

Sterling External Authentication Server

Implementation Guide

Version 2.2



Sterling External Authentication Server Implementation Guide **Version 2.2**

First Edition

(c) Copyright 2005-2009. Sterling Commerce, Inc. All rights reserved. Additional copyright information is located at the end of this document.

STERLING COMMERCE SOFTWARE

*****TRADE SECRET NOTICE*****

THE STERLING EXTERNAL AUTHENTICATION SERVER SOFTWARE (“STERLING COMMERCE SOFTWARE”) IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.

This documentation, the Sterling Commerce Software it describes, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of Sterling Commerce, Inc., its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Sterling Commerce entity. This documentation and the Sterling Commerce Software that it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright notice. As and when provided to any governmental entity, government contractor or subcontractor subject to the FARs, this documentation is provided with RESTRICTED RIGHTS under Title 48 52.227-19. Further, as and when provided to any governmental entity, government contractor or subcontractor subject to DFARS, this documentation and the Sterling Commerce Software it describes are provided pursuant to the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation.

These terms of use shall be governed by the laws of the State of Ohio, USA, without regard to its conflict of laws provisions. If you are accessing the Sterling Commerce Software under an executed agreement, then nothing in these terms and conditions supersedes or modifies the executed agreement.

Where any of the Sterling Commerce Software or Third Party Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14, and FAR 52.227-19(c)(1-2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation including DFAR 252.227-7013, DFAR 252,227-7014, DFAR 252.227-7015 and DFAR 252.227-7018, all as applicable.

The Sterling Commerce Software and the related documentation are licensed either “AS IS” or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

Connect:Direct is a registered trademark of Sterling Commerce. Connect:Enterprise is a registered trademark of Sterling Commerce, U.S. Patent Number 5,734,820. All Third Party Software names are trademarks or registered trademarks of their respective companies. All other brand or product names are trademarks or registered trademarks of their respective companies.

Sterling Commerce, Inc.

4600 Lakehurst Court Dublin, OH 43016-2000 *
614/793-7000

Contents

Chapter 1 About Sterling External Authentication Server	9
Sterling External Authentication Server Operation	10
Certificate Validation Steps	10
User Authentication and Authorization Steps	11
Interaction with Sterling Secure Proxy (SSP) and Connect:Direct Secure+ Option	12
About the EA GUI	13
File Naming Guidelines	13
Administration and Navigation from the GUI	14
Check and Confirm Parameters	14
Definition of Security Terms	14
Certificate Validation (CV) Definitions and Authentication Definitions	16
Elements of CV Definitions	17
Elements of Authentication Definitions	18
Prerequisite Tasks for Establishing Secure Connections	18
Tasks Required to Use CA-Issued Certificates	19
Tasks Required to Use Self-Signed Certificates	19
Chapter 2 Install and Start EA on UNIX	21
Review Resources	21
Installation and Startup Checklist	22
Install EA on UNIX	22
Start the EA Server on UNIX	23
Start the EA Server on UNIX Using a Stored and Encrypted Passphrase	24
Start the EA Server on UNIX and Require a Passphrase	24
Start the EA GUI On UNIX	24
Start the EA GUI From the Computer Where the EA GUI Is Installed	25
Start the EA GUI from a Remote Computer	25
Login Fields	26
Log Off	26
Shut Down EA on UNIX	26
Chapter 3 Install and Start EA on Windows	27
Review Resources	27

Installation and EA Startup Checklist	28
Install EA on Windows	28
Start the EA Server on Windows	29
Start the EA Server on Windows Using a Stored Passphrase	29
Start the EA Server on Windows And Require a Passphrase	30
Start the EA GUI on Windows	30
Start the GUI from the Local Windows Computer	30
Start the GUI from a Remote Computer	31
Login Fields	31
Log Off	32
Shut Down EA on Windows	32

Chapter 4 Configure System Resources 33

Modify the Non-Secure Listener Port	33
Disable the Non-Secure Listener Port	33
Change the Port Number of the Servlet Container	34
Change the Admin Password	34
Configure Logging Options	35
Change the Logging Level from the GUI	35
Turn Logging to the Console On or Off	36
Change the Maximum Log File Size	36
Change the Maximum Number of Archive Log Files	37
Change the Logging Level in a Logging Properties File	37
Refresh GUI Lists from the Server	38
Set Listener Connection Settings (Backlog and Timeout)	38
Create a System-Wide LDAP or HTTP Connection Definition	39

