parameters
Base DN	Type the distinguished name for the host containers for example, CN=Allowed Hosts,CN=SEAS,DC=example,DC=com
Return Attributes	dn, flags
Scope	One Level
Match Attributes	Name=ipNetworkNumber Value=IpAddress

Select the query you created and move it to the first position in the list.

EA Field	Value to Assign
Name	Name for the attribute query definition, such as, FindSSHKeyGroup.
Connection Specification	Use globally defined connection Select the connection definition for the AD server
Specify query parameters	To define the query parameters

EA Field	Value to Assign
Base DN	Distinguished name for the SSH public key groups for example, CN=SSH Public Key Groups,CN=SEAS,DC=example,DC=com
Return Attributes	dn
Scope	One Level

Match Attributes	Name=sshPublicKey Value=sshPublicKey_b64
	Name=uniquemember value=FindHostDN

Select the query you created and move it to the second position in the list.

Service Principal Definitions for a Change Password Definition

Define an account with the following permissions for the service principal:

Database Used	Permission Values
Active Directory 2003	◆ Read the global domain password policy ◆ Read the userAccountControl and pwdLastSet attributes on other users' records ◆ On AIX, permission to modify other user's passwords Obtain the permissions using an administrative account or a normal user with permissions through delegation of control
Active Directory 2008	◆ Read the global domain password policy ◆ Read fine-grained password policies ◆ Read the userAccountControl, pwdLastSet, and msDS-ResultantPSO on other users' records ◆ On AIX, permission to update other user's passwords Obtain the permissions using an administrative account or a normal user with permissions through delegation of control

About Changed Password

For one hour after a password is changed, the user can be authenticated with either the old or new password and access network resources. For more information, refer to: http://support.microsoft.com/kb/906305/en-us

Change how long the old password can be used using the Registry key in the domain controller called HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa \OldPasswordAllowedPeriod. Set the value to 0 to prevent old passwords from being used to authenticate the user.

Delegation of Control

Create a service account for EA and specify the account in the Service Principal Bind Information on the Change Password settings, instead of using an administrative account. Delegate control be to this account to allow it to read other user attributes and read the fine-grained password policy in AD 2008 as well as change other users' passwords when EA is installed on AIX.

Create a Service Account

To create a service account for EA:

1. On the domain controller, be sure to log in as a domain administrator. Select **Start>Programs>Administrative Tools>Active Directory Users and Computers**.
2. Right click the user container where the service account will be added and select **New>User**.
3. Type a user ID in the **User logon name** field and a full name. Click **Next**.
4. Type a password for the account.
5. Disable the **User must change password at next logon** field.
6. Enable the **Password never expires** field.
7. Click **Finish**.

Delegate Control of the EA Service Account

To delegate control to the EA service account:

1. Right click the folder containing the users authenticated by EA (Ex: Partner Users) and select **Delegate control**. Click **Next**
2. Click **Add** on the Users or Groups page.
3. Type the EA service account name on the **Enter object names to select edit** field (Ex: seas) and click **Check Names**. The EA service account should be resolved and underlined.
4. Click **OK**.
5. Click **Next** on the Users or Groups page.
6. If EA is installed on AIX, enable **Reset user passwords and force password check at next logon**.
7. Enable **Read all user information**.
8. Click **Next**.
9. Click **Finish**.

Configure a Password Policy in Active Directory 2003 and 2008

In Active Directory 2003, the password policy is global and applies to all users of the domain. It is not possible to define password policies for individual users or groups.

To configure a password policy:

1. On the domain controller, while logged in as a domain administrator, launch the Microsoft Management Console.

2. Under Console Root, expand the Group Policy Management node. If not there, add it. Refer to as follows:
   a. Select **File>Add/Remove Snap-in**.
   b. Click **Add**.
   c. Select **Group Policy Management**.
   d. Click **Add**.
   e. Click **Close**.
   f. Click **OK**.
3. Expand the Forest node.
4. Expand Domains node.
5. Right click the node for the domain and select **Create and Link a GPO here**.
6. On the New GPO dialog, type a name for the policy and **OK**.
7. Right click the policy you created and click **Edit**.
8. On the Group Policy Object Editor window, on the tree to the left, expand **Computer Configuration > Windows Settings > Security Settings > Account Policies**.
9. Click **Password Policy**.
10. To enable each policy setting:
    a. Double-click each policy setting to the right.
    a. Enable **Define this policy setting**.
    b. Type a value for the setting.
    c. Click **OK**.
    d. Close the Group Policy Editor window.
11. Right click the new policy and click **Enforced**.
12. Click **OK** on the confirmation message box
13. Click the move up icon (single triangle icon on the left) until the new policy is positioned before the Default Domain Policy.
14. Right click the Default Domain Policy and click **Edit**.
15. On the Group Policy Object Editor window, expand **Computer Configuration>Windows Settings>Security Settings>Account Policies**.
16. Click **Password Policy**.
17. To disable each policy setting
    a. Double-click each policy setting to disable all settings:
    b. Disable **Define this policy setting**.
    c. Click **OK**
    d. After disabling all settings, close the Group Policy Editor window.

Fine Tune the Password Policy Configuration in AD 2008

The global domain password policy for Active Directory 2008 is configured the same as in Directory 2003.

Active Directory 2008 provides the ability to define fine-grained password policies that can be applied to individual users or global security groups to override the global domain password policy. Before fine-grained password policies can be defined, the domain controller functional level must be raised to Windows 2008

Raise the Domain Controller Functional Level to Windows 2008

To raise the domain controller functional level:

1. Select **Start>Programs>Administrative Tools>Active Directory Domains and Trusts**.
2. Select the domain node under Active Directory Domains and Trusts root node.
3. Right click the domain and select **Raise Domain Functional Level**.
4. Select **Windows 2008** and click **Raise**.
5. Click **OK** on the warning message.
6. Click **OK** on the confirmation message.

Create a Fine-grained Password Policy

Before you complete this procedure, the ADSI Edit node and the Domain node must exist. To check for the ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To create a fine-grained password policy:

1. While logged in as a domain administrator, launch MMC.
2. Under Console Root, expand the ADSI Edit node.
3. Expand the Default naming context node.
4. Expand the DC=domain node.
5. Expand the System node
6. Double-click CN=Password Settings Container.

   The list of currently defined policies is displayed.
7. Right click CN=Password Settings Container and select **New>Object**.
8. Select the msDS-PasswordSettings class and click **Next**.
9. Type a name for the policy and click **Next**.
10. Type values for the following fields and click **Next** after each definition:
    - msDS-PasswordSettingsPrecedence—Number greater than 0 to determine which policy to apply when a user is assigned more than one policy. Active Directory does not merge policies into one policy.
    - msDS-PasswordReversibleEncryptionEnabled—Type false.
    - msDS-PasswordHistoryLength—Type a number to define how many passwords to keep in history. When the user changes the password, a password in this list cannot be reused.

- msDS-PasswordComplexityenabled—Type true if passwords must contain characters from at least three of the following four character groups: uppercase characters, lowercase characters, numeric characters, or special characters.
- msDS-MinimumPasswordLength—Minimum characters required for passwords.
- msDS-MinimumPasswordAge—How old a password must be before it can be changed. Format is d:hh:mm:ss. For example, to specify 1 day, enter 1:00:00:00.
- msDS-MaximumPasswordAge—Maximum length of time a password can be used before it expires. Format is d:hh:mm:ss. For example, to specify 30 days, enter 30:00:00:00.
- msDS-LockoutThreshold—How many invalid login attempts can occur before the account is locked. 0 means the account is not locked.
- msDS-LockoutObsevationWindow—How long after an invalid login attempt that system invalid login attempts are tracled. This value is reset when the user logs in successfully. Format is d:hh:mm:ss. For example, to specify 5 minutes, enter 0:00:05:00.
- msDS-LockoutDuration—How long to lock an account after too many invalid login attempts. Format d:hh:mm:ss. For example, to specify 30 minutes, enter 0:00:30:00.

11. Click **Finish**.
12. Right click the policy just added and select **Properties**.
13. Click the Security tab.
14. Click **Add**.
15. On **Enter object names to select**, type the EA service account name and click **Check Names**.

    The EA service account is resolved and underlined.

25. Click **OK**.
26. Click **Apply**.
27. Click **OK**.

After a password policy is created, it can be assigned to individual users or global security groups.

Assign a Password Policy to a User or Global Security Group

Before you complete this procedure, the ADSI Edit node must exist. To check for the ADSI Edit node, open AD and expand Console Root. Look for Domain node under ADSI Edit. If these nodes do not exist, refer to *Add the ADSI Edit Node and the Domain Node* on page 63.

To assign a password policy:

1. While logged in as the domain administrator, launch MMC.
2. Under Console Root, expand the ADSI Edit node.
3. Expand the Default naming context node.
4. Expand the DC=*domain* node.
5. Expand the System node.
6. Double-click CN=Password Settings Container.

   The list of currently defined policies is displayed.

7. Right-click the policy you want to assign to a user or group and select **Properties**.
8. On the Attributes tab, select msDS-PSOAppliesTo and click **Edit**.
9. Click **Add Windows Account**.
10. On the **Enter the object names to select** field, type a user ID or global security group name and click **Check Names**.

    If the user ID or the global security group exists, the name is resolved and underlined.
11. Click **OK** twice.
12. Click **Apply**.
13. Click **OK**.

For more information about fine-grained password policies, refer to http://technet.microsoft.com/en-us/library/cc770842(WS.10).aspx.

Configure Tivoli to Allow a User to Change a Password in EA

Complete this section to configure Tivoli and allow a user to change his password in EA.

Service Principal Definitions for a Change Password Definition

Define an account with the following permissions for the service principal:

Database Used	Permission Values
IBM Tivoli Directory Server	◆ Read the global password policy ◆ Read individual and group password policies ◆ Modify userPassword attribute of other users ◆ Read the pwdChangedTime, pwdReset, ibm-pwdIndividualPolicyDN, and memberOf attributes ◆ Read the ibm-pwdGroupPolicyDN attributes from the group records Obtain permissions using an administrative account or a normal user with permissions though an ACL.

Configure an IBM Tivoli 6.x Password Policy

IBM Tivoli version 6.x provides the ability to define individual and group password policies that override the global password policy. The global password policy is similar to the global password policy for version 5.x. The main difference is that the distinguished name of the container holding the global password policy changed to "CN=PWDPOLICY, CN=IBMPOLICIES" (versus "CN=PWDPOLICY" in 5.x), and the addition of a new attribute (ibm-pwdGroupAndIndividualEnabled) that enables group and individual password policies.

Enable Group and Individual Password Policies in Tivoli Version 6.x

To enable group and individual password policies, run the following command on the directory server while logged in as root:

```
idsldapmodify -D admin DN    admin password       filename          ://host:port
```

where *filename* is a file containing the following lines:

Note: Modify ibm-pwdGroupAndIndividualEnabled attribute with the idsldapmodify command on the Directory server, while logged in as root. You cannot modify this attribute through an LDAP editor. If the -k flag is omitted, the command fails with error unwilling to perform.

Configure the Global Password Policy in Tivoli Version 6.

To configure the global password policy:

1. Use JXplorer or Apache Directory Studio to connect to the directory as the administrator.
2. Edit the CN=PWDPOLICY, CN=IBMPOLICIES container.
3. Set the ibm-pwdPolicy attribute to **true to** enable the global password policy.
4. Edit the following policy attributes to define desired policy settings:
   - pwdAllowUserChange—true allows users to change their passwords. Users must also have modify access to the userPassword attribute.
   - pwdCheckSyntax—2 enables checking password syntax for complexity such as minimum length and types of characters. 0 disables syntax checking.
   - pwdMinLength—minimum password length. If 0, no minimum length is enforced.
   - passwordMinAlphaChars—minimum alphabetic characters required in the password. If 0, password is not checked for alphabetic characters.
   - passwordMinOtherChars—minimum numeric or special characters required in the password. If 0, password is not checked for numeric or special characters.
   - passwordMaxRepeatedChars—maximum times a character can appear in the password. If 0, password is not checked for repeated characters.
   - passwordMaxConsecutiveRepeatedChars—maximum times a character can appear consecutively. 0= password is not checked for consecutive characters.
   - pwdInHistory—number of previous passwords remembered. A password cannot be reused if it is in this list. 0=passwords can be reused indefinitely.
   - passwordMinDiffChars—minimum characters that must be different from the previous password. If 0, characters in the password are not checked against the previous password.
   - pwdMaxAge—maximum password age in seconds. Password expires after it has been in use this long. If 0, password does not expire.
   - pwdMinAge—minimum password age in seconds. Password cannot be changed until it has been used for at least this amount of time. If 0, password can be changed at any time.
   - pwdMustChange—true if users must change their password after it has been reset by an administrator. If true, pwdAllowUserChange must also be set to true.

Give Users Access to LDAP Attributes in Tivoli Version 6.x

To be able to change their own passwords, users must be given modify access to the userPassword attribute and read access to the pwdLastChanged and pwdReset operational attributes. This allows EA to determine when passwords expire, or whether passwords must be changed. Users must also have read access to ibm-pwdIndividualPolicyDN and ibm-pwdGroupPolicyDN so to allow EA to retrieve the password policy.

To give users access to LDAP attributes in Tivoli:

1. Type the following command on the Directory server while logged in as root:

```
                    admin DN    admin password       filename
```

filename is the file containing the following lines.

```
group:CN=ANYBODY:(objectclass=*):normal:rsc:system:rsc:restricted:rsc:sensitive:
rsc:critical:rsc
ibm-filterAclEntry:
access-id:cn=this:(objectclass=*):at.userPassword:grant:w:at.pwdChangedTime:gran
t:r:at.pwdReset:grant:r:at.ibm-pwdIndividualPolicyDN:grant:r:at.ibm-pwdGroupPoli
cyDN:grant:r
```

2. Replace the sample dn value (ou=SEAS2.3,O=IBM,C=US) with the distinguished name of the container where users are stored.

Run this command only once for each user container.

Create a Group or Individual Password Policy in Tivoli Version 6.x

To create a group or individual password policy:

1. Use an LDAP editor (JXplorer or Apache Directory Studio) to connect to the directory as the administrator.
2. Select the CN=IBMPOLICIES container.
3. Create a new entry under the CN=IBMPOLICIES container. To create a new entry:
   a. Select the following classes:
      - container
      - ibm-pwdPolicyExt
      - pwdPolicy
      - top

   b. Type a name for the policy in the CN attribute.

   c. Type the value of the mandatory pwdAttribute attribute to userPassword
4. Add the policy attributes to override. Refer to *Configure the Global Password Policy in Tivoli Version 6.* on page 94. In addition, you can set the following attribute:

   ibm-pwdPolicy—true enables the policy.

After a policy is created, it can be associated with any user or group.

Associate a Password Policy with a User

To associate a password policy with a user:

1. Type the following command on the Directory server while logged in as root:

```
idsldapmodify -D          -w               -k -f
```

   where *filename* is the file containing the following lines:

```
dn:cn=testUser,ou=Users,ou=SEAS2.3,O=IBM,C=US
changetype:modify
add:ibm-pwdIndividualPolicyDN
ibm-pwdIndividualPolicyDN:cn=testPwdPolicy_1,cn=ibmpolicies
```

2. Replace the value of the dn (cn=testUser,ou=Users,ou=SEAS2.3,O=IBM,C=US) with the distinguished name of the user who you will assign the password policy.
3. Replace the value of the ibm-pwdIndividualPolicyDN attribute (cn=testPwdPolicy_1,cn=ibmpolicies) with the distinguished name of the password policy you are assigning to the user.

Set the ibm-pwdIndividualPolicyDN attribute by running the idsldapmodify command on the Directory server while logged in as root. You cannot set this attribute through an LDAP editor.

Only one password policy can be assigned to a user at a time. If the user is a member of a group, and the group has a policy, all attributes are merged and the most restrictive attributes are enforced.

Associate a Group Password Policy with a Group

To associate a group password policy with a group:

1. Type the following command on the Directory server while logged in as root:

```
idsldapmodify -D          -w               -k -f
```

where *filename* is the name of a file containing the following lines:

```
dn:cn=testGroup,ou=Groups,ou=SEAS2.3,O=IBM,C=US
changetype:modify
add:ibm-pwdGroupPolicyDN
ibm-pwdGroupPolicyDN:cn=testPwdPolicy_1,cn=ibmpolicies
```

2. Replace the value of dn (cn=testGroup,ou=Groups,ou=SEAS2.3,O=IBM,C=US) with the distinguished name of the group where you will assign the password policy.
3. Replace the value of the ibm-pwdGroupPolicyDN attribute (cn=testPwdPolicy_1,cn=ibmpolicies) with the distinguished name of the password policy you are assigning to the group.

Note: The ibm-pwdGroupPolicyDN attribute can only be set by running idsldapmodify on the Directory server while logged in as root. It cannot be set through an LDAP editor.

Only one password policy is assigned to a group at a time. If a group is a member of another group with its own policy, all attributes are merged and the most restrictive attributes are enforced.

For information about a group policy and how the policy is constructed in Tivoli 6.x, refer to:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.IBMDS.doc/toc.xml

Import Connect:Enterprise Users into Microsoft Active Directory

EA enables you to import Connect: Enterprise for UNIX (CEU) users into Active Directory (AD). In addition, you can update users in Active Directory and Tivoli.

Configure the CEU Import Tool

To configure the CEU import tool, update the property file to identify where the file is located and how to connect to the LDAP server. The property file is called ldapImportTool.properties.