Chapter 5 Create and Manage System Certificates 41

Procedures to Generate and Use Certificates	41
Procedures to Generate a Self-Signed Certificate to Secure the Connection to the EA Server	41
Procedures to Generate a CA-Issued Certificate to Secure the Connection to the EA Server	42
Procedures to Generate a Self-Signed Certificate to Secure the Connection between the GUI and the EA Server	43
Procedures to Generate a CA Certificate to Secure the Connection between the GUI and the EA Server	43
Generate a Self-Signed Certificate	44
Generate a Self-Signed Certificate for the EA Server	44
Generate a Self-Signed Certificate for the GUI	46
Parameters to Generate a Self-Signed Certificate	47
Export a Self-Signed Certificate for the EA Server or the GUI	47
Export a Self-Signed Certificate for the EA Server	48
Export a Self-Signed Certificate for the GUI	48
Commands to Export a Self-Signed Certificate	48
Create Certificate Signing Request	49
Create a PKCS#10 Certificate Signing Request for the Server	49
Create a PKCS#10 Certificate Signing Request for the GUI to Submit to a CA	49
Parameters to Create a CSR	50

Import the CA-Issued Certificate Keystore	50
Import the CA-Issued Certificate into the Server Keystore	51
Import the CA-Issued Certificate to the GUI Keystore	51
Parameters to Import the CA-Issued Certificate into the Keystore	52
Import Certificates into the Trust Store	52
Import a Certificate into the EA Server Trust Store	52
Import the Server Certificate into the GUI Trust Store	53
Parameters to Import the Certificate into the Trust Store	53
Configure EA to Access the SSL Keystore	53
Configure EA to Access the SSL Trust Store	54
Configure the Secure Connection Listener	55
Configure SSL or TLS Between the GUI and the EA Server	55

Chapter 6 Create and Manage Certificate Revocation Lists (CRL) Definitions 57

Create a CRL Definition	57
Edit a CRL Definition	61
Copy a CRL Definition	61
Delete a CRL Definition from the Manage Menu	62

Chapter 7 Create and Manage Certificate Validation (CV) Definitions 63

Create a CV Definition	63
Configure and Test a Custom Exit for a CV Definition	65
Prerequisites for Using a Custom Exit	65
Develop and Deploy a Custom Exit Class in Java	66
Specify Java Class in a CV Definition	66
Specify an Operating System Command for a Custom Exit	67
Specify Certificate Subject Verification for an Attribute Query	68
Reference a CRL Definition	70
Configure Supported Extension Use for a CV Definition	70
Create and Configure a Custom Extension for a CV Definition	71
Edit or Copy a CV Definition	71
Edit a CV Attribute Query Definition	72
Edit a CV Attribute Assertion Definition	72
Edit Supported Extensions in a CV Definition	73
Edit Custom Extensions in a CV Definition	73
Delete a Certificate Validation Definition	74
Manage CV Attribute Query Definitions	74
Copy a CV Attribute Query Definition	74
Delete a CV Attribute Query Definition	74
Manage CV Attribute Assertion Definitions	75
Copy a CV Attribute Assertion Definition	75
Delete a CV Attribute Assertion Definition	75
Manage Custom Extensions in a Certificate Validation Definition	75
Add a Custom Extension to a CV Definition	76
Delete a Custom Extension in a CV Definition	76

Chapter 8 Create and Manage LDAP Authentication Definitions 77

Create an LDAP Authentication Definition 77

Create an Application Outputs Definition for an LDAP Authentication Definition . . . 80

 Prepare the Directory for Use with Lookup Login Credentials 81

 Extend the Schema for OpenLDAP 81

 Extend the Schema for IBM Tivoli Directory Server 81

 Create Entries for Login Credentials 81

 Map Query Return Attributes to Application Output Names in an Application Outputs Definition 82

Edit or Copy an LDAP Authentication Definition 83

Delete an Authentication Definition 84

Chapter 9 Create and Manage SSH Key Authentication and Mapping Definitions 85

Create an SSH Key Authentication Definition 86

Create an SSH Application Output Definition for an SSH Authentication Definition . . 87

 Create an Application Output Definition for the loginCredentials (sterling) Definition 87

 Create an Application Output Definition for the loginCredentials (custom) Definition 87

Edit or Copy an SSH Key Authentication Definition 88

Delete an SSH Key Authentication Definition 89

Prepare the Directory to Store Keys, a User ID, and Password for an SSH User . . . 89

Implement the SSH and SCI Schemas for Open LDAP 90

Implement the SSH and SCI Schemas for IBM Tivoli 90

Create Entries for SSH Public Keys 90

Create Entries for Login Credentials 91

 LDIF Entry Example 92

Chapter 10 Create Generic Authentication Definitions 93

Create a Generic Authentication Definition 93

Configure and Test a Custom Exit for a Generic Authentication Definition 94

 Prerequisites for Using a Custom Exit 94

 Develop and Deploy a Custom Exit Class in Java 94

 Specify a Java Class for a Custom Exit in a Generic Authentication Definition . . 95

 Specify an Operating System Command for a Custom Exit 95

Create an Application Output Definition for a Generic Authentication Definition . . . 97

Chapter 11 Perform Gentran Integration System (GIS) User Authentication through an EA Custom Exit 99

Prepare the Certificates for Authentication in the GIS User Store 99

 Configure the HTTP Server Adapter Certificate 100

 Export the System Certificate from GIS 100

 Import the HTTP Server Adapter System Certificate into the EA Trust Store . . . 101

 Export a Keystore from the EA Keystore 101

 Import the EA System Certificate into the GIS CA Certificate Store 102

Configure a GIS HTTP Server Adapter for EA Support 102

Configure an EA User Authentication Profile 103

 Custom Exit Configuration Properties 103

Log Messages	104
Sterling Secure Proxy Messages	104
EA Server Messages	104
GIS Authentication Log Messages	105
Chapter 12 Create and Manage Tivoli Access Manager (TAM) Authentication Definitions	107
Authenticating with Tivoli Access Manager	107
Prerequisites for Tivoli Access Manager Authentication	107
Logging Information for Tivoli Access Manager Authentication	107
Create a Tivoli Access Manager Authentication Definition	108
Create an Application Output Definition for TAM.	110
Edit or Copy a TAM Authentication Definition	111
Delete an Authentication Definition	111
Chapter 13 Create and Manage Attribute Queries and Assertions	113
Create and Manage Attribute Query Definitions	113
Specify General Details for a Query	113
Specify Query Parameters	115
Specify Match Attributes	116
Specify LDAP Connection Settings	117
Specify JNDI Properties for a Connection.	118
Edit an Attribute Query Definition	118
Create and Manage Attribute Assertion Definitions.	119
Create an Attribute Assertion Definition	119
Edit an Attribute Assertion Definition.	120
Chapter 14 CV and Authentication Definition Variables	121
Syntax and Rules	121
Variables for Certificate Validation Requests	122
Referencing the Cert Variable in an Attribute Assertion	124
Variables for Certificate Subject and Certificate Issuer.	124
Using the Abbreviated Notation for Subject	124
Variables for Distinguished Name	124
Referencing Distinguished Name Attributes with Multiple Occurrences	125
Referencing a Relative Distinguished Name with a Multi-Valued Attribute.	125
Variables for Authentication Requests	126
Chapter 15 X.509 Extensions	129
X.509 Extensions and RFC 3280	129
Allow and Require Settings.	130
Boolean Expressions for Extension Properties.	130
KeyUsage Extension	131
BasicConstraints Extension.	132
CRLDistributionPoints Extension.	132

Properties	133
Distribution Point Formats	133
Conditions for Using Variables with the CRLDistributionsPoints Extension	134
Custom Extensions	137
Chapter 16 Manage Users and Roles	139
Manage Users	139
Create a User Definition	139
Change a User Definition	140
Delete a User Definition	140
Manage Roles	141
Create a Role Definition	141
Change a Role Definition	143
Delete a Role Definition	143
Chapter 17 Customize Layout Views	145
Hide Columns	145
Restore Columns	145
Manage Columns	146
Rearrange and Resize Columns	146
Save a Column Layout	146
Select a Column Layout View	146
Manage Column Layout Views	147
Rename a Column Layout View	147
Delete a Column Layout View	147
Index	149

About Sterling External Authentication Server

Sterling External Authentication Server (EA) allows you to implement extended authentication and validation services for Sterling Commerce products, referred to as client applications. EA includes a server that client applications connect to and a GUI that you use to configure EA requirements.

For SSL or TLS authentication, the connection between EA and the client application is authenticated. Then, the client application sends a request that contains a certificate chain and/or a user ID and password. EA uses the certificate validation or authentication definition that corresponds to the profile name referenced in the request to perform the requested operations.

For SSH authentication, the client application sends a request to EA that contains a profile name, user ID, or SSH public key. EA uses the configuration information in the profile to bind to an LDAP directory and look up the SSH key assigned to the user. It also performs an attribute assertion to match the key provided against the list of keys found in the LDAP directory.

After you install EA, configure it for operation in your environment. EA supports a flexible configuration to meet a variety of certificate validation and user authentication and authorization needs. You can configure:

- ◆ TCP ports (listeners)
- ◆ SSL/TLS protocol operation
- ◆ System-wide server connections
- ◆ Logging operation
- ◆ Other global system parameters

After you configure the system, create certificate validation and user authentication definitions.