Following is a sample file:

```
## LDAP HOST ip address
LDAP_SERVER_HOST=127.0.0.1

## LDAP SERVER PORT
LDAP_SERVER_PORT=636

## valid input : AD (Active Directory)| OTHER (Tivoli, OPEN LDAP,
## APACHE DIRECTORY, ...)
LDAP_SERVER_TYPE=AD

## LOCATION IN LDAP where users are stored
USER_BASE_DN=CN=Users,DC=SSPDomain,DC=labs

## ID of the LDAP Principal whose Credential is be used to run
## this tool
SERVICE_PRINCIPAL=CN=Administrator,CN=Users,DC=SSPDomain,DC=labs

## password of the LDAP Principal whose Credential is used to run
## this tool
SERVICE_PRINCIPAL_PASSWORD=ppppppp

## LDAP protocol : ldap|ldaps (non-secure and secured mode)
PROTOCOL=ldaps

## type of user to add to USER_BASE_DN
OBJECT_CLASSES=organizationalPerson,person,top,user
##OBJECT_CLASSES=top,person,organizationalPerson,inetOrgPerson

##common name attribute to represent users
USER_ATTRIBUTE=cn

## prefix to distinguish QA, Dev, production executing
## this tool (user created will be prefixed by this value)
USER_PREFIX=

## only apply search to users that match the pattern specified
SEARCH_FILTER=(&(objectClass=user)(sn=Oppeinhemer))
```

```
## for group operations, users will be added/removed from the
## specified group
GROUP_DN=CN=Dev Admin Group,ou=User Groups,DC=SSPDomain,DC=labs

## If overwrite is set to true, existing users or groups
## will be modified with new values
OVERWRITE=TRUE

#the attribute for group operations (uniqueMember | member) for this server
MEMBER_ATTRIBUTE=uniqueMember

## location of input file for either the users to import
## or user attributes to update
INPUT_LOCATION=C:\\seas2300-10262009\\dist\\bin\\ceunixData.xml

## when adding users, the password attribute supported by the LDAP
## server
##PASSWORD_ATTRIBUTE=userPassword
PASSWORD_ATTRIBUTE=unicodePwd

##LDAP_IMPORT_HANDLER=com.sterlingcommerce.hadrian.client.CeunixDataImport
LDAP_IMPORT_HANDLER=com.sterlingcommerce.hadrian.client.ExternalDataImport

DATA_TRANSFORM_HANDLER=com.sterlingcommerce.hadrian.client.Ops3DES
```

Configure the attributes as needed for your environment. Following are the attributes:

Attribute	Description
LDAP_SERVER_HOST	LDAP host IP address
LDAP SERVER PORT	Port to use to connect to the LDAP server
LDAP_SERVER_TYPE	Server type. Valid options: AD (Active Directory)| OTHER (Tivoli, OPEN LDAP, APACHE DIRECTORY)
USER_BASE_DN	Location in LDAP where users are stored
SERVICE_PRINCIPAL	ID of the LDAP principal whose credential runs this tool
SERVICE_PRINCIPAL_PASSWORD	LDAP principal’s password **Note:** The user is prompted for the password if this is not set.
PROTOCOL	LDAP protocol to use: ldap|ldaps (non-secure or secure)
OBJECT_CLASSES	Type of user to add to the USER_BASE_DN
USER_ATTRIBUTE	Common name attribute to represent users
USER_PREFIX	Prefix to identify the department of the user running this tool.
SEARCH_FILTER	Apply this search filter to users and apply the search to users that match the pattern specified.
GROUP_DN=CN	For group operations, users are added or removed from the specified group.

Attribute	Description
OVERWRITE	Determines if values are updated. If this attribute is set to true, users or groups are modified with new values. Valid values: TRUE or FALSE
MEMBER_ATTRIBUTE	The attribute for group operations for this server. Value values: uniqueMember | member
INPUT_LOCATION	Location of file to import or update
PASSWORD_ATTRIBUTE	Password attribute supported by LDAP, when adding a user.
LDAP_IMPORT_HANDLER	Performs an import operation. Following are handlers in the migration tool: ◆ com.sterlingcommerce.hadrian.client.CeunixDataImport imports CEU users. ◆ com.sterlingcommerce.hadrian.client.ExternalDataImport updates attributes of users.
DATA_TRANSFORM_HANDLER	Decrypts a password of a CEU user during the import process. This attributes allows you to specify a different transform handler based on how the password is encrypted and how the input is formatted.

Import Users from CEU

When importing users from CEU to Sterling Integrator, the migration tool creates an export file that contains users, passwords, and SSH keys. You then import this information into Active Directory.

Import users by adding users from a CEU Export file or obtaining information from a spreadsheet.

Add Users from the CEU Migration Tool

To import users from the migration tool:

1. From CEU, copy the ceuexport file from the $CMUHOME/migration directory to the *EA_installdir*/bin directory on EA. The file is called ceuexport_Account.xml.
2. Add the following line to the ldapImportTool.properties file:

```
LDAP_IMPORT_HANDLER=com.sterlingcommerce.hadrian.client.CeunixDataImport
```

3. Update the INPUT_LOCATION to point to the import file.
4. Import users. Refer to *Import Users from CEU* on page 101 for instructions.

Add or Update Users from a Spreadsheet

To add or update users in Active Directory from a spreadsheet:

1. Build a spreadsheet. Define user IDs and attributes to add to Active Directory.
   a. Define a row of column headings, made up of Active Directory field names. For example, the first column in the table below is CN and identifies the ID to add.
   b. Define additional columns for each field in AD where you want to import information.

   Following is a sample spreadsheet:

CN	**Company**	**Department**	**telephoneNumber**
PartnerA	Widgets Unlimited	Accounts Receivable	800-555-1111
PartnerB	Widgets Unlimited	Accounts Receivable	800-555-1112
sshkeyA	Tools and Such	Accounts Payable	800-555-2111
sshkeyB	Tools and Such	Accounts Payable	800-555-2112
sshpwdA	Sky the Limit	Operations	800-555-3111
sshpwdB	Sky the Limit	Operations	800-555-3112
sslscA	Mom and Pop	Sales	800-555-4111
sslscB	Mom and Pop	Sales	800-555-4112

2. Save the spreadsheet.
3. Save the spreadsheet again as a comment separated values file (*.csv). Following is a sample:

```
CN,Company,Department,telephoneNumber
PartnerA,Widgets Unlimited,Accounts Receivable,800-555-1111
PartnerB,Widgets Unlimited,Accounts Receivable,800-555-1112
sshkeyA,Tools and Such,Accounts Payable,800-555-2111
sshkeyB,Tools and Such,Accounts Payable,800-555-2112
sshpwdA,Sky the Limit,Operations,800-555-3111
sshpwdB,Sky the Limit,Operations,800-555-3112
sslscA,Mom and Pop,Sales,800-555-4111
sslscB,Mom and Pop,Sales,800-555-4112
```

4. Copy the .csv file into the *EA_installdir*/bin directory.
5. Update the ldapImportTool.properties file with the following information:

```
LDAP_IMPORT_HANDLER=com.sterlingcommerce.hadrian.client.ExternalDataImport
```

6. Set the INPUT_LOCATION attribute to the name of the csv file.

Import Users from CEU

After you update the property file, you are ready to import users from CEU.

To import users into Windows, type the following command in the console:

```
ldapImportTool -f ldapImportTool.properties
```

To import users into UNIX, type the following command:

```
./ldapImportTool.sh -f ldapImportTool.properties
```

The Import tool logs errors to the EA output logs in *EA_installdir*/logs/seas.log. It produces an audit log of all updates it makes to the AD database. The audit log is saved to *EA_installdir*/logs/audit/auditlog.inc.

Migrate Connect:Enterprise for UNIX Data to Sterling Integrator or Sterling File Gateway

If your data is stored on Connect:Enterprise for UNIX and you want to move it to Sterling Integrator or Sterling File Gateway, Sterling Services can quickly and efficiently migrate your data.

Contact your Sterling sale representative for more information.

Configure EA to Validate Certificates

Certificate Validation (CV) definitions specify how EA validates a trading partner certificate. A basic definition validates the activation date and certificate chain and makes sure it has not expired.

Organization of the Certificate Validation Definition

The first scenario instructs you on how to create a basic setup. After you complete a scenario, test the connection to ensure it is correctly configured. If desired, configure advanced features.

How to Use the Worksheets

Before you configure a definition, gather information on the worksheet. Accept default settings when the field is not listed in the worksheet.

After you complete the definition, test it to make sure EA can connect to SSP and the outbound server.

Prerequisites

Before you define a CV definition, configure the following:

Configure Active Directory, LDAP, or Tivoli

Configure a system-wide definition to connect to the LDAP server.

Refer to *System-Wide LDAP Connection Definition* on page 37 or *Define a System-Wide HTTP Connection Definition* on page 39

Create a Basic CV Definition

A certificate validation (CV) definition specifies how EA validates a certificate sent by a client. The definition must have the same name as the profile defined in SSP. Fill out the worksheet and complete the procedures to configure a basic CV definition and to set up SSP.

Basic CV Definition Worksheet

Before you configure a CV definition, gather the following information:

Field	Description	Value
Name	Name to assign to the CV definition.	

The basic CV definition uses the default values. It verifies that the certificate is be used after the activation date, determines that a certificate has not expired, and validates the certificate chain.

Configure a Basic CV Definition

To create a CV definition:

1. From the Certificate Validation Definitions window, click +.
2. On the General screen, specify the name.

3. Click **Next** until the Confirm screen is displayed.
4. Click **Save** and click **Close**.

Configure SSP for Certificate Validation

You configured EA for certificate validation. Next, make sure that SSP is configured.

Configure an SSP node including a policy, netmap, and adapter.

Modify the policy associated with the inbound node. On the Advanced tab, be sure to enable External Authentication Certificate Validation. Then, identify the name of the CV definition in the External Authentication Profile field.

Make sure that the EA server connection is defined.

Test your configuration to insure that SSP connects to EA and uses the CV definition.

Add Advanced Features to a Basic CV Definition

You can add the following features to the basic CV definition: define what information is required in the validation by defining LDAP attribute queries and assertions; set up access to CRLs to determine if certificates are still valid; and determine if X.509 extensions are allowed or required during validation. A CV definition can include a custom exit to validate certificates using a Java class or an operating system command. CV definitions can be used to implement certificate-based routing for Sterling Secure Proxy (SSP) by defining a special attribute query.

After a certificate validation, EA notifies the client of the status. If successful, EA can use the conversation ID in the results to access data in a user authentication from the same client.

Advanced CV Definition Worksheet

Before you configure an advanced definition, determine which of the advanced features to enable. Use the following table to identify the value to set for the features to enable:

Field	Description	Value
Name	Name assigned to the basic CV definition.	
Clock tolerance	The difference in seconds allowed between EA and the CRL clock.	
Expiration grace period	How long EA can accept a certificate after expires.Value in hours.	
Expiration warning	When to warn the user about a certificate expiration. Default=14 days.	
CRL check required	Enable a CRL check.	
Validate to Trust Anchor	Enable the ability to validate all certificates in a request including the root certificate,	
Validate using custom exits	Enable a validation from a program you write. Provide the name of the program to call.	

Field	Description	Value
Public key minimum key length	Define the key length required to validate the certificate. Value=key length.	

Add Advanced Expiration and Validation Features to a CV Definition

To add advanced features to a basic CV definition:

1. From the Certificate Validation Definitions window, double-click the definition to modify.
2. Enable advanced features for your environment. Refer to the worksheet.
3. The basic CV definition validates certificates to the root. To use a different validation, enable one or more of the following parameters:
    - Validate to Trust Anchor
    - Validate using custom exits
    - Public key minimum key length
4. Click **Next** until the Confirm screen is displayed.
5. Click **Save** and click **Close**.

Advanced Certificate Subject Verification to a CV Definition

After creating a CV definition, you can add advanced features to your defining including verifying a certificate subject or defining an attribute query to locate a directory associated with the certificate subject. Before creating a subject verification, create a basic CV definition. Refer to *Create a Basic CV Definition* on page 105.

Certificate Subject Verification Query Worksheet

To add a query to define subject verification in a CV definition, determine which of the features to enable. Use the following table to identify the f value to set:

EA Field	Description	Value
Name	Name of the attribute query definition	
Use globally defined connection	Select the connection to use to connect to LDAP or Active Directory	
Query specification	Specify the query to use for the definition: ◆ Specify query parameters—Manually define the parameters to use in the query. ◆ Specify query as URL—Provide a URL to perform the LDAP attribute query.	

EA Field	Description	Value
Query Parameters	Specify the parameters to configure to enable to LDAP connection: ◆ Base DN—Where in the directory to begin the search. ◆ Return Attributes—Attribute types to return from the entries that match. ◆ Scope—Where to start the search. Specify one of the following options. ◆ Base—Search at the level of the Base DN. This retrieves data from a known entry in the directory. Specify EA variables to represent this element. ◆ One Level-—Search only the level immediately below the Base DN. ◆ Sub Tree—Search the entire sub-tree below the Base DN. ◆ Match Attributes—Search filter used to determine which directory entries are a match. The search filter (see RFC 2254) can be very complex, but defines one or two attribute names and their expected values. You can specify EA variables to represent this element. ◆ Query Timeout—How long in minutes and seconds (format MM:SS) before the LDAP attribute query times out and processing ends.	

Add Advanced Certificate Subject Verification to a CV Definition on HTTP

After creating a CV definition, you can add the feature to verify a certificate subject. Define an attribute query to locate a directory entry associated with the certificate subject. Create a CV definition before you complete this procedure. Refer to *Create a Basic CV Definition* on page 105.

To create a query:

1. From the Certificate Validation Definitions window, double-click the CV definition to modify.
2. Click the Attribute Query Definitions tab and click +.
3. Specify a name.
4. Enable use globally defined connection and select the connection you defined.
5. Specify how to perform the LDAP attribute query. Refer to the worksheet for the value you selected.
6. Click **Next**.
7. Specify query parameters. Refer to the worksheet for parameters.
8. Click **Next**.
9. Click **Save** and click **Close**.

Reference a CRL Definition Worksheet

To add a referenced CRL definition in a CV definition, make sure a CRL definition has been created. determine which features you want to enable. To create a CRL definition, go to *Create and Manage Certificate Revocation Lists (CRL) Definitions* on page 57.

Use the following table to identify the field values to complete to reference a CRL definition:

Field	Description	Value
No field name. Selection box on the left.	CRL to enable for the CV definition.	

Reference a CRL Definition

To reference an existing CRL definition:

1. From the Certificate Validation Definitions window, double-click the definition to modify.
2. Click the Referenced CRLs tab.
3. Double-click the CRL definition in the left column to enable it.
4. Click **OK**.

Supported Extension Definition Worksheet

To add a supported extension definition to a CV definition, determine which extensions you want to add. For information on supported extensions, see *X.509 Extensions* on page 165. Use the following table to identify the extension to support and how to enable validation:

Field	Description	Value
Supported Extensions	Supported extension to associate with a CV definition and how to enable validation. ◆ KeyUsage ◆ BasicConstraints ◆ CRL DistributionPoints Set each extension to Allowed or Required.	

Configure a Supported Extension for a CV Definition

To configure a supported extension:

1. From the Certificate Validation Definitions window, double-click the definition to modify.
2. Click the Supported Extensions tab.
3. Select the extension to configure and click 📝.
4. In the **Properties** dialog, specify the value of the CA, Client, or Server extensions. Click **OK**.
5. Do one of the following:
    - To allow an extension to be in a certificate, enable **Allowed**.

- To require a certificate to have a specific extension, enable **Required**.
- Disable both **Allowed** and **Required** to reject an extension.

6. Click **OK**.

Custom Extension Definition Worksheet

EA allows you to use custom certificate extensions. Custom extensions are identified by object identifier (OID) and Name. The key values used in the extensions are configurable. For detailed information on custom extensions, see *X.509 Extensions* on page 165.

To add a custom extension definition to a CV definition, use the following table to determine which extensions you want to add and the validation to enable:

Field	Description	Value
OID	Object identifier.	
Name	Name associated with the custom extension.	
Allowed	Validation passes even if the extension is not in the certificate.	
Required	Validation fails if the extension is not in the certificate. Disable both **Allowed** and **Required** to reject an extension.	

Add a Custom Extension

To add a custom extension:

1. Double-click a CV definition.
2. Click the Custom Extensions tab and click +.
3. Type the OID (object identifier) and Name.
4. Click 📋 to display the Properties screen.
5. Specify the **Name** and **Value** of the CA, Server, and Client extensions and click **OK**.
6. Do one of the following:
   - To allow an extension to be in a certificate, enable **Allowed**.
   - To require a certificate to have a specific extension, enable **Required**.
   - Disable both **Allowed** and **Required** to reject an extension.
7. Click **OK**.

Configure and Test a Custom Exit for a CV Definition

EA allows you to use a Java class or operating system command to implement a custom exit from a CV definition. If you use a Java class as a custom exit, the class must implement the Sterling-provided interface, SEASCustomExitInterface.

Prerequisites for Using a Custom Exit

Before you configure a custom exit, perform the following prerequisite tasks:

When a CV definition includes a custom exit to a script or program, create the functionality required by writing the code that runs from the operating system command line.

Review the files in the /doc and /samples directories before you develop a Java class for a custom exit.

When a CV definition includes a custom exit to a Java class, create the functionality required for the exit by writing a Java class that performs the required CV steps.

Copy class files or a .jar file for a Java class custom exit to the *install_dir*/lib/custom directory, where *install_dir* is the directory used for CV installation.

When you specify a custom exit for a CV definition, it is also helpful to set logging to an appropriate level (such as DEBUG or ALL) to enable you to review processing results of the Java class or script or program that implements your custom exit.

Develop and Deploy a Custom Exit Class in Java

The SEASCustomExitInterface interface and a sample class implementing the interface are documented in the javadoc located in the *install_dir*/doc directory and can be found in the archive, *install_dir*/lib/sterling/custom-exit.jar. The source for the sample implementation can be found at *install_dir*/samples/SampleCertValidationExit.java.

The interface provides an initialization method that accepts a list of custom properties you define for your class. You specify these properties (names and values) from the GUI as part of the Custom Exit configuration in the CV definition.

Compile exit classes and provide them in a jar file, or as class files with package structure preserved, in the *install_dir*/lib/custom directory. The custom exit class loader searches all jar files and packages for the custom exit class name specified in the CV definition.

Specify Java Class in a CV Definition

To specify the Java class for a custom exit in a CV definition:

1. From the Certificate Validation Definitions window, select the CV definition and click 📋.
2. Click the **General** tab on the Certificate Validation Definition Properties screen. Click **Validate using custom exits**.
3. Click ... and select **Java class**.
4. Specify the fully-qualified class name in the format *packageName.className* when you specify the custom exit that implements SEASCustomExitInterface.
5. Click ... next to **Properties**.
6. Type a name and value for each property required to initialize the custom exit class. Use + and − to add or remove rows of name and value pairs. Click **OK**.
7. Click **OK**.
8. Review the log to determine if the certificate was validated successfully by the custom exit.