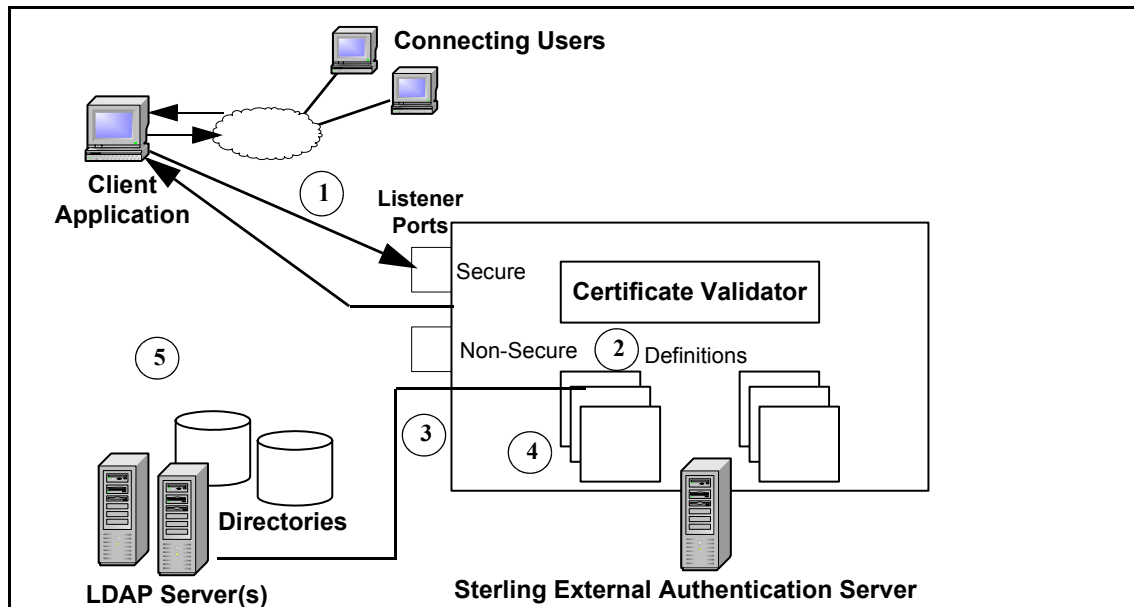
- ◆ A certificate validation definition specifies validation of certificates against certificate revocation lists (CRLs) and allows validation using attribute queries and assertions. It can include validation using a custom exit to a Java class or an operating system command (for running a program or script).
- ◆ Authentication definitions configure multifactor authentication using SSL client certificates, SSH keys, user ID and password, and client IP address as factors. They also enable application outputs to allow you to map attributes, such as login credentials that are returned to a query, to outputs you specify.

Sterling External Authentication Server Operation

EA responds to a request from a client application and performs certificate validation and user authentication as described in the following sections.

Certificate Validation Steps

The following diagram shows the interaction between a client application, EA, and directories accessed through LDAP servers.



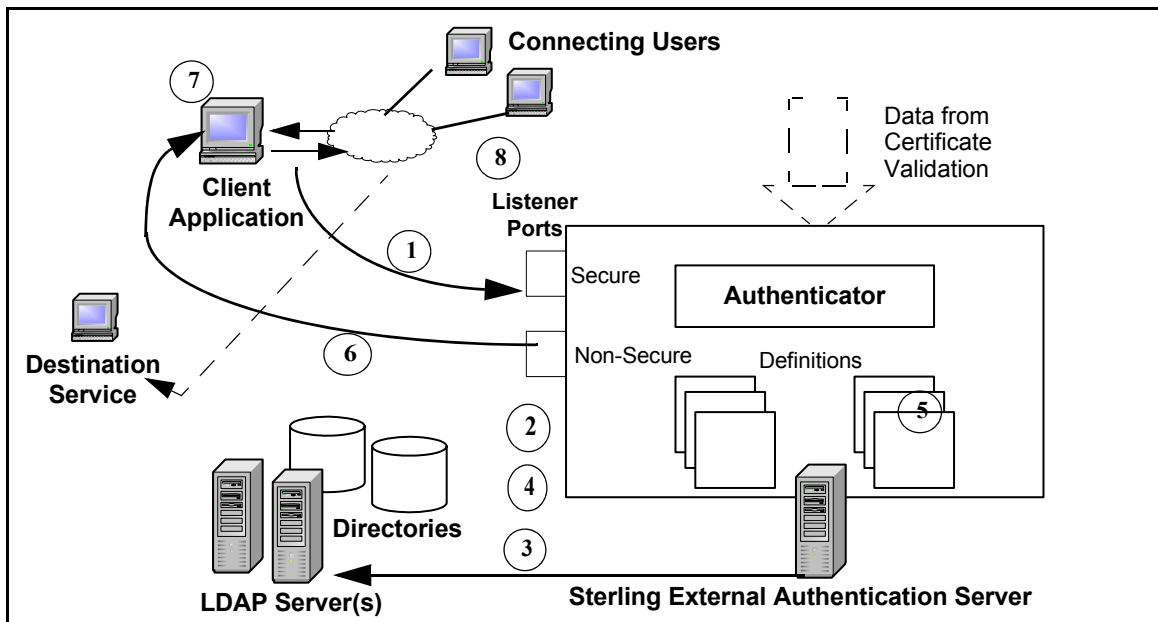
Following are the steps in the preceding diagram:

Step	Description
1	A client application sends a request message to EA and passes a certificate chain. Typically, the certificates in the chain are from a user who connected to the application using SSL. The client authenticates itself to EA as required and specifies the profile (definition) to use for certificate validation (CV). If the connection is made to the secure listener port, mutual authentication and encryption are used to secure all messages flowing through the connection.
2	The certificate validator references the CV definition specified by the client application and performs the validation steps. The CV definition can include LDAP server connection definitions, certificate revocation list (CRL) definitions, attribute query definitions, and attribute assertion definitions. If a custom exit is specified in the CV definition, a Java class or operating system command is also used to validate a certificate.
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.

Step	Description
4	EA verifies attribute query results, attributes from the end-user certificate, and other requested data as specified in the CV definition.
5	EA sends a response message to indicate the success or failure of a CV.

User Authentication and Authorization Steps

The following diagram illustrates the interaction between a client application, EA components, and directories accessed through LDAP servers:



End users typically connect securely to the application that acts as a client to Sterling External Authentication. Following are the steps in the preceding diagram:

Step	Description
1	A client application sends a user ID and password to EA, from a user attempting to log in to an application or access a destination service. The client authenticates itself to EA and specifies the definitions to use. If the connection is made to the secure listener port, mutual authentication and encryption secure all messages flowing through the connection.
2	EA references the authentication definition specified by the client application. It includes the LDAP server connection definitions, attribute query definitions, attribute assertion definitions, and/or application output definitions required to authenticate and/or authorize the connection.

Step	Description
3	EA connects to the LDAP server specified in the authentication definition. The user ID and password from the request is validated and other tasks, such as performing LDAP attribute queries and assertions, are performed to respond to the request. For example, attribute query definitions in an authentication definition could include information needed to locate a user ID entry, validate group membership, and look up login credentials to pass to the client application.
4	EA uses results from the directory to determine if the user should be authenticated to an application or authorized for access to a requested destination service. When EA is authenticating a user as a continuation of certificate validation, variables and information established during certificate validation are available for authentication.
5	An authentication definition can include application output definitions to specify how return attributes from a query map to outputs that are passed to the client application. When an application output definition is included, the mapping of return attributes is performed.
6	EA sends a response to indicate the results of user authentication. If authentication is successful, the response can include credentials. For example, EA can provide the user ID and password returned from a query in an application output definition as part of the response message.
7	If the client application is a proxy for the destination service, it logs in to the destination service with the credentials retrieved by EA.

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

EA enhances the security functions of Sterling Secure Proxy (SSP) and Connect:Direct Secure+ Option (Secure+). For example, in response to a CV request, EA can extract data from the certificate chain and use the specified CV definition to connect to an LDAP server and validate the certificate subject and that a digital certificate has not been revoked. EA uses authentication definitions to authenticate and authorize users. After receiving an authentication or authorization request, EA can access an entry in an LDAP directory to determine that a user ID and password are valid for access to a destination service or application.

The server component of the EA application receives and performs processing requests from SSP to validate certificates and authenticate users; it validates certificates for Secure+ client requests. The server accepts requests on a secure listener port and on a nonsecure listener port. All SSL/TLS connections are established through the secure listener port. The nonsecure port is used for testing, or when EA is installed behind the DMZ and the connection request originates from a client that is deployed in the trusted zone of your network. The nonsecure port can be disabled after the secure port is set up.

In a typical scenario, SSP establishes a secure session with EA to validate the identity of an external client attempting to connect to a Web application, destination service, or a Connect:Direct node using a proxy connection and to validate that the connection is authorized. Based on whether the Sterling Secure Proxy client references a CV definition (referred to as profile in the SSP

application) or an authentication definition, the server initiates a secure client connection to an LDAP server and queries the directory to verify any or all of the following information specified for the proxy connection:

- ◆ The digital 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 connecting user match the UID attribute specified for the user account on the LDAP server.
- ◆ Attribute queries or attribute assertions defined in the definition referenced in the request can be validated using information stored in the LDAP directory.
- ◆ The originating IP address of the connection request is valid for initiating a proxy connection and that the client connection to Sterling Secure Proxy actually originated from an IP address valid for the organization.