Specify an Operating System Command for a Custom Exit

To specify the operating system command to use for the custom exit:

1. From the Certificate Validation Definitions window, select the CV definition and click 📝.
2. Click the **General** tab. Click **Validate using custom exits**.
3. Click ... to display the **Custom Exits** dialog box and select **Native OS command** to validate a certificate using a native operating system command as a custom exit.
4. Type the operating system command to use, including all command line arguments, in the **Command line** field.
5. Specify the method to use to pass the certificate chain to the operating system command.
   - To pass the certificate chain as a certificate file:
     a. Select **Certificate file**.
     b. Type the file name for the certificate chain or a valid variable expression (such as the default, cert{counter}.pem).
     c. Specify the certificate chain file format as **PEM** or **DER**.
     d. Select **Delete file after exit** to remove the certificate file after custom exit processing.

     Tip: The default file name uses a counter to ensure a unique file name. The variable {counter} begins with 0 and increments after each exit, resulting in the following file names: cert0.pem, cert1.pem, cert2.pem, and so on. The file name can be passed on the command line as the variable {filename}. For example, the following command is a valid use of the file name:

     ```
     openssl x509 -in {filename}
     ```

   - To pass the certificate chain through the standard input stream, click **Standard input (PEM format)**.
6. Specify the timing for running the custom exit and performing certificate validation as configured in the certificate validation definition:
   - Select **Run default validator after exit** to continue processing the CV definition after the custom exit.
   - Select **Run custom exit synchronously** to enable synchronous use of this custom exit. That is, if a client application sends a certificate validation request with a reference to a definition including the custom exit and the exit is currently running, then current exit processing must complete before a subsequent invocation can run.
7. Specify **standard error log level** and **standard output log level** to control how output from the custom exit is logged. All error output is logged in SEAS.log. All standard (console) output is logged in the SEAS.log.
8. Set the log levels required to meet your needs.
9. To redirect standard error and output from the custom exit to the response message that EA returns to the client, select one or both of the following parameters:
   - **Log output from stderr to response message**—send error log to the response message.
   - Select **Log output from stdout to response message—**send log output to the message.

10. Click **OK**.

Review the log to determine if the certificate was validated successfully by the custom exit.

Edit or Copy a CV Definition

To edit or copy a CV definition:

1. To edit a CV definition, double-click the definition.
2. To copy a definition, select the definition to copy and click ⧉. Specify a new name.
3. Edit parameters as needed. Refer to *Create a Basic CV Definition* on page 105.
4. Click **Next**.
5. Click **Save** and click **Close**.

Delete a CV Definition

Deleting a certificate validation definition deletes all parameters and definitions for the definition and invalidates all references to it from a client application.

To delete a certificate validation definition:

1. From the Certificate Validation Definitions window, select the definition and click −.
2. Click **OK**.

Edit Supported Extensions

Delete a Supported Extension

Edit Supported Extensions

Supported extensions can only be edited from the definition where they are used.

To edit a supported extensions:

1. From the Certificate Validation Definitions list, double-click the definition to modify.
2. Click the **Supported Extensions** tab, select the extension to edit, and click ✎.
3. Change the extensions as required, or click **Restore Defaults** and click **OK**.
4. Modify how each extension can be used in a certificate, as required.
5. Click **OK**.

Delete a Supported Extension

To delete a supported extension:

1. From the Certificate Validation Definitions list, double-click the certificate validation with the supported extension to delete.
2. Click the **Supported Extensions** tab.
3. Select the extension to delete and click −.
4. Click **OK**.

Edit Custom Extensions in a CV Definition

To edit a custom extension:

1. Double-click the certificate validation to modify.
2. Click the **Custom Extensions** tab, select the custom extension, and click 📝.
3. Modify the values as required.
4. Click **OK**.

Delete a Custom Extension in a CV Definition

To delete a custom extension:

1. Double-click the certificate validation to modify.
2. Click the **Custom Extensions** tab.
3. Select the custom extension to delete and click −.
4. Click **OK**.

Create an RSA Authentication Definition

EA supports RSA SecurID authentication. To configure basic RSA authentication in EA, complete the tasks in *Configure EA Support for RSA SecurID* on page 115 and *Create an RSA Authentication Definition* on page 115.

Configure EA Support for RSA SecurID

To configure basic RSA SecurID support in EA, complete the following tasks:

1. Contact your RSA server administrator to get a copy of the following files:
   - sdconf.rec—This file is required. It contains the information that allows an RSA SecurID Host Agent (EA) to communicate with the RSA SecurID server. Usually, the RSA server administrator will need the IP address of the Host Agent(s) to generate this file.
   - sdopts.rec—This file is optional but may be needed for proper connectivity between the RSA SecurID Host Agent (EA) and the RSA SecurID server. The “server options file” specifies the preferred RSA SecurID server for the RSA SecurID Agent (EA) to use during communications because multiple RSA SecurID servers may be used within an environment.
2. After you install EA, copy the sdconf.rec and sdopts.rec files to the *install_dir*/conf/jaas directory.
3. Create an RSA SecurID authentication definition using EA. For more information, see *Create an RSA Authentication Definition* on page 115.
4. In Sterling Secure Proxy, specify the RSA SecurID profile you created in EA in the appropriate policy.
5. The first time a successful connection is made to the RSA SecurID Server, a file named secureid is generated in the {EA_INSTALL}/conf/jaas directory. This file must be presented during subsequent connections and authentication requests. Make a copy of this file and store it in a safe place. If you lose this file, your RSA SecurID administrator will have to reset it.

Create an RSA Authentication Definition

Authentication definitions specify how EA authenticates a security principal when a client application sends a request. To enable EA for RSA SecurID support, create an RSA user authentication definition.

To create an RSA SecurID authentication definition:

1. From the Authentication Definitions window, click + to add an authentication definition.
2. On the LDAP Authentication screen, type a Profile Name.
3. Select RSA SecurID as the Authentication type.
4. Click **Next**.
5. Click **Next** twice.
6. Click **Save**.

RSA in SSP Worksheet

In Sterling Secure Proxy, specify the name of the RSA SecurID profile you created in the appropriate policy. Use the table below to identify the values to assign in the EA RSA authentication definition.

EA Field	Value to Assign
Profile name	Name for the profile
Authentication type	RSA SecurID

Edit or Copy an Authentication Definition

To copy and edit an authentication definition:

1. From the Authentication Definitions window, perform one of the following actions:
   - To copy an authentication definition, select the definition to copy and click ⧉.
   - To edit an authentication definition, double-click the definition to edit.
2. Type a unique **Profile Name** if you are copying an authentication definition.
3. Click **OK**.

Delete an Authentication Definition

To delete an authentication definitions:

1. From the Authentication Definitions window, select the authentication definition to delete and click −.
2. Click **OK**.

Configure JAAS Authentication Definitions

To configure the EA environment for JAAS, modify the default JAAS configuration and RSA SecurID properties files located in the ${EA_INSTALL}conf/jaas directory.

The JAAS configuration file called seas_default_jaas.config specifies the authentication modules supported by EA. The EA installation updates the JAAS configuration file with your environment variables. The contents of this configuration file is shown below:

```
SeasRSALoginModule {
  com.sterlingcommerce.component.authentication.impl.SeasSecurIDLoginModule required
debug=true
   properties=".../conf/jaas/secureid.properties";

};

SeasRSALDAPLoginModule {
   com.sterlingcommerce.component.authentication.impl.SeasSecurIDLoginModule
sufficient debug=true
   properties=".../conf/jaas/secureid.properties";

    com.sterlingcommerce.component.authentication.impl.SeasLdapLoginModule sufficient
        userProvider="ldaps://ldap_host:{LDAP_PORT}/ cn={USERNAME}
CN=Users,DC=SSPDomain,DC=labs"
        authIdentity="cn={USERNAME},CN=Users,DC=SSPDomain,DC=labs"
  useSSL=false
  debug=true;
};

SeasLDAPLoginModule {
    com.sterlingcommerce.component.authentication.impl.SeasLdapLoginModule sufficient
        userProvider="ldaps://ldap_host:{LDAP_PORT}/CN=Users,DC=SSPDomain,DC=labs"
        authIdentity="cn={USERNAME},CN=Users,DC=SSPDomain,DC=labs"
  useSSL=false
  debug=true;
};
```

Edit the JAAS configuration file, seas_default_jaas.config and make your edits. Then save it.

Modify the JAAS Configuration File for RSA SecurID

To implement a JAAS RSA SecurID or RSA SecurID with LDAP fallback authentication scheme, update the seas_default_jaas.config file with the following attributes needed for your environment:

Attribute	Description
com.sterlingcommerce.component.authentication.impl.SeasSecurIDLoginModule	Fully-qualified class path of the SeasSecurIDLoginModule. The file contains the class path of the EA RSA SecurID module.
properties	Location of the securid.properties file that implements RSA SecurID. The default is {SEAS_INSTALL}/conf/jaas. If you move the file, update the variable to provide its fully-qualified path.

Modify the JAAS Configuration File for LDAP

To implement a JAAS LDAP or LDAP fallback authentication scheme, update the seas_default_jaas.config file with the following attributes for your environment:

Attribute	Description
com.sterlingcommerce.component.authentication.impl. SeasSecurIDLoginModule	Fully-qualified class path of the SeasLDAPLoginModule. The file contains the class path of the EA LDAP module. If you want to implement your own LDAP module, replace this class path with your own. See *Implement Your Own LDAP Module in EA* on page 118 for more information.
debug	Enables debug mode for this module. True—Debug is enabled. False—Debug is disabled. This attribute is not required for proper initialization of the LDAP login module.
userProvider	Points to the LDAP server that may be contacted by the SeasLDAPLoginModule. The base DN must be specified. Specify either ldap or ldaps. If you specify ldaps to run the LDAP login module in secure mode, update the truststore file located in {SEAS_INSTALL}/conf/system/truststore with the location of the LDAP server's public certificate. {LDAP_PORT}—Replace this variable with the port number of the LDAP host.
authIdentity	Location of the users in the specified LDAP server. Do not replace the {USERNAME} variable in the seas_default_jaas.config file.This variable will be substituted with a user by the SeasLDAPLoginModule when a request to authenticate a specific user is made.
useSSL	Enables the secure mode. True—The module will run in secure mode. False—The module will run in nonsecure mode.

Implement Your Own LDAP Module in EA

EA provides the ability implement your own LDAP module.

To implement your own LDAP module:

1. Copy the jar file that contains your LDAP module to *EA_install_dir*/lib/custom. The jar file must contain the class name you will specify in the seas_default_jaas.config file.
2. In the seas_default_jaas.config file, replace the following line with your own class name:

```
com.sterlingcommerce.component.authentication.impl.SeasLdapLoginModule
```

3. Save the seas_default_jaas.config file.

Modify the RSA SecurID Properties File

The SecurID properties file specifies information necessary to communicate with the SecurID server and the EA host that will communicate with that server. During EA installation, environment variables are updated in the RSA SecurID properties file, secureid.properties. All other attributes are optional. To update this file, open it in a text editor, make your changes, and save the file.

A sample RSA SecurID properties file is shown below:

```
# RSA Authentication API Properties
# Override Host IP Address (SEAS Host Machine IP address)
RSA_AGENT_HOST=seas.host.machine
# Interval in seconds between which configuration is refreshed.
RSA_CONFIG_READ_INTERVAL=600
# [This section is for Data Repository configuration.]
# Type of the Server configuration.
SDCONF_TYPE=FILE
# Path of the Server configuration.
SDCONF_LOC=C:\\development\\seas-03162009\\dist\\conf\\sdconf.rec
# Type of the Server statuses.
SDSTATUS_TYPE=FILE
# Path of the Server statuses.
#SDSTATUS_LOC=C:\\development\\seas-03162009\\dist\\logs\\JAStatus.1
#SDSTATUS_LOC=

# Type of the Server options.
SDOPTS_TYPE=FILE
# Path of the Server options.
SDOPTS_LOC=C:\\development\\seas-03162009\\dist\\conf\\sdopts.rec
# Type of the Node Secret.
SDNDSCRT_TYPE=FILE
# Path of the Node Secret.
SDNDSCRT_LOC=

# Logs event messages to a file.
RSA_LOG_TO_FILE=YES
# Name of the log file.
RSA_LOG_FILE=C:\\development\\seas-03162009\\dist\\logs\\rsa_jaas.log
# Minimum severity level allowed to log.
RSA_LOG_LEVEL=DEBUG

# [This section is for debugger.]
# Enables debug tracing.
RSA_ENABLE_DEBUG=YES
# Sends tracing to the console.
RSA_DEBUG_TO_CONSOLE=NO
# Sends tracing to a file.
RSA_DEBUG_TO_FILE=YES
# Name of the trace file.
RSA_DEBUG_FILE=C:\\development\\seas-03162009\\dist\\logs\\rsa_jaas_debug.log
# Allows function entry tracing.
RSA_DEBUG_ENTRY=YES
# Allows function exit tracing.
RSA_DEBUG_EXIT=YES
# Allows control flow tracing.
RSA_DEBUG_FLOW=YES
# Allows regular tracing.
RSA_DEBUG_NORMAL=YES
# Traces the location.
RSA_DEBUG_LOCATION=YES
```

Create JAAS Authentication Definitions

Authentication definitions specify how EA authenticates a security principal when a client application sends a request. To enable EA for JAAS support, create a JAAS user authentication definition.

To create a JAAS authentication definition:

1. From the Authentication Definitions window, click +.
2. On the LDAP Authentication screen, type a Profile Name.
3. Select JAAS as the Authentication type.
4. Select a JAAS Module Name.
5. Click **Next**.
6. Click **Next** twice.
7. Click **Save**.

In Sterling Secure Proxy, specify the name of the JAAS profile you created as the EA Profile in the appropriate policy.

Edit, Copy, or Delete a JAAS Authentication Definition

To copy or edit a JAAS authentication definition:

1. From the Authentication Definitions window, perform one of the following actions:
   - To copy an authentication definition, select the definition to copy and click ⧉. Type a unique **Profile Name**.
   - To edit an authentication definition, double-click the definition to edit. Modify the fields as necessary.
   - To delete an authentication definition, select the definition and click −.
2. Click **OK**.

Perform Sterling Integrator (SI) User Authentication through an EA Custom Exit

EA provides a custom user authentication exit to validate a trading partner user ID and password against the Sterling Integrator user store. Refer to *Create Generic Authentication Definitions* on page 137 for instructions.

Before you use this custom exit to validate user information against the Sterling Integrator user store, you must configure a separate HTTP server adapter in Sterling Integrator to enable both user authentication and SSL, and to invoke a do-nothing business process called HelloWorld.

Prepare Certificates for Authentication in the Sterling Integrator User Store

To prepare to authenticate user IDs and passwords in Sterling Secure Proxy using the Sterling Integrator user store, you must prepare certificates by performing the following tasks:

Configure the HTTP Server Adapter Certificate

Export the System Certificate from Sterling Integrator

Import the HTTP Server Adapter System Certificate into the EA Trust Store

Export a Keystore from the EA Keystore

Import the EA System Certificate into the Sterling Integrator CA Certificate Store

Configure the HTTP Server Adapter Certificate

Decide which system certificate to use for the HTTP server adapter. Use the default certificates provided by Sterling Integrator, or import your own. For security reasons, use your own certificates.

Note: PEM key certificates must have a txt extension. If your certificate has a different extension, rename it to txt. PKCS12 certificates must have a pfx extension. If necessary, rename the PKCS12 certificate to *certificatename*.pfx.

To import a system certificate into the Sterling Integrator certificate store:

1. On the Sterling Integrator dashboard, select Trading Partners > Digital Certificates > System.
2. Click **Go!** in the check in section for the key you are checking in: PEM or PKCS12 certificate.
3. Specify the location of the certificate file and the password for the private key and click **Next**.
4. Click **Next**.
5. Click **Finish**.

Export the System Certificate from Sterling Integrator

After you identify the system certificate to use for the HTTP server adapter, export the public part of the certificate. After it is exported, it will be imported into the EA trust store.

To export the system certificate:

1. On the Sterling Integrator dashboard, select Trading Partners >Digital Certificates > System.

2. Do one of the following:
    - Type the system certificate name in the Search by certificate name field and click **Go**.
    - Click **Go** on the List section to get a list of all certificates and locate the desired certificate.
3. On the System Certificates screen, click the checkout button next to the certificate to export.
4. Select BASE64, then click **Go**.
5. Click **Save** and select the location where you want to save the exported certificate.

Import the HTTP Server Adapter System Certificate into the EA Trust Store

Add the exported certificate to the EA trust store, located in the conf/system/truststore folder. To import the system certificate into the EA trust store, navigate to the *install_dir*/jre/bin directory on the computer where the EA server resides, type the following command, and press **Enter**.

```
keytool -import -keystore truststore_path -alias alias_name -storepass password -file
certificate
```

Following is a description of the keytool parameters used to import an EA certificate:

Parameter	Description
-import	Instructs keytool to import a certificate into the keystore.
-keystore truststore_path	The path and file name of the truststore file.
-alias alias_name	The alias name to identify the certificate in the keystore. Use the same alias as you used to create the certificate.
-storepass password	The password of the keystore file.
-file certificate	The location of the certificate for EA to import.

Export a Keystore from the EA Keystore

To allow the HTTP server adapter to trust the client certificate from EA, the client certificate must be exported from the EA keystore and then imported into the Sterling Integrator CA certificate store. The EA keystore is located in the conf/system/keystore directory, by default.

To export the client certificate from the EA keystore, navigate to the *install_dir*/jre/bin directory on the computer where the server resides, type the following command, and press **Enter**.

```
keytool -export -alias alias_name -keystore keystore_path -storepass password -rfc -file cert_file_name.
```

Following is a description of the keytool parameters used to export an EA certificate:

Parameter	Description
-export	Instructs keytool to export a certificate from the EA keystore.
-keystore keystore_path	The path and file name of the keystore file.

Parameter	Description
-alias alias_name	The alias of the EA client certificate in the keystore. Use the same alias you used to create the certificate.
-storepass password	The password of the keystore file.
-rfc	Exports the certificate in PEM format. To export the certificate in DER format, do not include the –rfc parameter.
-file certificate	The location of the certificate for EA to import.