If EA confirms that the credentials and documents submitted for verification are valid, it returns a success message to the SSP client and SSP completes the connection request; otherwise, the connection fails.

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

About the EA GUI

The EA GUI is the interface for configuring system functions and definitions (client application profiles) that specify how to process a request from Secure+ or SSP.







File Naming Guidelines

Definition names can be up to 255 characters, and can include any alphanumeric character and space, underscore (_), and period (.). They cannot begin or end with a space because EA discards definition names that begin or end with a space. The following examples demonstrate valid use of special characters in definition names:

- ◆ Routing Names
- ◆ corpnet.ldap.home_server234
- ◆ Cert_Subject_Romuli8Query

Administration and Navigation from the GUI

After creating a definition, you can edit, copy, and delete the entire definition, or copy, edit, and delete definitions that comprise it. The following icons and buttons direct your progress as you create or change definitions:

Icon or Button	Description
	Add a definition or component of the type listed in the window. Click to start the wizard or display the first screen.
	Delete the selected definition or component from the list.
	Copy the definition or components listed on the screen above the icon.
	Review or change the properties of the selected definition or a component. For an authentication or CV definition, clicking this button displays a tabbed list of components or areas of functionality.
	Display help.
	Display a dialog box for entering related details, such as property name and value pairs, for JNDI properties or for the match attributes used with an attribute query.

Check and Confirm Parameters

Because definitions can include a variety of elements, check that all parameters are set correctly before you save the definition. After you create a definition, review the list of parameters displayed on the Confirm screen. If you find a parameter that is not correct on the Confirm screen, click **Back** to navigate through parameters. Access Help as needed to make the corrections required. After you make corrections, click **Next** to move forward to the Confirm screen and save. When you edit a definition, review the parameters listed on the **Summary** tab to identify the area that requires a change. Then, click the appropriate tab to make corrections to the parameters that are editable.

Definition of Security Terms

Following are security terms used in EA:

Term	Definition
Self-Signed Certificate	Digital document that is digitally 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, you must exchange certificates with all your trading partners.

Term	Definition
Simple Authentication	Authentication by sending the fully-qualified DN and clear-text password of the user named in a request. Simple authentication can be used on an encrypted channel.
CA-Signed Certificate	Digital document issued by a certificate authority (CA) that binds a public key to the identity of the certificate owner, to allow the certificate owner to be authenticated. 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's identity and the services that the owner is authorized to use, issues new certificates, renews existing certificates, and revokes certificates belonging to users who are no longer authorized to use them. The CA digital signature is assurance that anybody who trusts the certificate signed by that CA can also trust that the certificate is an accurate representation of the certificate owner.
Certificate Signing Request (CSR)	Message sent from an applicant to a CA in order to apply for a CA-signed certificate. Before creating a CSR, the applicant first generates 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)	A list of certificates that have been suspended or revoked before the scheduled expiration date. A certificate revocation list (CRL) usually indicates the CRL issuer's name, the date of issue, the date that the CRL will be reissued, the serial numbers of revoked or suspended certificates, and the times and reasons certificates were revoked or suspended.
Distinguished Name (DN)	A unique name for a directory entry that includes the list of names of parent entries back to the root entry for the directory.
Public Key	Public part of a complementary public-private key pair. The asymmetric cipher of the public key is used to encrypt 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 complementary public key. Data that is encrypted with a public key can only be decrypted using its complementary 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. It is used to validate that the public key received during negotiation of a SSL/TLS session was signed by the CA to verify the identity of the client requesting the connection.

Term	Definition
Keystore	File that contains the private keys and matching key certificates that EA uses for SSL and TLS sessions. Each key/certificate pair in the keystore has an associated alias. The secure listener and connection definitions that specify SSL/TLS use the alias to reference the key/certificate.
Trust store	The trust store includes the following digital certificates: <ul style="list-style-type: none"> ◆ All the trusted CA or self-signed certificates of the client applications EA communicates with over the secure listener. ◆ All the trusted CA or self-signed certificates of the secure servers EA communicates with, including HTTPS, LDAPS, and LDAP v3 Start TLS. ◆ All the trusted CA certificates needed by the Certificate Validation Service when validating requests from client applications.
Principal	The name that identifies a user or service. The principal is the name EA needs to authenticate. Principal can also refer to the name EA uses when authenticating to another server. For example, EA uses a principal to authenticate to an LDAP server to search for a user entry.
LDAP	Lightweight Directory Access Protocol. An open industry standard that defines a set of rules for the messages used by directory clients and directory servers. LDAP is used to locally or remotely access and update information in a directory. EA can operate as a client of an LDAP directory as it provides a requested service, performing actions such as authenticating user credentials, validating the certificate subject or checking for the certificate in a list of certificates revoked before expiration.

Certificate Validation (CV) Definitions and Authentication Definitions

EA uses definitions you create to process requests to validate certificates and authenticate or authorize users. Certificate validation (CV) definitions are configuration files that define how EA validates certificates. Authentication definitions are configuration files that define how EA authenticates users and verifies authorization to access an application or destination service.

Requests from client applications reference the name assigned to a CV definition or an authentication definition in EA. For example, SSP uses a profile name that must exactly 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 digital certificate presented by a client application on behalf of an end user. It can include the following optional elements:

Element	Description
Attribute query definition	<p>Specifies an LDAP search operation to locate directory entries and optionally return attributes from those entries. The search must succeed for certificate validation to succeed. The query is composed by specifying all query parameters in a Uniform Resource Locator (URL), or by specifying parameters individually on the Query Parameters screen.</p> <p>Attribute query definitions can include variables as described in Chapter 14, <i>CV and Authentication Definition Variables</i>.</p>
Attribute assertion definition	<p>Specifies a Boolean statement that must evaluate as true in order for certificate validation to succeed. Attribute assertions allow the specification of additional conditions and can compare details from the request (such as an IP address or attributes from a certificate) to fixed data or to attributes returned from queries.</p> <p>Attribute assertion definitions can include variables as described in Chapter 14, <i>CV and Authentication Definition Variables</i>.</p>
Custom exit	<p>Specifies details for exiting from an EA or authentication definition to perform related tasks using a Java class on a script or program executed by running an operating system command.</p>
Certificate revocation list (CRL) definition	<p>Specifies how to access a CRL which is a list of identified certificates that have been suspended or revoked before the scheduled expiration date. After creating a CRL definition, one or more of the defined CRLs can be referenced so that they are checked during certificate validation. A CRL usually indicates the CRL issuer's name, the date of issue, the date that the CRL is next scheduled to be reissued, the serial numbers of revoked or suspended certificates, and the number of times and reasons certificates were revoked or suspended. When EA is validating a certificate, if that certificate is found on a CRL, certificate validation fails. Certificate revocation list definitions can be created independently of the CV definition and referenced in multiple CV definitions.</p>
Supported extensions	<p>Defines processing instructions for the set of X.509 v3 extensions directly supported for EA.</p>
Custom extensions	<p>Registers and defines processing instructions for X.509 v3 extensions that are unknown to EA.</p>

When you create a CV definition, you can configure as many of the optional elements allowed within it. See Chapter 7, *Create and Manage Certificate Validation (CV) Definitions*, for instructions. You can also add any of the optional elements at a later time. See Chapter 13, *Create and Manage Attribute Queries and Assertions*, for instructions to create, copy, and edit attribute queries and assertions used for CV definitions.

Elements of Authentication Definitions

An authentication definition specifies how EA authenticates a user of a destination service. The authentication definition specifies how to use attributes associated with the user specified in a request. In particular, it specifies a user ID and password to use to authenticate and optionally authorize the user. An authentication definition can include the following optional elements:

Element	Function
Attribute query	Specifies an LDAP search operation for locating directory entries and optionally returning attributes from those entries. The search must succeed for authentication to succeed. The query is composed by specifying all query parameters in a Uniform Resource Locator (URL), or by specifying parameters individually on the Query Parameters screen. Attribute query definitions can include variables as described in Chapter 14, <i>CV and Authentication Definition Variables</i> .
Attribute assertion	Specifies a Boolean statement that must evaluate as true in order for authentication to succeed. Attribute assertions allow the specification of additional conditions and can compare details from the request (such as a user ID or destination service) to fixed data or to attributes returned from queries. Attribute assertion definitions can include variables as described in Chapter 14, <i>CV and Authentication Definition Variables</i> .
Applications outputs	Enables use of a directory object in association with an attribute query to map the query return attributes to an output name that is known by the client application. This is used primarily for looking up login credentials to be passed back to the client application, typically to log in to the destination service.
Custom exit	Specifies details for exiting from a EA generic authentication definition to perform related 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 that you customize with a custom exit, attribute query, or attribute assertion. Within an authentication definition you can create any or none of the optional elements. See Chapter 8, *Create and Manage LDAP Authentication Definitions*, Chapter 10, *Create Generic Authentication Definitions*, or Chapter 12, *Create and Manage Tivoli Access Manager (TAM) Authentication Definitions*, for instructions on creating optional elements.