Import the EA System Certificate into the Sterling Integrator CA Certificate Store

After the EA client certificate is exported, import it into the Sterling Integrator CA certificate store.

To import the certificate into the Sterling Integrator CA certificate store:

1. On the Sterling Integrator dashboard, select Trading Partners > Digital Certificates > CA.
2. Click **Go!** on the Check in section.
3. Specify the certificate file and click **Next**.
4. Type a name for the certificate and click **Next**.
5. Click **Finish** on the Confirm screen.

Configure a Sterling Integrator HTTP Server Adapter for EA Support

To configure a Sterling Integrator server adapter to for user authentication through a custom exit:

1. On the Sterling Integrator dashboard, select Deployment > Services >Configuration
2. Click **Go!** on the Create New Service panel.
3. Select HTTP Server Adapter as the service type and click **Next**.
4. Type a name and description for the adapter and click **Next**.
5. Specify a listen port, perimeter server, and queue depth (max concurrent sessions). Select **Yes** for User Authentication Required. Select **Must** for Use SSL. Click **Next**.
6. Select the server certificate that the HTTP server adapter will present to EA from the System Certificate combo box.

   Refer to *Configure the HTTP Server Adapter Certificate* on page 123 for information.
7. Select the CA certificate to use to validate the EA certificate from the CA Certificates list, and click the arrow button to move it to the list on the right.

   See *Import the EA System Certificate into the Sterling Integrator CA Certificate Store* on page 125 for information. If no certificate is selected, client authentication is disabled.
8. Press **Next**.
9. Click **+** to add a new URI.
10. On the URI field, specify a URI name *starting with a slash* (for example: /gisAuth). If the leading slash is missing, an error is returned to clients trying to access the URL. Click **Next**.
11. On the Business Process combo box, select HelloWorld and click **Next**.

12. Click **Next** on the URI page.
13. Click **Finish**.

Configure an EA User Authentication Profile

To configure an EA user authentication profile to support Sterling Integrator user authentication:

1. Launch the EA user interface and login.
2. On the Authentication Definitions window, click +.
3. On the Authentication type combo box, select **Generic**.
4. Type a profile name.
5. Click the **Authenticate using custom exits** check box and press the **…** button.
6. On the class name field, specify com.sterlingcommerce.component.authentication.impl.HttpUserAuthExit.
7. Click the **…** button next to Properties.
8. Add the following property: url = <fully-qualified URL for HTTP server adapter>

   For example: https://dev-blade2:11080/gisAuth
9. Click **OK**.
10. Click **Next** to move through the definition pages.
11. Click **Save**.

Custom Exit Configuration Properties

Following is a description of the configuration properties:

Name	Value
url	Fully-qualified URL for the primary HTTP Server Adapter. The format is:<protocol>://<host>:<port>/<uri>. This property is required.
alt.url.1	Fully-qualified URL for the first alternate HTTP Server Adapter. If the connection to the primary adapter fails, the first alternate is tried next. This property is optional.
alt.url.2	Fully-qualified URL for the second alternate HTTP Server Adapter. If the connection to the first alternate adapter fails, the second alternate is tried next. This property is optional.
alt.url.3	Fully-qualified URL for the third alternate HTTP Server Adapter. If the connection to the second alternate adapter fails, the third alternate is tried next. This property is optional.
bind.addr	IP address of NIC to use for outbound connection. Used with systems with more than one NIC. This property is optional.
client.alias	Alias of client certificate to use for outbound SSL connection. Used only if the EA keystore has more than one key certificate. This property is optional.

Log Messages

Following are the log messages written in the secureproxy log file, EA log file, and Sterling Integrator log file.

Sterling Secure Proxy Messages

Following are the SSP messages written to the secureproxy log file:

Message	Sample
Success Authentication	08 Aug 2011 13:08:40,548 INFO [ProxyNearScheduler-Thread-6] sys.ADAPTER.httpAdapter - SSE1827I Engine Name=iikonne1, Adapter Name=httpAdapter, EA Name=eaServer. Received user authentication response from EA server. Client: null Profile: gisAuth User: admin Message: AUTH073I admin successfully authenticated
Failed Authentication	08 Aug 2011 13:12:14,042 INFO [ProxyNearScheduler-Thread-5] sys.ADAPTER.httpAdapter - SSE1827I Engine Name=iikonne1, Adapter Name=httpAdapter, EA Name=eaServer. Received user authentication response from EA server. Client: null Profile: gisAuth User: admin Message: AUTH074E Authentication failed for admin. Exception encountered during custom exit: AUTH071E Authentication failed for admin (Reason: invalid userid/password).

EA Server Messages

Following are the EA server messages written to the EA server log file:

Message	Description
Success Authentication	08 Aug 2011 13:08:41,986 730209 [Pool Worker - 4] INFO com.sterlingcommerce.component.authentication.impl.CommonAuthenticator - AUTH073I iikon1 successfully authenticated.
Failed Authentication	F08 Aug 2011 13:12:14,027 942250 [Pool Worker - 5] ERROR com.sterlingcommerce.component.authentication.impl.HttpUserAuthExit - java.lang.Exception: AUTH071E Authentication failed for admin (Reason: invalid userid/password).

SI Authentication Log Messages

Following are the messages written to the Sterling Integrator log file:

Message	Sample
Failed Authentication	[2011-08-14 13:02:32.931] DEBUG 000000000000 GLOBAL_SCOPE SecurityManager user:user1 attempting to log in (SSO:false) [2011-08-14 13:02:32.931] DEBUG 000000000000 GLOBAL_SCOPE GISAuthentication user:user1 is identified as a LOCAL GIS User [2008-08-14 13:02:32.931] ALL 000000000000 GLOBAL_SCOPE SecurityManager user:user1 authorization FAILED (SSO:false)
Successful Authentication	[2011-08-14 13:03:35.9] DEBUG 000000000000 GLOBAL_SCOPE SecurityManager user:gvega attempting to log in (SSO:false) [2011-08-14 13:03:35.9] DEBUG 000000000000 GLOBAL_SCOPE GISAuthentication user:gvega is identified as a LOCAL GIS User [2011-08-14 13:03:35.9] ALL 000000000000 GLOBAL_SCOPE SecurityManager user:user1 authorization SUCCEEDED (SSO:false)

Create an SSH Key Authentication Definition

An SSH key authentication definition specifies how EA authenticates an SSH user when a client sends a request for authentication.

A client such as SSP sends a request with a profile name, user ID, and SSH public key. EA binds to an LDAP directory, looks up SSH keys, and matches the key to keys stored at the LDAP server.

> The credentials to bind to the directory are in the SSH key authentication. The user ID cannot bind to the directory because the password is not available. The administrator credentials are used to bind to the directory and are configured in an LDAP connection definition.
>
> The query to look up SSH keys is based on your directory layout. If you use the openssh schema, the query returns all sshPublicKey attributes for the user. If you use a customized schema, modify the query to make sure it returns the attributes associated with the schema.
>
> An assertion definition matches the public key against keys from the SSH public key lookup query. The pre-configured assertion included with EA uses the openssh schema to store the public keys. If you use a different schema, edit the assertion to use the appropriate schema.
>
> To use SSH key mapping, define another query to return a reference to the mapped key. The MapSSHCredentials query returns the routingKeyName of the loginCredentials record, and assigns it to the mappedRoutingKeyName output. The application uses the mappedRoutingKeyName output to locate mapped key for the user.

Prepare OpenLDAP or IBM Tivoli to Store User Credentials for an SSH User

Before performing user authentication and login credentials mapping, update the directory schema.

Use the files provided to extend the schema for OpenLDAP and IBM Tivoli Directory Server directories, or as reference when you manually extend other directories. If you use these directory extensions and create directory entries, the application output definition is created automatically. Alternatively, use any arbitrary directory object that stores a user ID and password, as long as the password can be retrieved as unencrypted data. When using an arbitrary directory object, define the attribute query that stores a user ID and password and the attribute query that retrieves the credentials for the application output definition. Then, use controls on the Application Output Definition to manually map attributes returned by the query to outputs that a client can access.

If you implement a custom SSH schema you do not configure the custom SSH schema provided with EA. You create or edit an SSH Key Authentication profile and identify the attributes defined in the custom schema, when defining queries and assertions.

Implement the SSH and SCI Schemas for Open LDAP

To implement the SSH and SCI schema for Open LDAP:

1. Copy the openssh-lpk.openldap.schema file from the *install_dir*/schema/ to the schema subdirectory of OpenLDAP.
2. Copy the sci.schema file to the schema subdirectory of the OpenLDAP installation.
3. Edit the slapd.conf file located in the /etc/openldap directory and add include statements with the added schema references.

```
include /etc/openldap/schema/sci.schema
include /etc/openldap/schema/openssh_lpk.openldap.schema
```

4. Restart the LDAP server.

Implement the SSH and SCI Schemas for IBM Tivoli

To implement the SSH and SCI schema for IBM Tivoli:

1. Copy the v3.openssh-lpk file from the *install_dir*/schema/ to the schema subdirectory of Tivoli. Schemas are often located in the /usr/ldap/etc directory.
2. Copy the file *install_dir*/schema/V3.sci to the schema subdirectory of Tivoli.
3. Restart the LDAP server.

Create Entries for SSH Public Keys in the LDAP Server

For each SSH key, define an sshPublicKey attribute to the public SSH key for the user. If a user has multiple public keys, define an attribute for each key.

Note: The data of the sshPublicKey attribute must be in PEM format. and be cleared of BEGIN/END comments and newlines. The key should be on one line.

Following is a sample of an LDIF file for a user entry that uses the openssh-lpk.openldap.schema file and contains two SSH public keys:

```
dn: cn=guser,ou=users,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: ldapPublicKey
objectClass: top
cn: guser
sn: userLast
sshPublicKey:: c3NoLXJzYSBBQUFBQjNOemFDMXljMkVBQUFBQkl3QUFBSUVBbkRUN09VYWROZmNXdH
pzV0QveFIzWXBYd2VmS3FLbVhaQnRsenlIWVRXTjhoOXZtaHdiY1NlNWVtYWZFeVh1eGJr eXBHRDFMK
0Y1aStVbUZadE1nSUtyblIwQ1hZazhwYmlxeXBSc1J4OXBEQWR5QzRrekZaTEJnQzR 2R3NibjRHTStTZUN
XTVA0Zy9oazRGNFRvWWx6Y0VENTBnaDgzTXVwc1dhOWZaRko4PSBxYXRlc3RAcWFzbGVzOAo=
sshPublicKey:: c3NoLWRzcyBBQUFBQjNOemFDMWtjM01BQUFDQkFLU3gyRGoyRmgyZjY5b0hNU2o2UFo
va3U2ZUJoZlA1enE5UHhUeHBadExXWjlxNFh6NWtkOVFmdzFuZTVNbDhhOHFBSmN2YmFwQStBRG50U2J
0bHZQVFh5MXdObnB2OTUxRjFaYUlMd0ZIejBLUzkxUGJ1aE5ZOE9JbEdJTEY1Q0JraWc2aFFPMXBu
SFJWRlVMMEx0a3lodnI0eG5CYTdqTmtKSm1hQUJpZkJBQUFBRlFEWDc4RlhVdDJpN052QjN4aTRXdG1NbUZ6
OEZ3QUFBSUFIT1JuUE5sdC9qa25mTW4zZWtlQ3ZHbEVrZjdEQlhIRlE4UGdEcmNpNWh0US9NekpjR0tCb2FX
RUVNQnNGLzBrVVlCdjZkWVZwZTR2dVM5VmZnRzFDV0lvMjV6N1BDM2FvQ1lmK0VGUXFReWtuL1BFV1M1UUU1
NlB6S29ueXBMa3ZLdFFkS3VtbVNFSFBCR1owbUVWT2lEbjhsdTZBb1Z0L28rMmZXZXkvSlFBQUFJQXdoMHJX
RU5UMXVFZFUxV1hOL2hBdmcrTkVlVy94SnkvQUpXeTNrMGxLajM4MVdnekdiODRneTFDL2FMam40bWo4Q29u
blhPeHVxZnBiL3Q4Q0c1U2xUVlUwaUUxYWpDR0o2ODNVT20wc2xNeTl3S1hYU3BJcWdnU25zTnJaQjJ6Y0lI
S29NTDNITHF4WEF4RXZnMndhaTZReHBGd1d3Q0UwOVM4eHBwbm4zdz09IHFhdGVzdEBxYXJoYXMyMQo=
userPassword:: e1NIQX1rZC9aM2JRWml2L0Z3WlROak9iVE9QM2tjT0k9
```

Create Login Credentials in the LDAP Server

After you add the schema to the directory, create loginCredentials to for the user ID, password, and SSH key. The supported directory structure creates separate loginCredentials entries in the authenticated user's directory entry for each destination service.

Set the loginId and logingPwd attributes to the ID and password needed to login to the destination service. Enter the password in binary text. Set the routingKeyName attribute to the label that maps to the public/private key pair that is needed to login to the destination service. Set the attribute called loginTarget to the destination service name defined in the authentication request from the client application. After this is defined, the query to obtain the credentials is pre-populated with the correct values and the mapping to outputs is performed automatically.

LDIF Entry Example

The LDIF entry below demonstrates an entry in the supported structure:

```
dn: cn=SSP1-App2 Login Credentials, cn=User010101, ou=SterlingEAS 2.0 Users,
dc=Sterling Commerce, dc=com
objectClass: loginCredentials
objectClass: top
cn: SSP1-App2 Login Credentials
description: User010101's login credentials for SSP1-App2 (loginPwd=loginPwd2)
loginId: loginId010101
loginPwd:: cGFzc3dvcmQ=
loginTarget: SSP1-App2
routingKeyName : internaKey
```

In the scenario above, EA authenticates the user, User010101, by binding to the following DN: cn=User010101, ou=SterlingEAS 2.0 Users, dc=Sterling Commerce, dc=com. Using the directory information tree structured in the example, a client sends an authentication request that references a destination service, SSP1-App2. The user ID to log in to this service is loginId010101 and the password is loginPwd2. EA queries the loginCredentials entry and returns the user ID, password, and routing key name to the client in the authentication response.

Note: The value of the loginPwd attribute is base64-encoded. OpenSSL provides a tool to base64-encode a password using the following command line syntax:

```
openssl enc -a -e -in clearTextPasswordFile -out base64EncodedPasswordFile
```

SSH Key Authentication Definition

Create an SSH key authentication to validate an SSH user. Select the assertion definition to use. It matches the public key from the request against keys returned by the lookup query. An assertion called VerifySSHPublicKey is provided with EA. It uses the openssh schema to store public keys. Use this definition or define your own. If you do not use the openssh schema, edit the assertion definition to reference the schema used.

Organization of the SSH Key Authentication Definition

The section instructs you how to create a basic setup.

How to Use the Worksheets

Before you configure a definition, gather information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

After you complete the definition, test it to make sure EA can connect to SSP and the outbound server. Continue to the next section to add more features.

Prerequisites

Before you create an SSH key authentication definition, define a global connection setting for the LDAP server. Refer to *System-Wide LDAP Connection Definition* on page 37.

Use the following procedures to configure a CV definition:

The basic definition uses the minimum information to validate a user against the LDAP database. It authenticates the trading partner by comparing the information presented to the LDAP database.

Create an Application Output Definition for the loginCredentials (sterling) Definition

The basic definition uses the minimum information to validate a user against the LDAP database. It authenticates the trading partner by comparing the information presented to the LDAP database.

Basic SSH Key Authentication Definition Worksheet

Before you configure an SSH Key Authentication definition, gather the following information. Use the following table to identify the values to assign to the EA fields:

EA Field	Description	Value
Authentication type	SSHKEY	
Profile name	Name for the profile	
Name	Automatically populated with sshPublicKeyQuery	
Connection specification	Use globally defined connection. Select the connection defined.	
Base DN	Distinguished name for the user container. For example, CN=Users,DC=example,DC=com.	
Application Feature	Lookup login credentials (Sterling) to configure user ID, password, or routing key mapping.	
Specify query parameters	Allow you to define the query parameters	
Return Attributes	Mapped credentials to return. By default, loginId, loginPwd, and routingKeyName are returned. Delete any attributes you don't want to map.	

Create a Basic SSH Key Authentication Definition

To create an SSH key authentication definition:

1. From the Authentication Definitions window, click + to add an SSH definition.
2. Select SSHKEY as the SSH Key Authentication type.
3. The name is automatically populated with sshPublicKeyQuery. Enable **Use globally defined connection** as the connection method.
4. Select the global connection definition you defined for the LDAP server.
5. Select **Specify query parameters** and click **Next**.

6. On the Query Parameters screen, define the Base DN and Return Attributes. Click **Next**.
7. Click **Save** to save the definition. Click **Close**.

Add an SSH Application Output to an Authentication Definition

Create an SSH Application Output definition to perform an SSH user and key query. Lookup loginCredentials returns login credentials to the client.

You configured the schema to define the objects allowed in the directory when you configured sci.schema for OpenLDAP or and v3. for IBM Tivoli Directory Server.

Before you create an SSH application output, define a basic SSH authentication. Refer to *SSH Key Authentication Definition* on page 131.

Advanced SSH Key Application Output Worksheet

Before you configure an SSH Key Application Output definition:

EA Field	Description	Value
Application Feature	Qutput definition to use.	Lookup loginCredentials (sterling)
Use globally defined connection	Connection for the Active Directory server	
Base DN	Distinguished name for the user container. For example, CN=Users,DC=example,DC=com.	

Create an Application Output Definition for the loginCredentials (sterling) Definition

To create an application output definition for the loginCredentials (sterling) definition:

1. On the Application Output screen, select **Lookup loginCredentials (sterling)** from the **Application Feature** drop-down box.
2. Click **Query** to create an LDAP attribute query.
3. Enable the Use globally defined connection option and select the server definition from the drop-down box. Click **Next**.
4. The Query Parameters screen is populated. Edit the Base DN field. Click **Next**.
5. Review the details summarized on the Confirm screen and click **Save**.
6. Click **Close**.

Advanced SSH Custom Application Output Worksheet

Before you configure an SSH Key Application Output definition, gather the following information. Use the following table to identify the values to assign to the EA fields:

EA Field	Description	Value
Application Feature	Qutput definition to use	Lookup loginCredentials (custom)

EA Field	Description	Value
Use globally defined connection	Connection definition for Active Directory or LDAP	
Output Name	Name of output file used to return attributes	mappedUid
Query Name	Name of return attribute of password for the destination service, located in the right pane	loginId
Output Name	Name of output file used to return attributes	mappedPwd
Query Name	Name of return attribute of password for the destination service, located in the right pane	loginPwd

Create an Application Output Definition for the loginCredentials (custom) Definition

To create an application output definition for the loginCredentials (custom) definition:

1. On the Application Output screen, click the **Application Feature** drop-down box. Select **Lookup loginCredentials (custom)**.
2. Click **Query** to create an LDAP attribute query that returns the attributes mapped to application output to the client application.
3. Select the **LDAP server definition** from the Use globally defined connection option. Click the Query Parameters tab.
4. Construct an attribute query to return the user ID from your directory object. See *Query Parameters Worksheet* on page 153 for instructions.

   After the wizard closes, manually map the return attributes to the output names.
5. In the left pane, click the Output Name called mappedUid. Selecting the output highlights it.
6. Click the query return attribute that corresponds to the user ID for the destination service in the right pane. Selecting the attribute highlights it, and **Map** is no longer dimmed.
7. Click **Map** to map the user ID attribute to the output name.
8. In the left pane, click the Output Name called mappedPwd.
9. Click the query return attribute that corresponds to the password for the destination service in the right pane.
10. Click **Map** to map the password attribute to the output name.
11. Click **Next** until the Confirm screen is displayed.
12. Click **Save**.

Add an SSH Application Output to an Authentication Definition

Create an SSH Application Output definition to perform an SSH user and key query. Lookup loginCredentials is an option you select in an Application Output definition. It returns login credentials to the client application.

You configured the schema to define the objects allowed in the directory when you configured sci.schema for OpenLDAP or and v3. for IBM Tivoli Directory Server.

Advanced SSH Key Application Output Worksheet

Before you configure an SSH Key Application Output definition, gather the following information. Use the following table to identify the values to assign to the EA fields:

EA Field	Definition
Authentication type	SSHKEY
Profile name	Name for the profile
Name	Automatically populated with sshPublicKeyQuery
Connection method	Use globally defined connection
Global connection definition	Definition you created for the LDAP server
Specify Query Parameters	Allows you to specify query parameters
Base DN	Distinguished name for the user container. For example, CN=Users,DC=example,DC=com.
Application Feature	Lookup login credentials (Sterling) to configure user ID, password, or routing key mapping.
Use globally defined connection	Connection definition for the Active Directory server
Specify query parameters	Allow you to define the query parameters
Return Attributes	Mapped credentials to return. By default, loginId, loginPwd, and routingKeyName are returned. Delete any attributes you don't want to map.

Create an Application Output Definition for the loginCredentials (sterling) Definition

To create an application output definition for the loginCredentials (sterling) definition:

1. On the Application Output screen, click the **Application Feature** drop-down box. Select **Lookup loginCredentials (sterling)**.
2. Click **Query** to create an LDAP attribute query.
3. Enable Use globally defined connection and select the server from the drop-down box. Click **Next**.
4. The Query Parameters screen is populated with parameters. Edit the **Base DN** field. Click **Next**.
5. Review the details summarized on the Confirm screen and click **Save**.
6. Click **Close** to return to the Application Output Definition screen, where return attributes to outputs have been mapped automatically.

Create an Application Output Definition for the loginCredentials (custom) Definition

To create an application output definition for the loginCredentials (custom) definition:

1. On the Application Output screen, click the **Application Feature** drop-down box. Select **Lookup loginCredentials (custom)**.
2. Click **Query** to create an LDAP attribute query that returns the attributes mapped to application output to the client application.

3. Enable the Use globally defined connection option and select your LDAP server from the drop-down box. Click the Query Parameters tab.
4. Construct an attribute query to return the user ID and password from your directory object. See *Query Parameters Worksheet* on page 153 for instructions.

   After the wizard closes, manually map the return attributes to the output names.
5. In the left pane, click the Output Name, mappedUid. Selecting the output highlights it.
6. Click the query return attribute that corresponds to the user ID for the destination service in the right pane. Selecting the attribute highlights it, and **Map** is no longer dimmed.
7. Click **Map** to map the user ID attribute to the output name.
8. Repeat steps 5 through 7 to map the password returned from a query to mappedPwd.

Edit or Copy an SSH Key Authentication Definition

You can change how EA authenticates SSH keys and users by copying or editing SSH key authentication definitions. Save time and reduce errors by copying, renaming, and editing a similar definition to create the new one.

To copy or edit an SSH key authentication definition:

1. From the Authentication Definitions window, perform one of the following actions:
   - To make a copy of an authentication definition, highlight the definition and click ⎘. Type a unique **Profile Name**.
   - To edit an authentication definition, double-click the definition.
2. Update the parameters as required.
3. Click **OK**.

Delete an SSH Key Authentication Definition

To delete an SSH key authentication definition:

1. From the Authentication Definitions window, select the definition to delete and click −.
2. Click **OK**.

Create Generic Authentication Definitions

Authentication definitions specify how EA authenticates a security principal when a client sends a request. Generic authentication definitions enable custom definitions that use a custom exit. Attribute queries and attribute assertions can be associated with a generic authentication definition.

Organization of the Generic Authentication Definition

The section instructs you how to create a basic setup.

How to Use the Worksheets

Before you configure a definition, gather information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

After you complete the definition, test it to make sure EA can connect to SSP and the outbound server. Continue to the next section to add more features.

Worksheet - Generic Authentication Definition

Before you configure a generic authentication, gather the following information:

Field	Description	Value
Profile name	Name for the generic authentication definition.	
Authentication type	Protocol to use to authenticate the user.	Generic
User ID required	Enable this option to require that the user present a user ID for authentication.	
Password required	Enable this option to require that the user present a password for authentication.	
Authenticate using custom exits	Enalbe this otpion to require the use of a custom exit for authentication. Provide the name of the custom exit.	

Create a Generic Authentication Definition

To create a generic authentication definition:

1. From the Authentication Definitions window, click +.
2. Type a name in the Profile name field and select Generic as the Authentication type.
3. Enable the components to use to authenticate the user: User ID required or Password required.
4. To use a custom exit, enable **Authenticate using custom exits**. Select the custom exit to use.

Note: If you define authentication as part of a certificate validation request, EA variables set during certificate validation are available for the authentication. Refer to *Use Definition Variables* on page 161 for more information about variables.

5. Click **Next** until the Confirm screen is displayed.
6. Click **Save** and click **Close**.

Configure a Custom Exit for a Generic Authentication

EA allows the use of a Java class or operating system command to implement a custom exit from a generic authentication definition. If you use a Java class as a custom exit, the class must implement the Sterling-provided interface, SEASCustomExitInterface.

Prerequisites for Using a Custom Exit

Before you begin configuring a custom exit, perform the following prerequisite tasks:

Review the files in the *install_dir*/doc and *install_dir*/samples subdirectories, where *install_dir* is the EA installation.

Create a definition with a custom exit to a script or program. Define the functionality by writing code that runs from the operating system command line.

For Java classes created, copy the class files or a .jar file to the *install_dir*/lib/custom directory.

Set logging to an appropriate level (such as DEBUG or ALL) to enable reviewing the results of the Java class, script, or program that implements your custom exit.

Develop and Deploy a Custom Exit Class in Java

The SEASCustomExitInterface interface and a sample class implementing the interface are documented in the javadoc located in the *install_dir*/doc directory and can be found in the archive, *install_dir*/lib/sterling/custom-exit.jar. The source for the sample implementation can be found at *install_dir*/samples/SampletAuthenticationExit.java.

The interface provides an initialization method that accepts a list of custom properties you define for your class. You specify these properties (names and values) from the GUI as part of the Custom Exit configuration in the generic authentication definition.

Compile your exit classes and provide them in a jar file, or as class files with package structure preserved, in the *install_dir*/lib/custom directory. The custom exit class loader searches all jar files and packages in this directory for the custom exit class in the generic authentication definition.

Specify a Java Class for a Custom Exit in a Generic Authentication Definition

To specify a Java class for a custom exit in a generic authentication definition:

1. Open the generic authentication definition.
2. Click the **Generic Authentication** tab on the Update Authentication Definition screen.
3. Enable **Authenticate using custom exits** and then click [...].
4. On the Custom Exits dialog box, enable **Java class**.
5. In the **Class name** field, type the fully-qualified class name in the format *packageName.className* when you specify the custom exit class that implements SEASCustomExitInterface.

6. To specify properties for the class, click …. On the Properties dialog box, specify the name and value for each property that is required to initialize your custom exit class. Use + and − if you need to add or remove rows of name and value pairs.

After the generic authentication definition processes an incoming authentication request, review the log for messages related to authentication through the custom exit.

Specify an Operating System Command for a Custom Exit

To specify the operating system command to use for the custom exit:

1. Double-click the generic authentication definition.
2. Click the **Generic Authentication** tab on the Update Authentication Definition screen. Enable **Authenticate using custom exits** and then click ….
3. To authenticate an operating system command as a custom exit, enable **Native OS command.**
4. For **Command line**, specify the operating system command to use, including all command line arguments. A user ID and password must be passed as variables on the command line.
5. Specify the method to use to pass the certificate chain to the operating system command:

Note: A certificate validation request must be performed before you send a certificate chain.

- Enable **Certificate file** to send the certificate chain as a file. Define the following parameters:

  a. File name for the certificate chain or a valid variable expression (such as the default, cert{counter}.pem).

  Tip: The default file name uses a counter to ensure that the file name is always unique. The variable {counter} begins with a value of 0 and increments after each invocation of the exit, resulting in the following file names: cert0.pem, cert1.pem, cert2.pem, and so on. The file name can be passed on the command line as the variable {filename}. For example, the following command is a valid use of the file name:

  ```
  openssl x509 -in {filename}
  ```

  b. Specify the certificate chain **File format** as **PEM** or **DER**.

  c. To remove the certificate after the exit completes, enable **Delete file after exit.**

  d. Click **Standard input (PEM format)** to send the certificate chain through the standard input stream.

- Specify when to run the custom exit and authentication:

  a. Select **Run default validator after exit** to process the authentication validation definition after the custom exit.

  b. Select **Run custom exit synchronously** to enable synchronous use of the custom exit. If you select this option, and if a client application sends an authentication request with a reference to a definition with the custom exit and the exit is currently running, current exit processing must complete before a subsequent invocation can run.

6. Specify **standard error log level** and **standard output log level** to define how messages and errors from the custom exit log. Errors and console output is logged in the SEAS.log.
7. To redirect errors and output that EA returns to the client, select one or more of the following parameters:
   - **Log output from stderr to response message—**send error logs to the response message.
   - **Log output from stdout to response message—** send logs to the response message.

Create an Application Output Definition for a Generic Authentication Definition

Create an Application Output definition if you want EA to return application-specific data to the client application. **mappedUid** and **mappedPwd** are defined to map log in credentials for Sterling Secure Proxy (SSP).

When SSP is configured to use this feature, a user logs in to SSP with a credentials. EA authenticates the credentials and returns a different set of credentials to use to log in to the service in the trusted zone. This feature protects your internal systems because the internal user IDs and passwords are not provided to external users. External users are only able to log in through the SSP.

When creating an Application Output definition within a Generic Authentication definition, specify the output values for mappedUid and mappedPwd as fixed values or variable expressions. Refer to *Use Definition Variables* on page 161 for more information.

Refer to *Create LDAP Authentication Definitions* on page 141 for information about authenticating users using LDAP. When you create LDAP authentication definitions, it is assumed that the mapped credentials will be stored in the LDAP directory, and that a query can be constructed to retrieve those credentials. A wizard launched from the Application Output Definition panel constructs the LDAP query and creates expressions that are assigned to the application output: mappedUid and mappedPwd.

In the typical case, these will be assigned as shown in the table below:

Output Name	Sample Value
mappedUid	{attr[MapCredentials].loginId}
mappedPwd	{attr[MapCredentials].loginPwd}

When creating an Application Output definition within a Generic Authentication definition, you can use LDAP as the credential store for mapping credentials. To use this method, first create an Attribute Query definition, as described in *Create LDAP Authentication Definitions* on page 141. Then, manually make the assignment to the output names.

For example, if you create an Attribute Query named MapCredentials to return the loginId and loginPwd attributes of a loginCredentials entry as described in *Create LDAP Authentication Definitions* on page 141, the values shown in the preceding table are used for the Application Output definition.

Note: Application outputs can also be created and assigned directly using a custom exit written in the Java programming language. For details, see SEASCustomExitInterface.REQKEY_APPOUTPUTS in the Javadoc installed with EA.

Create LDAP Authentication Definitions

Authentication definitions specify how EA authenticates a security principal when a client sends a request. They include parameters for connecting to a server, determining the authentication principal, and mechanisms used for authentication. Authentication definitions specify parameters used when accessing directories. It can include definitions for any attribute queries, attribute assertions, and application-specific outputs required to perform authentication.

Organization of the Basic LDAP Authentication Definition

This section instructs you how to create a basic setup. Determine which features apply to your environment and configure only those.

How to Use the Worksheets

Before you configure a definition, gather information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

After you complete the definition, test it to make sure EA can connect to SSP and the outbound server. Continue to the next section and add another security feature.

Prerequisites

Before you create an LDAP authentication definition, define a global connection setting for the LDAP server. Refer to *System-Wide LDAP Connection Definition* on page 37.

The basic definition uses the minimum information to validate a user against the LDAP database. It authenticates the trading partner by comparing the information presented to the LDAP database.

Basic LDAP Authentication Definition Worksheet

Before you configure an LDAP definition, gather the following information.

Parameter	Description	Value
Profile Name	Name of the definition.	
Host	Host name of the LDAP server.	
Port	Port used to connect to the LDAP server.	
LDAP principal to bind	Security principal to bind to the LDAP server. Select User ID from request. This option binds to the LDAP server using the client user ID. Use this option if the client sends a user ID that can be used in the bind request.	Select User ID from request

Create a Basic LDAP Authentication Definition

To create a basic LDAP authentication definition:

1. From the Authentication Definitions window, click +.
2. Use the default authentication type, LDAP. Type a name in the **Profile Name** field.

3. Type the host and port for the LDAP server.
4. In the LDAP principal to bind section, enable **User ID from request**.
5. Click **Next** until the Confirm screen is displayed.
6. Click **Save** and click **Close**.

Add Features to an LDAP Authentication Definition

You defined a basic LDAP Authentication definition, use the information in this section to add advanced features to an LDAP authentication definition. Options include:

Change the Bind Method Used Between the EA Connection and LDAP on page 142

Enable TLS Between the EA Connection and LDAP on page 143

Enable Advanced LDAP Connection Functions on page 144

Change Password Settings Associated with an LDAP Authentication Definition on page 145

Create an Application Outputs Definition on page 146

Extend the Schema for OpenLDAP on page 146

Change the Bind Method Worksheet

Before you modify the bind method, identify which of the following methods to use:

Field	Description	Enable?
Search for user DN	Search the directory for the user ID, usually in the CN or UID attribute of the directory. The DN is the security principal for the bind operation. The default search is CN={*userId*}, where {*userId*} is a variable of the user ID. If the directory uses a different attribute to store the user ID, use that attribute (for example, UID).	
Search for user DN	Bind using the DN specified. This option can only be used if the name of the directory contains the user ID being authenticated. The RDN of the user entry must match the user ID received in the request; typically, the Common Name (CN).	
Specify user DN	The DN for the principal. Information pre-populated indicates how to specify the DN. Use the format: *cn*={*userID*}, *base DN*, where: *cn* is the attribute to name the entry, *userID* is the user ID from the request, and *base DN* is a comma-delimited list of RDNs (Relative Distinguished Names) of the parent of the user entry.	
DN from Certificate Validation	Bind using the DN returned from the subject verification query used during a certificate validation.	
Other principal format	Bind to the LDAP server using the security principal specified by the expression you type in the field. You can use EA variables.	

Change the Bind Method Used Between the EA Connection and LDAP

The basic definition uses the user ID from the request to bind to LDAP. Other methods are available. Methods include the ability to search for a user DN using the user ID in the request, the

DN to use to bind, the DN defined in the Certificate Validation definition, or a method using a expression you define.

To change the LDAP bind method:

1. In the Authentication Definitions window, double-click the LDAP definition to modify.
2. In the LDAP principal to bind section, enable one of the following methods:
   - Search for user DN
   - Specify user DN
   - DN from Certificate Validation
   - Other principal format
3. Click **Next** until the Confirm screen is displayed.
4. Click **Save** and click **Close**.

Enable TLS Worksheet

Before you enable TLS, use the worksheet to define the values for the following fields:

Field	Description	Value
Protocol	Enable the TLS protocol.	ldaps://
Client Key Certificate Alias	Type the key certificate to use with the LDAP server when client authentication is enabled.	
LDAP Version	Select 3 to turn on v3 extended operation.	3
Start TLS	Select yes to enable to TLS 3 version	Yes

Enable TLS Between the EA Connection and LDAP

The basic definition does not enable TLS to secure the connection between EA and LDAP.

To enable TLS:

1. In the Authentication Definitions window, double-click the definition to modify.
2. In the **Protocol** field, select ldaps:// and click **Next**.
3. Type the key certificate in the **Client Key Certificate Alias** field.
4. In the **LDAP Version field**, select 3.
5. In the **Start TLS** field, select **Yes**.
6. Click **Next** until the Confirm screen is displayed.
7. Click **Save** and click **Close**.

Enable Advanced Connection Worksheet

Before you modify the connection settings, identify which of the following features to enable and the value for each field:

Field	Description	Value
Authentication method	Change the authentication method to one of the following values: ◆ Digest-MD5—Authenticate using the Digest-MD5 encrypted password and transmit message digests instead of a clear text password. ◆ CRAM-MD5—Authenticate against the CRAM-MD5 encrypted password and transmit message digests instead of a clear text password. ◆ GSSAPI—Use Kerberos V authentication, the authentication used in Active Directory.	
Referral Action	To identify what to do when a request is referred by an LDAP server to another, enable one of the following actions in the field: ◆ Follow—Follow the referral to the referred directory. ◆ Ignore—Ignore the referral. ◆ Throw—Ignore the referral and generate an exception.	
Advanced	Define properties required by a JDNI service provider. Type the local bind address.	

Enable Advanced LDAP Connection Functions

The connection definition in a basic LDAP authentication definition configures the simple authentication method and no other connection settings.

To configure advanced connection settings:

1. In the Authentication Definitions window, double-click the definition to modify.
2. In the LDAP authentication window, click **Next.**
3. To modify the authentication method, change the authentication method in the **Authentication method** field.
4. To identify what to do when a request is referred by an LDAP server to another, enable an option in the **Referral Action** field.
5. To define properties required by a JDNI service provider:
    a. Click the box beside **Advanced** options.
    b. Type the local bind address.
    c. Click the box beside JNDI Properties.
    d. Provide the names and values of any special properties.
    e. Click **OK**.
6. Click **Next** until the Confirm screen is displayed.
7. Click **Save** and click **Close**.

Change Password Settings Worksheet

Before you modify the password settings, use the following table to identify the values to assign to the EA fields:

EA Field	Description	Value
Profile can be used to change password	Enable this option to enable the ability for the user to change his password. Default=disabled.	
Warn users when password about to expire	Enable this option to send a warning to the user that the password is close to expiration. This option is only available when the server is IBM Tivoli Directory Server or Microsoft Active Directory.	
Number of days before warning	Type how many days before the password expires to begin notifying the user.	
Service Principal Bind Information	The globally defined LDAP connection definition to use for the Directory server.	

Change Password Settings Associated with an LDAP Authentication Definition

The basic definition does not enable change password settings.

To enable the ability for the user to change password settings:

1. In the Authentication Definitions window, double-click the definition to modify.
2. In the LDAP authentication window, click **Next** until the Change Password dialog is displayed**.**
3. Provide values for the change password fields. Refer to the worksheet.
4. Click **Next** until the Confirm screen is displayed.
5. Click **Save** and click **Close**.

Edit or Copy an LDAP Authentication Definition

Changes in requests from client applications require that you make related changes in the authentication definitions. You can change how EA operates by copying, editing, and deleting authentication definitions.

To copy or edit an LDAP authentication definition:

1. From the Authentication Definitions window, perform one of the following actions:
   - To copy an authentication definition, select the definition to copy and click ⧉. Type a unique **Profile Name**.
   - To edit an authentication definition, double-click the definition to edit.
2. Update the parameters as required and click **OK**
3. Click **OK**.

Create an Application Outputs Definition

Create an Application Output definition for an authentication definition when you need to perform an LDAP query and return login credentials to the client application. Lookup Login Credentials is can be set in an Application Outputs definition to return login credentials to the client application.

The schema of an LDAP directory defines the objects allowed in the directory. A directory schema object is defined to store login credentials. EA includes schema extension files for OpenLDAP (sci.schema) and IBM Tivoli Directory Server (v3.schema) in the *install_dir*/schema directory, where *install_dir* is the EA application.

Use the schema files directly to extend your schema for OpenLDAP and IBM Tivoli Directory Server directories, or use the schema file as a reference when you manually extend other directories. If you use these directory extensions and populate directory entries, the creation of an application output definition is mostly automated. Alternatively, you can use an arbitrary directory object that stores a user ID and password, as long as the password can be retrieved as unencrypted data. When using an arbitrary directory object, you must define the attribute query that stores a user ID and password, as long as the password can be retrieved as unencrypted data. When using an arbitrary directory object, you define the attribute query that obtains the credentials for the application outputs definition. Then you can use controls on the Application Outputs Definition dialog to manually map attributes returned by the query to outputs that a client application can access.

Prepare the Directory for Use with Lookup Login Credentials

To add a directory object for Lookup Login Credentials, you must extend the schema for the directory. Sterling Commerce provides schema extension files for use with OpenLDAP. Use the following procedures to extend the schema for the server.

Note: Refer to *Configure Active Directory to Use with EA* on page 61 for instructions on configuring Active Directory.

For other LDAP servers, follow instructions provided with the product to manually extend the schema. Reference the schema file, *install_dir*/schema/sci.schema, for definition of the object class, loginCredentials, and its associated attributes.

Extend the Schema for OpenLDAP

To edit the schema for OpenLDAP:

1. Copy the OpenLDAP schema file (at *install_dir*/schema/sci.schema) to the schema subdirectory of OpenLDAP. Schema files are in the /etc/openldap/schema subdirectory.
2. Edit the slapd.conf file to add an include statement that includes the sci.schema. The slapd.conf file is normally in /etc/openldap.

   The following line includes the sci.schema for a standard OpenLDAP installation:

   ```
   include /etc/openldap/schema/sci.schema
   ```

3. Restart the LDAP server.

Extend the Schema for IBM Tivoli Directory Server

To edit the schema for IBM Tivoli directory server:

1. Copy the V3.sci and V3.openssh-lpk files, located in the *install_dir*/schema directory, to the schema subdirectory of the Tivoli installation, normally in the /usr/ldap/etc folder.
2. Edit the ibmslapd.conf file, located in the /user/ldap/etc directory. Add include statements for the V3.sci and V3.openssh-lpk schemas.

   Following is a sample of the include statements to add to V3.sci and openssh-lpk schemas to the ibmsladp.conf file:

```
include ibm-slapdIncludeSchema: /usr/openldap/etc/ldapschema/V3.sci
includeibm-slapdIncludeSchema: /usr/openldap/etc/ldapschema/V3.openssh-lpk
```

3. Restart the LDAP server.

Create Entries for Login Credentials

After you add the SCI schema objects to your directory, you can create loginCredentials entries. The supported directory structure creates separate loginCredentials entries as children of the authenticated user's directory entry; one for each destination service. Set the loginId and logingPwd attributes to the ID and password needed to login to the destination service. The password must be entered in binary. The attribute, loginTarget, must be set to the destination service name that is passed in the Authentication Request from the client application. With this arrangement, the query to fetch the credentials is pre-populated with the correct values and the mapping to outputs is performed automatically. Refer to *LDIF Entry Example* on page 147 to see an example of an entry in the supported structure. If you use a different structure from the preceding example, you must modify the attribute query to find the entry in your tree as required.

LDIF Entry Example

The LDIF entry below demonstrates an entry in the supported structure:

```
dn: cn=SSP1-App2 Login Credentials, cn=User010101, ou=SterlingEAS 2.0 Users,
dc=Sterling Commerce, dc=com
objectClass: loginCredentials
objectClass: top
cn: SSP1-App2 Login Credentials
description: User010101's login credentials for SSP1-App2 (loginPwd=loginPwd2)
loginId: loginId010101
loginPwd:: cGFzc3dvcmQ=
loginTarget: SSP1-App2
```

In the preceding LDIF entry, EA authenticates the user, User010101, by binding to the following DN: cn=User010101, ou=SterlingEAS 2.0 Users, dc=Sterling Commerce, dc=com. Assume that with the directory information tree structured as indicated in the example, a client application sends an authentication request that references a destination service, SSP1-App2. The user ID to log in to this service is loginId010101 and the corresponding password is loginPwd2. EA queries the loginCredentials entry and returns the user ID and password to the client application in the authentication response.

Note: The value of the loginPwd attribute is base64-encoded. If you need a tool to base64-encode a password, OpenSSL can do this using the following command line syntax:

```
openssl enc -a -e -in clearTextPasswordFile -out base64EncodedPasswordFile
```

Map Query Return Attributes to Application Output Names in an Application Outputs Definition

To create an application outputs definition:

1. On the Application Outputs screen, from the **Application Feature** field, select the method to use to return attributes to the client application for the authentication definition:
   - To query the Sterling Commerce loginCredentials directory object of returning attributes in EA, select **Lookup loginCredentials (Sterling)**.
   - To query any other directory object, select **Lookup loginCredentials (Custom)**.
2. Click **Query** to create an LDAP attribute query that returns the attributes to be mapped to application outputs for return to the client application.
   - If you selected the Sterling loginCredentials application feature:
     a. If the authenticated user has read permission on these entries, click **Next**. Otherwise, select your connection preference before proceeding to the next screen.
     b. With directory entries arranged as described in *LDIF Entry Example* on page 147, the Query Parameters screen includes the appropriate parameters; you can simply review them and click **Next**. Otherwise, edit the Base DN, Scope, and Match Attributes as needed before proceeding to the next screen. See *Query Parameters Worksheet* on page 153 for instructions.
     c. Review the details summarized on the Confirm screen. Click **Save** if all parameters are set correctly. Click **Done** to return to the Application Outputs screen, where the mapping of return attributes to outputs has been performed automatically.
   - If you selected the Lookup loginCredentials (Custom) application feature:
     a. Construct an attribute query to return the user ID and password from the directory object. See *Query Parameters Worksheet* on page 153.

        After the Attribute Query wizard closes, you must manually map the return attributes to the respective output names.
     b. In the left pane, click the Output Name, mappedUid. Selecting the output highlights it.
     c. Click the query return attribute that corresponds to the user ID for the destination service in the right pane. Selecting the attribute highlights it and **Map** is no longer dimmed.
     d. Click **Map** to complete the mapping of the user ID attribute to the output name. Repeat this procedure to map the password attribute returned from your query to the Output Name, mappedPwd.

Create Tivoli Access Manager (TAM) Authentication Definitions

Create a Tivoli Access Manager (TAM) authentication definition to specify parameters that EA uses when accessing TAM resources. EA interfaces with TAM and provides user ID and password authentication and/or user DN authentication. DN authentication verifies the subject of a certificate received during certificate validation. TAM authentication can also provide application-level authorization to access a destination service specified in the authentication request and credential lookup to log into the destination service.

Before you configure a TAM authentication definition, install TAM on the computer with EA.

Configure TAM for EA Use

To configure TAM for EA:

1. Install 1.4.2 JRE on the computer where EA is installed, either as a system JRE or a private JRE for a user.
2. Set the JAVA_HOME environment variable to point to the JRE.
3. Run the IBM wizard called install_amjrte. This wizard installs the TAM API into the JRE.

   Refer to *IBM Tivoli Access Manager Base Installation Guide, Version 5.1*.
4. In EA, create a TAM authentication definition that references the JRE and the configuration file created by the Java utility. Make sure you define the following fields in EA:
   - Target JRE location—location where the JRE is installed. When you set up a definition, it requires the JRE configured with Access Manager Runtime for Java.
   - TAM Config File URL—configuration file created by the Java utility in step 4.

   Because EA uses JRE 1.6, it cannot run in the same JRE as the TAM interface.

Install the TAM API

Run the java utility, com.tivoli.pd.jcfg.SvrSslCfg, to create a TAM configuration file, SSL key, and data to communicate securely with TAM servers. Obtain this utility from IBM.

The following example demonstrates how the IBM Java utility configures EA into TAM.

```
> export JAVA_HOME=/home/SeasAdmin/java/j2sdk1.4.2_12
> export PATH=$JAVA_HOME/bin:$PATH
> java com.tivoli.pd.jcfg.SvrSslCfg -action config -admin_id sec_master
-admin_pwd masterpass -appsvr_id SterlingEAS_ID -appsvr_pwd ldapPassword
-host SterlingEAS_host -mode remote -port 999 -policysvr tamPolicySvr:7135:1
-authzsvr tamAuthzSvr:7136:1 -cfg_file /home/SeasAdmin/tam/config_file.conf
-key_file /home/SeasAdmin/tam/keystore_file.ks -domain Default -cfg_action
create
```

In the example, a private JRE is installed at /home/SeasAdmin/java/ and SeasAdmin is a user account for administering EA for TAM. Refer to the parameters in the following table to understand how the Java utility generates the SSL key and configuration file.

Parameter	Description
-host	Host name of the EA server.
-appsvr_id	ID defined by SvrSslCfg creates, for example EA_ID/EA_host.
-appsvr_pwd	Password for the new user account created in the TAM user registry.
-port	Listen port for definition updates. It must be specified although it is not used by EA.
-cfg_file	Configuration file created by the IBM com.tivoli.pd.jcfg.SvrSslCfg utility. Reference this file from the definition you create in EA.
-key_file	Java key store created by the utility. Private key and certificates are written to this file for SSL communications to the TAM policy and authorization servers.

Log Information for TAM

Because the TAM API communicates over standard I/O streams to the CV process, the logger cannot use the console appender for output. By default, both processes share conf/log4j.properties for configuring logging as well as the active log file. You can create a separate log4j.properties file for the TAM API process, to allow the parent process to log to the console. The TAM API process looks for its own log4j.properties file in the lib/sterling/retro14 directory. If it does not find its own properties file, the parent log4j.properties file is used.

Organization of the Basic TAM Authentication Definition

This section instructs on you how to create a basic setup. Determine which features apply to your environment and configure only those.

How to Use the Worksheet

Before you configure a definition, gather information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

After you complete the definition, test it to make sure EA can connect to SSP and the outbound server. Continue to the next section and add another security feature.

Prerequisites

Before you create an TAM authentication definition, configure TAM for EA and install a TAM API.

The basic TAM definition uses the minimum information to validate a user against TAM. It authenticates the trading partner by comparing the information presented to the TAM database.

Basic TAM Authentication Definition Worksheet

Before you configure a TAM authentication definition, gather the following information.

Parameter	Description	Value
Profile name	Name for the TAM authentication definition.	
Authentication type	Select TAM to display the TAM Authentication dialog.	TAM
TAM config file URL	URL to the TAM configuration file. It contains an SSL key and configuration data for secure communications with the authorization server. For example, the file:///home/SeasAdmin/tam/config_file.conf specifies a local configuration file.	
Target JRE location	Path to the JRE where TAM API is installed. For example, /home /SeasAdmin/java/j2sdk1.4.2_12 for a JRE.	
TAM user to authenticate	Method to define the TAM user to bind for authentication: ◆ Certificate subject DN—A follow-up request to a previous certificate validation request. The certificate subject DN from the request is the security principal sent to the TAM API for authentication. The password and/or user ID are required only if specified by the corresponding options. ◆ User ID from request—User ID in the request must be a user in the TAM user registry. ◆ Other—Supports specification of the security principal as a reference to an attribute query; for instance, to use the DN returned from an LDAP search for a certificate subject.	
User ID required	Request must specify a user ID.	
Password required	Request must specify a password.	
Authorize access to Destination Service	Authorize a user to access the destination service. Specify the TAM resource of the destination service and the access permissions to authorized. The default resource is /SEAS/profileName/{dstsvc}. Specify a valid resource name in the secure domain. The default permissions are Traverse and view (Tv). Specify valid permissions, that reflect the access anticipated by the client application.	

Create a Basic Tivoli Access Manager Authentication Definition

To create a Tivoli Access Manager (TAM) authentication definition:

1. From the Authentication Definitions window, click + to display the LDAP Authentication.
2. In the **Authentication type** field, select **TAM** to display the Tivoli Access Manager Authentication screen.

Note: When using TAM authentication, the first line of the log4j.properties file should remain commented out. The TAM authentication does not function if console output is enabled.

3. On the Tivoli Access Manager Authentication screen, specify the parameters and click **Next**. Refer to the values you identified in the worksheet.
4. Click **Next** until the Confirm screen is displayed.
5. Click **Save** and click **Close**.

Create a TAM Application Output Definition

The application output for TAM is implemented using TAM GSO resource credentials.

To create an application output definition for TAM:

1. On the Application Output screen, specify outputs to return to the client.
   - Return TAM Credentials
   - Return Destination Service Login Credentials
2. Click **Next** and click **Save**.

Edit or Copy a TAM Authentication Definition

Changes in requests from client applications require that you make related changes in the authentication definitions. Change how EA operates by copying, editing, and deleting authentication definitions.

To copy or edit a TAM authentication definition:

1. Do one of the following:
   - To copy a TAM authentication definition, select the definition to copy and click ⧉. Type a new **Profile Name**.
   - To edit a TAM authentication definition, double-click the definition.
2. To change a TAM authentication, update parameters as required.

Delete an Authentication Definition

To delete an authentication definitions:

1. From the Authentication Definitions window, select the authentication definition and click −.
2. Click **OK**.

Create an Attribute Query and Assertion

The certificate validation and authentication definitions can include LDAP attribute queries to find and check data from a request against entries in a directory.

Organization of the Attribute Query Definition

This section instructs you on how to create a basic setup. Determine which features apply to your environment and configure only those.

How to Use the Worksheets

Before you configure a definition, gather information on the worksheet provided. Accept default settings when the field is not listed in the worksheet.

Prerequisites

Before you create an LDAP authentication definition, define a global connection setting for the LDAP server. Refer to *System-Wide LDAP Connection Definition* on page 37. Define an authentication definition.

The basic definition uses the minimum information to validate a user against the LDAP database. It authenticates the trading partner by comparing the information presented to the LDAP database.

Basic Attribute Query Worksheet

Before you configure an attribute query, gather the following information:

Parameter	Description	Value
Name	Name of the LDAP attribute query definition.	
Connection specification	Method to use to connect to the LDAP server: Select a connection definition that you defined.	Use globally defined connection
Query specification	Query specification to use for the definition: ◆ Specify query parameters—Manually define parameters to use in the query. If you select this option, complete the values in the LDAP Query Parameters worksheet. ◆ Specify query as URL to provide a URL to perform the LDAP attribute query.	

Query Parameters Worksheet

If you want to manually define query parameters, gather the following information:

Parameter	Description	Value
Protocol	Protocol to connect to the LDAP server: ldap:// (nonsecure) or ldaps://(secure).	

Parameter	Description	Value
Host	Host name of the LDAP server.	
Port	Port number to connect to the LDAP server.	
Base DN	Starting point in the directory to begin the search.	
Return Attributes	Attribute types to return from the entries that match.	
Scope	Starting point when performing a search. Specify one of the following options: ◆ Base—At the Base DN to retrieve data from a known entry in the directory. ◆ One Level-—Only the level immediately below the Base DN. ◆ Sub Tree—The entire sub-tree below the Base DN.	
Match Attributes	Filter to determine which directory entries are a match. Click +, specify a name, a value to match, and click **OK**.	
Query Timeout	How long (format MM:SS) before the LDAP attribute query times out and processing ends.	

Create an Attribute Query Definition

LDAP parameters specify a search operation to locate directory entries and return attributes.

To create a query:

1. Double-click the authentication definition to modify.
2. Click **Next** until the Attribute Query Definitions screen is displayed.
3. Click + to display the Add Assertion Definition dialog box.
4. Type a name for the attribute query, the connection specification, and the query specification. Click **Next**.
5. Type the query parameters. Refer to *Query Parameters Worksheet* on page 153.
6. Click **Next** until the Confirm screen is displayed.
7. Click **Save** and click **Close**.

Advanced Query Options

Define one or more of the following advanced options to further filter search criteria:

Add a Query to Check for Allowed IP Addresses on page 155

Complete the procedure, Create an Attribute Query Definition on page 154, to add the query. on page 155

Add a Query to Validate an IP Address and Certificate on page 156

Add a Query to an EA Authentication Profile to Validate the User ID and Service on page 157

JNDI Properties Worksheet

If you use a JNDI (Java Naming Directory Interface) service provider that requires special properties, you can assign the custom JNDI properties to associate with the connection. Use the following table to gather the information required to configure JNDI properties:

Parameter	Description	Value
Name	Name of the JDI property to define	
Value	Value to define for the property	
Name	Name of the JDI property to define	
Value	Value to define for the property	

Add a Query to Check for Allowed IP Addresses

Define a query to look up the incoming IP address on the Allowed Hosts container. If found, the query is successful and the dn attribute of the host record is returned. If not found, the query fails and the certificate validation, user authentication, or SSH key authentication request fails.

Use the following table to gather the information required to configure this query:

EA Field	Description	Value
Name	Name of the attribute query definition.	
Connection specification	Use globally defined connection Select the connection definition for LDAP	Use globally defined connection Select the connection
Base DN	Distinguished name where service groups are stored	
Return Attributes	Attribute to return from the entries that match	dn
Scope	Starting point when performing a search. Specify One Level to allow only the level immediately below the base DN	One Level

Match Attributes	Filter to determine which directory entries match.	Name=ipNetworkNumber
		Value=IpAddress

Complete the procedure, *Create an Attribute Query Definition* on page 154, to add the query.

Add an Assertion to the User Authentication Profile to Validate an IP Address and User ID

Create a user authentication definition before you add this query.

Use the following table to gather information required to configure this assertion:

EA Field	Description	Value
Name	Name of the attribute query definition	
Connection specification	Use authenticated user's connection	
Base DN	Distinguished name where service groups are stored	{principal}
Return Attributes	Attribute to return from the entries that match	dn, ipHostNumber
Scope	Starting point when performing a search. Specify Base to allow the base DN to retrieve data from a known entry in the directory.	Base
Attribute Assertions Definitions		
Name	Name for the assertion	
Assertions	{attr[FindUserIPAddrs].ipHostNumber} == any || {attr[FindUserIPAddrs].ipHostNumber} == {ipAddress}	

Complete the procedure, *Create an Attribute Assertion Definition* on page 158, to add the assertion.

Add a Query to Validate an IP Address and Certificate

Create a certificate authentication definition before you complete this procedure.

The FindHostGroup query looks up the host group corresponding to the certificate's organization, including the incoming IP address. If the group is not found, the certificate validation request fails.

Use the following table to gather information required to configure this query. Use the first table to define the query to look up an incoming IP address and the second table to create a query to find the host group for the certificate's organization. When you add the queries, place the first query called FindHostDN first in the order and FindHostDN second in the list.

EA Field	Definition	Value
Name	Attribute query definition	FindHostDN
Connection specification	Use globally defined connection Select the connection definition for the AD server.	
Specify query parameters	To define the query parameters	

EA Field	Definition	Value
Name	Attribute query definition	FndHostGroup

Connection Specification	Use globally defined connection Select the connection definition for the AD server	
Base DN	Distinguished name for the hosts group, for example, CN=Host Groups,CN=SEAS,DC=example,DC=com).	
Return Attributes	Attribute to return from the entries that match	dn, uniqueMember
Scope	Starting point when performing a search.	One Level

Match Attributes	Filter to determine which directory entries match
	Name=o Value=I{subject.o, none}
	Name uniqueMember Value= {attr[FindHostDN].dn}
	Note: Specify None if the certificate subject does not have an organization. You can create a host group named No Org Hosts with an o attribute equal to none to group hosts that present certificates with no organizations.	

Complete the procedure, *Create an Attribute Assertion Definition* on page 158 to create an assertion to compare the incoming IP address against the list of IP addresses assigned to the user.

Add a Query to an EA Authentication Profile to Validate the User ID and Service

Add a query to an authentication profile to validate the user ID and service. Create an authentication definition before you complete this procedure.

Use the following table to gather information required to configure this query.

EA Field	Description	
Name	Attribute query definition	
Connection specification	Use authenticated user’s connection	
Base DN	Distinguished name where service groups are stored	
Return Attributes	Attribute to return from the entries that match	dn
Scope	Starting point when performing a search	One Level

Match Attributes	Filter to determine which directory entries match
	Name=ou Value=destinationService
	Name=uniqueMember Value={principal}	

Add an Advanced Feature to a Basic Authentication Definition

LDAP parameters specify a search operation to locate directory entries and return attributes. Determine which of the advanced queries you want to perform and complete the worksheet. Use the follow procedure to add the advanced query to the definition:

To add an advanced query to a basic definition:

1. Double-click the definition to modify.
2. Click **Next** until the Attribute Query Definitions screen is displayed.
3. Click + to display the Add Assertion Definition dialog box.
4. Type a name for the attribute query, the connection specification, and the query specification. Click **Next**.
5. Type the query parameters. Refer to *Create an Attribute Query Definition* on page 154.
6. Click **Next** until the Confirm screen is displayed.
7. Click **Save** and click **Close**.

Manage an Attribute Assertion Definition

You can create an attribute assertion definition to specify a Boolean statement that must evaluate as true in order for the authentication request or certificate validation request from a client application to succeed. Attribute assertions allow the specification of additional conditions and can compare details from the request to fixed data or to attributes returned from queries.

Create an Attribute Assertion Definition

To create an attribute assertion:

1. From the Attribute Assertion Definition screen, click + to display the Add Assertion Definition dialog box. Specify the following parameters:
    - Name
    - Assertion
2. Click **OK**.

Edit or Copy an Attribute Query Definition

To edit or copy an attribute query definition, click the **Summary** tab to view a list of the parameters for each functional area. Then click the tab for the area to edit.

To edit or copy an attribute query definition:

1. Double-click the definition to modify.
2. Click **Next** until the Attribute Query Definitions tab is displayed.
3. Double-click the Query Definition to modify or copy.
4. Perform one of the following actions:
    - To copy a definition, select the definition to copy and click ⧉. Type a unique **Name**.
    - To edit a definition, double-click the definition to edit.
5. Make the changes required and click **OK**.

Delete an Attribute Query Definition

To delete an attribute query definition:

1. From the Certificate Validation Definitions window, double-click the certificate validation definition to delete.
2. Click **Next** until the Attribute Query Definitions screen is displayed.
3. Click the Attribute Query Definitions tab.
4. Select the query to delete and click −.
5. Click **OK**.

Copy or Edit an Attribute Assertion Definition

Delete an Attribute Assertion Definition

Copy or Edit an Attribute Assertion Definition

Copy an attribute assertion definition to create a new assertion definition with similar parameters.

To copy or modify an attribute assertion:

1. Do one of the following:
   - To copy a definition, highlight the definition with the attribute assertion definition and click 📝.
   - Double-click the definition to edit.
2. From the Attribute Assertion Definitions window, select the definition and click 📋.
3. On the Add Assertion Definition screen, specify a name.
4. Define the assertion and click **OK**.

Delete an Attribute Assertion Definition

To delete an attribute assertion definition:

1. From the Certificate Validation Definitions window, double-click the definition with the attribute assertion definition.
2. Highlight the attribute assertion definition to delete and click −.
3. Click **OK**.

Use Definition Variables

You can use variables in certificate validation and authentication definitions. Variables are resolved at runtime by data from the certificate validation or authentication request, from the entity's certificate, and from data returned from attribute queries you have configured.

Syntax and Rules

Referencing the Cert Variable in an Attribute Assertion

The following example illustrates the base DN from the preceding example after the variable is resolved:

Syntax and Rules

A variable consists of hierarchical groupings with nodes delimited by a period (.) or square brackets ([ ]). Observe the following rules or guidelines when you use variables:

EA variables are not case sensitive. The following examples represent the Common Name attribute (CN) of the subject field of a certificate and illustrate valid syntax formats:

- subject.cn
- subject[cn]
- Subject.CN

To reference a variable in a definition, enclose the variable in curly braces, for example, {Subject[cn]}.

A variable can be used in the specification of an attribute query. In the following example, the URL changes at runtime, based on the contents of the certificate subject referenced by the variable highlighted in bold:

```
ldap://ldaphost:389/cn={subject.cn},ou=users,dc=myCompany,dc=com?DN?base?objectClass=pkiUser
```

Variables representing a single element are represented by a single-node variable. For example, the client ID field of the request is represented by the single-node variable clientId.

Variables that represent complex objects are represented by multiple nodes. For example, a certificate includes a subject and issuer, both of which contain attributes such as CN and OU. The CN attribute of the subject is represented by the multi-node variable cert.subject.cn.

Many multi-node variables can be abbreviated by omitting the parent node, or nodes, if naming collisions are not created by doing so. For example, cert.subject.cn can be abbreviated as subject.cn, or cn. And cert.issuer.cn can be abbreviated as issuer.cn, but not as cn, because it would be indistinguishable from the subject CN abbreviation.

The root or intermediate node names used in some multi-node variables have a value associated with them. For example, when cert is specified alone, it represents the raw data of the end entity certificate in the certificate validation request.

You can assign a default value to variables to prevent a failure in the operation when a variable is specified in a configuration parameter, but the variable cannot be resolved.

For example, if you specify a Match Attribute in an LDAP Query Definition as: ou={issuer.ou}, the query and the validation fail if no OU attribute is defined for the issuer

Distinguished Name. To prevent a failure, append a comma and specify the default value inside the curly braces: ou={issuer.ou, default Issuer OU}. If the issuer DN in the certificate has no OU attribute, the Match Attribute resolves to: ou=default Issuer OU.

To prevent an illegal expression from being passed to the expression evaluator, you should define default values for variables in expressions (for example, when you configure formulas for evaluating X.509 Extensions).

Assume that the following formula has been configured: "{x} + {y} + {z}", and x=1, z=2 but "y" does not exist. The formula will resolve to "1 + + 2", which causes an error in the expression evaluator. You can prevent this type of error by defining a default value for the "y" term {y, 0} to ensure that the formula resolves to the legal and correct expression: "1 + 0 + 2".

Referencing the Cert Variable in an Attribute Assertion

The cert variable contains the raw data of the end entity X.509 certificate that is received in the certificate validation request. In the following example, this data is referenced in an Attribute Assertion statement to perform a binary compare of the certificate received in a request to the certificate returned from an Attribute Query named FindCert:

```
"{cert}" == "{attr[FindCert].userCertificate}"
```

Variables for Certificate Subject and Certificate Issuer

Variables for certificate validation definitions can reference attributes of the Distinguished Name (DN) parameter for the certificate subject or certificate issuer listed in the table on page 62. If the certificate subject or issuer parameter contains any of these attributes, you can reference the value of that attribute by using a variable in the format: {subject.*attrName*} or {issuer.*attrName*}, where *attrName* is an attribute in the preceding list. The variables in the following examples are valid representations of the CN attribute of a certificate subject and the user ID attribute of a certificate issuer:

{subject.CN}

{issuer.UID}

Using the Abbreviated Notation for Subject

Because attributes of the certificate subject are expected to be the most commonly used, you can abbreviate these attributes by omitting the subject component of the variable name, leaving the standalone attribute name. For example, {subject.cn} can be abbreviated as {cn}.

Variables for Distinguished Name

In addition to the individual attributes, you can reference the complete Distinguished Name (DN) by the variable name DN, for example, {subject.dn}. The DN string is always normalized for LDAP in the variable data. Specifically, if the DN begins with the Country or Domain Component attribute, the DN is reversed.

For example, if a certificate has the following Distinguished Name in the subject field:

```
C=US, ST=Texas, L=Irving, O=Sterling Commerce, CN=Example
```

the variable referenced by {subject.dn} is resolved to the following string:

```
CN=Example,O=Sterling Commerce,L=Irving,ST=Texas,C=US
```

Referencing Distinguished Name Attributes with Multiple Occurrences

If multiple occurrences of the same attribute occur within a Distinguished Name, you reference the various occurrences with a numeric subscript. Start with 0 and enclose the subscript in square braces to indicate which occurrence of the attribute you want to reference.

Note: This subscripting scheme is applied after any normalization for LDAP.

For example, the following subject DN has two occurrences of the OU attribute:

```
CN=example, OU=ou0val, OU=ou1val, C=US
```

The following example references the first occurrence of the OU attribute in the preceding example:

```
Subject.ou[0]
```

The following example references the second occurrence of the OU attribute:

```
Subject.ou[1]
```

The examples that follow show the OU attribute of the subject DN from the first example. In the following example, the Base DN is shown as configured, expressed in variables:

```
Cn={subject.cn}, ou={subject.ou[1]}, dc=my org, dc=com
```

The following example illustrates the Base DN with the variables resolved:

```
Cn=example, ou=ou1val, dc=my org, dc=com
```

Referencing a Relative Distinguished Name with a Multi-Valued Attribute

Each node within a Distinguished Name is a Relative Distinguished Name (RDN). Typically, the RDN consists of a single attribute name/value pair, with a textual representation: "*name*=*value*". However, you can include multiple attributes within a single RDN. This is represented (RFC 2253) by separating each name/value pair with the plus (+) symbol: "*name1*=*value1*+*name2*=*value2*".

To reference the individual attributes in a multi-valued RDN variable, use the following syntax: "*name1*+*name2*.*name1*" and "*name1*+*name2*.*name2*". For example, if a certificate subject contains the following multi-valued RDN: "cn=example+ou=multi-value", a base DN could be specified in the configuration as:

```
CN={subject[cn+ou].cn}, OU={subject[cn+ou].ou}, DC=my org, DC=com
```

The following example illustrates the base DN from the preceding example after the variable is resolved:

```
CN=example, OU=multi-value, DC=my org, DC=com
```

X.509 Extensions

Certificate extensions were introduced in version 3 of the X.509 standard. These v3 extensions allow certificates to be customized to applications by supporting the arbitrary fields in the certificate. X.509 v3 extensions provide for the association of additional attributes with users or public keys. Each extension, identified by its OID (Object Identifier), is marked as “Critical” or “Non-Critical,” and includes the extension-specific data.

X.509 Extensions and RFC 3280

After the introduction of X.509 v2 for Certificate Revocation Lists (CRL) and X.509 v3 for certificates, the Internet Engineering Task Force (IETF) has since adopted the standard documented in RFC 3280, “Internet X.509 Public Key Infrastructure -- Certificate and Certificate Revocation List (CRL) Profile.” The IETF adoption led to the standardization of several extensions; however, the customization that extensions allow is a source of interoperability issues.

EA supports the following standardized extensions:

Extension	Description
Key Usage	Defines the purpose of a key in a certificate.
Basic Constraints	Whether the subject of a certificate is a Certificate Authority (CA) and the maximum depth of a valid path for the certificate.
CRL Distribution Points	How Certificate Revocation List (CRL) information is obtained using the appropriate fields.

RFC 3280 requires that a system reject any critical extension that it does not recognize. To to prevent interoperability issues, EA provides the following mechanisms for extension support:

- Allow and require settings to explicitly allow and disallow specific extension to prevent failures from unrecognized critical extensions
- Boolean expressions for extension properties provide for application-specific interpretation and enforcement of extensions

Allow and Require Settings

EA provides support for an application to explicitly allow or disallow an extension in a certificate. If an extension that is disallowed appears in a certificate, the Certificate Validation Request fails. It allows an application to require that an extension appear in certificates. If a required extension is not included in a certificate, the Certificate Validation Request fails.

Boolean Expressions for Extension Properties

EA allows an application to have control of the interpretation and enforcement of a particular extension. EA allows you to configure a custom expression for each extension, which is evaluated at runtime. The expression is declared independently for client, server, and CA certificates so that different rules can be applied. In the EA user interface, this is done in the properties panel for the specific extension. In general, the property names are as follows:

“Client-*ExtensionName*”

“Server-*ExtensionName*”

“CA-*ExtensionName*” where *ExtensionName* is the extension’s actual name; for example: **Client-KeyUsage.**

Each property contains a default Boolean expression that you can modify or replace. If the expression is false, the certificate is rejected. Where applicable, each default expression enforces the rules specified for the extension in RFC 3280.

The Boolean expressions constructed for extension properties are modeled after the Java language, using Java operators, precedence rules, balanced parentheses for controlled precedence, and keywords such as true and false. Variables available in these expressions include all the variables from the certificate, such as subject and issuer attributes, plus variables specific to extensions.

Trivial Expressions: Keywords True and False

The simplest expressions consist of a single keyword:

true—Evaluation of the extension always succeeds, regardless of the data.

false—Evaluation of the extension always fails, regardless of the data.

Note: The keywords **true** and **false** must be written in lower case

Boolean Operators

You can use the following primary boolean operators:

Operation	Description
&&	Logical AND
||	Logical OR
!	Logical NOT
(	Begin grouping
)	End grouping

Extension Variables

The full name of any extension variable is “ext.*extensionName*.*variableName*”, for example, ext.keyUsage.keyCertSign. Depending on the reference, certain abbreviations are allowed:

Within its own extension definition: {*variableName*}

Within another extension definition: {*extensionName*.*variableName*}

Outside of extension definitions: {ext.*extensionName*.*variableName*}

Each extension has a Boolean variable, “isCritical”, which reflects the critical/non-critical designation of the extension within the certificate. The other extension variables are specific to the extension. In general, variables defined correlate directly to fields documented for the extension in the relevant RFC or reference document.

KeyUsage Extension

The KeyUsage extension defines the following variables, which correlate directly to the bit fields defined in RFC 3280 for the extension:

digitalSignature

nonRepudiation

keyEncipherment

dataEncipherment

keyAgreement

keyCertSign

cRLSign

encipherOnly

decipherOnly

Because the KeyUsage extension is a common area for problems with interoperability, the default formulas for KeyUsage specify a minimal set of rules that demonstrate the mechanics of the feature:

Client-KeyUsage: !({encipherOnly} && {decipherOnly})

Server-KeyUsage: !({encipherOnly} && {decipherOnly})

CA-KeyUsage: !({encipherOnly} && {decipherOnly}) && {keyCertSign}

The first two rules state that it is not legal to set both the encipherOnly and decipherOnly bits in the same certificate. The third rule adds that CA certificates must include the keyCertSign bit. Replace or modify the expressions to implement an application-specific policy for the key usage setting.

BasicConstraints Extension

The BasicConstraints extension is intended primarily for CA certificates. It has a single Boolean variable, “cA”, which reflects whether or not the certificate is a CA certificate. If the certificate is a CA certificate, it can also declare a pathLen constraint that dictates how many sub-CAs are allowed to exist in the hierarchy of CAs. The pathLen constraint is automatically enforced by EA.

The following expression is the default formula for CA certificates:

CA-BasicConstraints: {cA} && {KeyUsage.keyCertSign, false}

This default prevents problematic operation for many configurations. However, to enforce rules for the BasicConstraints extension as specified in RFC 3280, use the following formula:

CA-BasicConstraints: {isCritical} && {cA} && {KeyUsage.keyCertSign, false}

This rule states that CA certificates must designate the BasicConstraints extension as critical, set the CA indicator, and set the keyCertSign bit in the keyUsage extension.

CRLDistributionPoints Extension

EA supports the CRLDistributionPoints Extensions for identifying how to obtain certificate revocation list information. EA uses CRL definitions and the CRL information included in certificates to locate the appropriate directory and CRL.

When the CRLDistributionPoints extension references a CRL definition, it provides information for the CRL except for the following details that are always provided by the extension:

Directory Name distribution points—The DN specified in the extension overrides the Base DN specified in CRL definition and the scope is always set to Base.

URI distribution points—The protocol, host, port, and query in a CRL definition are overridden by the protocol, host, port, and query information for the URI specified in the extension. For LDAP this includes the Base DN, Scope, Match Attributes, and Return Attributes.

Properties

Properties configured for the CRLDistributionPoints extension deviate from the general “Client|Server|CA-*ExtensionName*” properties. Two properties are defined for configuration:

Ignore CRL Distribution Point—Instructs EA to ignore CRL Distribution Points encountered in end-entity certificates. This property can be set to the keywords **true** or **false**, or to a formula that evaluates to true or false. The CRL from an “ignored” distribution point will not be retrieved; however, the extension is still parsed. In fact, the data in the cRLDistributionPoints extension can be used in the formula to determine whether or not a particular distribution point is ignored.

Referenced CRL Definition—Allows a CRL definition to be associated with CRL Distribution Points found in end-entity certificates. This property should be set to the name of a previously-defined CRL definition. The name configured for this property can include variables so that different CRL definitions can be referenced based on, for example, the certificate issuer or the distribution point data.

CA - Ignore CRL Distribution Point—Instructs EA to ignore CRL Distribution Points encountered in CA certificates. This property can be set to the keywords **true** or **false**, or to a formula that evaluates to true or false. The CRL from an “ignored” distribution point will not be fetched; however, the extension is still parsed. In fact, the data in the cRLDistributionPoints extension can be used in the formula to determine whether or not a particular distribution point is ignored.

CA - Referenced CRL Definition—Allows a CRL definition to be associated with CRL Distribution Points found in CA certificates. This property should be set to the name of a previously-defined CRL definition. The name configured for this property can include variables so that different CRL definitions can be referenced based on, for example, the certificate issuer or the distribution point data.

Distribution Point Formats

CRL Distribution Points refers to a feature of the X.509 v2 CRL that allows a CA to partition its CRL into subsets, primarily in an effort to control the size of the CRL. The CA can then encode a cRLDistributionPoints extension into each certificate it issues to indicate the location of the distribution point(s) covering that particular certificate. The cRLDistributionPoints Extension defines several formats for publishing the address(es) of the distribution points. EA currently supports DirectoryName and UniformResourceIdentifier (URI).

DirectoryName Distribution Points

The DirectoryName must be the full distinguished name (DN) of the directory entry where the CRL resides. The directory hosting the distribution point must support LDAP access.

A Directory Name distribution point specifies an X.500 Distinguished Name, but not the location of the directory. EA uses one of two mechanisms to locate the LDAP server hosting the distribution point(s):

Through DNS-based automatic service discovery. For this to work, your environment must support service discovery, and the DN specified in the cRLDistributionPoints extension must include Domain Components (DC attributes).

Through configuration that is accomplished in two steps:

1. Create a CRL Definition that specifies the LDAP server address.
2. Set the “Referenced CRL Definition” property or the “CA - Referenced CRL Definition” property in the CRL Distribution Points configuration to the name you assigned to the CRL Definition in step 1.

At runtime EA uses the *Referenced CRL Definition*, overriding the Base DN configured (if any) with the DN specified in the cRLDistributionPoints extension, to find the distribution point CRL.

Note: All other fields specified in the CRL definition are used, including cache and connection settings.

URI Distribution Points

The URI must be a full LDAP, LDAPS, HTTP, or HTTPS URL.

Typically, it is not necessary to reference a CRL definition when the distribution point format is URI. However, if the server hosting the distribution point(s) requires authentication, you may need to configure log-on credentials in a CRL definition to be allowed access.

If this is the case, set the “Referenced CRL Definition” property or “CA - Referenced CRL Definition” property in the CRL Distribution Points configuration to the name of a CRL Definition you set up with the log-on credentials required to access the server. At runtime, EA uses the credentials from the *Referenced CRL Definition*, and any other properties configured (with the exception of the URL), to find the distribution point CRL(s). The URL is always obtained from the cRLDistributionPoints extension in the certificate.

You can also reference a CRL definition to use other settings, such as cache properties. As stated in the preceding example, any URL data configured in the CRL definition is overridden by the URL from the cRLDistributionPoints extension in the certificate.

Conditions for Using Variables with the CRLDistributionsPoints Extension

Variables are unnecessary if you do not use a CRL definition, a single CRL definition supports all distribution points, and the definition is referenced by name, without the use of variables.

Variables are necessary if you use multiple CRL definitions to access multiple directories or the cRLDistributionPoints data in the certificate does not represent the true address of the distribution point.

The CrlDistributionPoints Variable

The cRLDistributionPoints extension normally contains a single entry for one distribution point, but allows for multiple distribution points, each of which can contain multiple entries that designate alternate locations for finding the same distribution point CRL. To accommodate this possibility, EA stores distribution point entries in a two-dimensional array where the rows represent each distribution point and the columns represent each entry defined for a given distribution point.

The full variable name used to reference any given cRLDistributionPoint entry is:

{ext.crlDistributionPoints.distributionPoint[N1][N2]}

where N1 is the distribution point index and N2 is the index for the entries of a given distribution point. If there is a single distribution point entry in the certificate, then this name can always be abbreviated as {distributionPoint}.

You can also use this abbreviation to represent the current entry when referenced from one of the following:

The "Ignore CRL Distribution Point" property or "CA-Ignore CRL Distribution Point" property in the CrlDistributionPoints configuration.

The "Referenced CRL Definition" property or "CA-Referenced CRL Definition" property in the CrlDistributionPoints configuration.

Within the actual CRL Definition specified by the "Referenced CRL Definition" property in the CrlDistributionPoints configuration.

As EA iterates through each distribution point entry during cRLDistributionPoints processing, the variable {distributionPoint} always resolves to the current distribution point being processed. This abbreviated format will work in most cases.

Each *distributionPoint* variable (specified as either {distributionPoint} or {ext.crlDistributionPoints.distributionPoint[N1][N2]}) contains the actual distributionPoint data, which is a single *GeneralName* as specified in RFC 3280; such as a URI or directory name, depending on the type. The full GeneralName is specified by the distributionPoint variable name. Its parsed component parts can be specified by appending "*.partName*" to the end of the variable name. For directoryName distribution points, the parsed component parts are the DN attribute names, such as cn, plus "dn" to specify the complete DN normalized for LDAP.

For example, a single directoryName distribution point extension, "dc=com, dc=acme, ou=CA, cn=DP1" could be accessed in whole or in part by any of the following:

Variable Name	Value
{distributionPoint}	dc=com, dc=acme, ou=CA, cn=DP1
{distributionPoint.dn}	cn=DP1,ou=CA,dc=acme,dc=com
{distributionPoint.cn}	DP1
{distributionPoint.ou}	CA
{distributionPoint.dc[0]}	acme
{distributionPoint.dc[1]}	com

For URI distribution points, the parsed component parts are “protocol”, “host”, “port”, “path” and “query”. For example, the distribution point specified above could also be represented as the following URI: “ldap://svr:389/cn=DP1,ou=CA,dc=acme,dc=com?certificateRevocationList?base ?objectClass=cRLDistributionPoint.”

This distribution point could then be accessed in whole or in part by any of the following:

Variable Name	Value
{distributionPoint}	ldap://svr:389/cn=DP1,ou=CA,dc=acme,dc=com?certificateRevocationList?base?objectClass=cRLDistributionPoint
{distributionPoint.protocol}	ldap
{distributionPoint.host}	svr
{distributionPoint.port}	389
{distributionPoint.path}	/cn=DP1,ou=CA,dc=acme,dc=com
{distributionPoint.query}	certificateRevocationList?base?objectClass=cRLDistributionPoint

The distribution point type is also available from the distributionPoint variable by appending “.type”, “.typeName” or “.typeLongName” to the distributionPoint variable, as described in the following table:

Variable Name	Value for URI	Value for DirectoryName
{distributionPoint.type}	6	4
{distributionPoint.typeName}	URI	DN
{distributionPoint.typeLongName}	uniformResourceIdentifier	directoryName

Example of Multiple CRL Definitions

The following example applies if multiple CRL definitions are required as in the case where directoryName distribution points are spread across multiple directories that are not resolved automatically through referrals. For example, a CA with issuer name: “ou=CA, dc=acme, dc=com”, may have two directoryName distribution points:

DN=“cn=DP1, ou=CA, dc=acme, dc=com” Host=ldap1

DN=“cn=DP2, ou=CA, dc=acme, dc=com” Host=ldap2

To support this situation, set up two CRL definitions:

Name=“DP1-CrlDef” Host=“ldap1”

Name=“DP2-CrlDef” Host=“ldap2”

Then set the CrlDistributionPoints properties as follows:

Ignore CRL Distribution Point: false

Referenced CRL Definition: {distributionPoint.cn}-CrlDef

At runtime, EA resolves the variable “Referenced CRL Definition” to DP1-CrlDef or DP2-CrlDef, depending on the CN extracted from the distribution point DN in the extension, which allows EA to access the correct directory hosting the distribution point CRL. The example described below allows the use of the abbreviated distributionPoint variable.

Example of Distribution Point Variables

This example illustrates the need to use variables when the crlDistributionPoints data in the certificate do not represent the true address of the distribution point server, for instance, due to an address change. For example, a CA may have issued certificates with either of the following URI distribution points:

ldap://ldap1/cn=DP1,ou=CA,dc=acme,dc=com?certificateRevocationList?base?objectClass=cRLDistributionPoint

ldap://ldap2/cn=DP2,ou=CA,dc=acme,dc=com?certificateRevocationList?base?objectClass=cRLDistributionPoint

Due to a network reconfiguration, or some other reason, you may need to address these servers with their full DNS name, ldap1.acme.com or ldap2.acme.com. To support this, you can set up a single global CRL definition with the following URL specified:

ldap://{distributionPoint.host}.acme.com{distributionPoint.path}?{distributionPoint.query}

Additionally, set the CrlDistributionPoints property “Ignore CRL Distribution Point” to **true** to prevent access to the original, unreachable URI address specified for the LDAP servers in the distribution point URI.

At runtime, EA checks the global CRL and resolves the URL to one of the following, depending on the distribution point data:

ldap://ldap1.acme.com/cn=DP1,ou=CA,dc=acme,dc=com?certificateRevocationList?base?objectClass=cRLDistributionPoint

ldap://ldap2.acme.com/cn=DP2,ou=CA,dc=acme,dc=com?certificateRevocationList?base?objectClass=cRLDistributionPoint

Custom Extensions

The Custom Extensions feature is a mechanism provided in EA to allow X.509 v3 extensions unknown to the system to become known. EA will not process the extension, but can disallow or require the presence of the extension, and if appropriate, can accept an otherwise unknown critical extension. The Custom Extensions feature is also useful for the elimination of log file messages for unsupported extensions and for providing more meaningful debug-level log entries.

To register the extension with EA, it is only necessary to enter the OID of the extension and assign a name. The standard extension-handling options apply and are provided in the following list with their default settings:

Allow—True

Require—False

Properties:

- Client-*ExtensionName*—!{isCritical}
- Server-*ExtensionName*—!{isCritical}
- CA-*ExtensionName*—!{isCritical}

However, with the default settings allowing or requiring the presence of the extension (other than the effect on logging) is no different than if the extension were never registered. You may need to modify one or more of the Allow or Require settings, or modify properties. For instance, if the extension is marked critical, set the Client-*ExtensionName* formula to **true** to prevent the system from rejecting client certificates.

Create Users and Roles

EA can be used by administrators who have different network administration and configuration tasks to perform. Use the following procedures to customize the user and role definitions:

Manage Users

Manage Roles

Manage Users

User definitions identify users in EA. When you define users, you specify a user name and password and assign the user role. The admin role is the only role available for assignment initially; it enables all permissions by default. See *Create a Role Definition* on page 176 to create additional roles that enable you to allow only the required permissions for users.

Create a User Definition

To create or copy a user definition:

1. From the **Manage** menu, select **Users**.
2. On the External Authentication User Definitions window, click + and specify the following parameters:
   - Name
   - Password
   - Confirm Password
   - Role
   - Description
   - Properties
3. Click **Save**.

Change a User Definition

To change a user definition.

1. From the **Manage** menu, select **Users**.
2. Double-click the user definition to edit.
3. On the **Update User** dialog box, update the user definition.
4. Click **Save**.

Delete a User Definition

You cannot delete a user that is currently logged in.

To delete a user definition:

1. From the Manage menu, select **Users**.
2. Select the user definition to delete and click −.
3. Click **OK**.

Manage Roles

The admin role is predefined and is the only role you can assign to users initially. By default, the admin role allows all permissions for users assigned the role. Create additional roles to allow only the required permissions for users assigned that role.

Note: The user roles that exist in EA include the anon role. The anon role is used by incoming client applications that request certificate validation and cannot be assigned to users.

Create a Role Definition

You can create new roles and allow EA users to create, read, update, delete, and execute permissions in the functional areas.

To create a role definition and set permissions:

1. From the **Manage** menu, select **Roles.**
2. On the External Authentication Role Definitions screen, click +.
3. On the **Add Role** dialog box, specify the following parameters:
   - Role name
   - Permissions
   - Select All
   - Cert Validation
   - Cert Revocation
   - Authentication
   - Accepter
   - User
   - Role
   - System

4. Click **OK** to save the role.

Change a Role Definition

To change the definition of a role:

1. From the **Manage** menu, select **Roles**.
2. Select the role definition to edit and click 📝.

3. On the **Update Role** dialog box, update the role definition:
4. Click **OK** to save the changes.

Delete a Role Definition

You cannot delete a role that is assigned to a user who is currently logged in.

To delete a role definition.

1. From the **Manage** menu, select **Roles**. The External Authentication Role Definitions screen is displayed with a list of the role definitions.
2. Select the role definition to delete and click −.
3. Click **OK**.

Customize Layout Views

You can view a variety of information for the definitions displayed. Each definition window has a default view, but you can also customize views by performing the following actions:

- Display or hide the columns you select
- Rearrange columns in an order that is important to you
- Save a view for future use
- Rename a view
- Delete a view

Hide Columns

To hide a column, right-click the column to hide and click **Hide Column**.

To hide one or more columns:

1. Right-click the column heading and click **Manage Columns**.
2. Move the columns you want to hide to the **Available Columns** list using the arrow buttons.

Restore Columns

To restore a hidden column:

1. Right-click the column heading and click **Manage Columns**.
2. Move the columns you want to restore to the layout by moving them from the **Available Columns** list using the arrow buttons.

Manage Columns

To create a custom view:

1. Right-click a column and click **Manage Column**.
2. To add a column, select the column to be added in the **Available Columns** list and click **>**.
3. To remove a column, select the column to remove in the **Selected Columns** list and click **<**.
4. To rearrange columns, select the column to rearrange in the **Selected Columns** frame, and click the up or down arrow to move it to a new location.

   Columns appear in the layout in the order in which they appear in the Selected Columns list.
5. Click **OK**.

Rearrange and Resize Columns

You can rearrange the order of columns in a view by dragging a column heading to the desired position. You can also change the width of a column by dragging a column heading border until the column is at the desired width. These settings are saved for each user.

Save a Column Layout

Saving a column layout enables you to change the arrangement of columns and see details that are relevant for certain tasks. First rearrange, resize, and hide columns to create an alternate view, then save the column layout with a descriptive name.

To save a new column layout:

1. Right-click any column and click **Save Layout**.
2. Type a name for the new layout in **Layout Name**.
3. Click **OK**.

Select a Column Layout View

To select a column layout that you have defined, right-click any column and select **Select Layout >** ***name of customized layout***.

Manage Column Layout Views

You can rename a column layout view or delete a column layout view from the Manage Layout window.

Rename a Column Layout View

To rename a column layout:

1. Right-click any column and select **Manage Layouts**.
2. Select the layout you want to rename and click **Rename**.
3. Type a name for the layout in the **Layout Name** field and click **OK**.
4. Click **Close**.

Delete a Column Layout View

To delete a column layout:

1. Right-click a column and select **Manage Layouts**.
2. Select the layout you want to remove and click **Remove**.
3. Click **Close**.

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.

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 ficticious 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 2010. Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. 2010.

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

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: *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, Microsoft Windows NT, and the Microsoft 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.