Prerequisite Tasks for Establishing Secure Connections

As a prerequisite to establishing secure communications sessions using the Secure Sockets Layer (SSL) protocol or the Transport Layer Security (TLS) protocol between the entities that EA communicates with as server and as client, your security administrator should determine whether

your security policy requires using self-signed certificates, CA-issued certificates, or a combination of 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 Chapter 5, *Create and Manage System Certificates*, for instructions on generating and storing both types of 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 CA root certificate (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, you must complete all the procedures listed here for using CA-issued certificates for both the GUI and the server.

Tasks Required to Use Self-Signed Certificates

To use self-signed certificates, you must 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 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, or import the CA root certificate.
- ◆ To establish a secure connection to EA from the GUI when it is running on a remote computer, you must complete all the procedures listed here for using self-signed certificates for both the GUI and the server.

Install and Start EA on UNIX

Use the following procedures to verify installation resources, install EA, and start the EA server and the GUI.

Review Resources

EA provides certificate validation and user authentication over a network, in security environments that are configured based on organizational needs and policies. Before you install EA, review any network- and security-specific configuration details that are relevant for the EA server and GUI. You may need to consider details that are environment specific.

Refer to the following list of resources as you plan network and security related resources for installing and configuring EA:

Installation or Configuration Resource	EA Usage
TCP Ports	Use available port numbers, in appropriate port ranges to set the secure listener, non-secure listener, and the servlet container port used to download the GUI.
Network Interface Addresses	Confirm the local bind address of a network interface you want to use for a connection.
LDAP Directory Information Tree	Apply related knowledge when selecting and specifying LDAP attribute query and assertion parameters for checking attributes in directory entries.
Requirements for data encryption	Set SSL/TLS-related parameters appropriately 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.

Installation or Configuration Resource	EA Usage
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 key length appropriately in certificate validation definitions.

Installation and Startup Checklist

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

Installation Task	Procedure to Complete
Install EA	<i>Install EA on UNIX</i> on page 22
Start the EA server	<i>Start the EA Server on UNIX</i> on page 23
Start the EA GUI	Use one of the following procedures: <ul style="list-style-type: none"> ◆ <i>Start the EA GUI From the Computer Where the EA GUI Is Installed</i> on page 25 ◆ <i>Start the EA GUI from a Remote Computer</i> on page 25
Change the password for the admin user	<i>Change the Admin Password</i> on page 34

Install EA on UNIX

During installation, you define a passphrase. A passphrase is six or more characters long and contains any combination of characters. Be sure to write it down because you may need to provide it when you start the EA server.

To install EA on UNIX:

1. Navigate to the directory where the installation file is downloaded or mount the drive containing the distribution media and navigate to the root directory to locate the installation file.

Note: To install EA from CD on an HP platform, mount it using RockRidge Interchange Format.

2. From the directory for your platform, type the following command. The directory for each platform is identified in the table that follows the command.

```
sh SEASInstall.bin
```

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

3. Select the locale using the corresponding numeric value and press **Enter**.
4. Accept the default installation directory or specify a different directory and press **Enter**.
5. Accept the default value for the port for the nonsecure listener or specify a different port and press **Enter**. The default port is 61365.
6. Type a passphrase that is 6 or more characters and press **Enter**. Write it down because you may need it to start the server.
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 details and press **Enter**. When the installation is complete, the command prompt is displayed.

Start the EA Server on UNIX

When you install the EA server, you define a passphrase. It is required to start the server. You can use the default installation and start the EA server with the encrypted passphrase stored on the server or you can modify the startup to require that the user type a passphrase at startup.

Storing the passphrase eliminates the need to supply it at startup. Determine which method to use and complete the procedure for the method you select.

Start the EA Server on UNIX Using a Stored and Encrypted Passphrase

To start the EA server using a stored passphrase:

1. Navigate to *install_dir/bin*, where *install_dir* is the directory where EA is installed and type the following command:

```
./runSeas.sh
```

2. Confirm that the following messages are displayed when the server starts:

```
Sterling External Authentication Server Starting...  
Waiting for bootstrap data...  
Sterling External Authentication Server is ready for Service
```

Start the EA Server on UNIX and Require a Passphrase

To start EA and require that the 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. Navigate to the *install_dir/bin* directory and type the following command, where *passphrase* is the passphrase specified during installation:

```
./runSEAS.sh passphrase
```

3. Confirm that the following messages are displayed when the server starts:

Start the EA GUI On UNIX

When you start the GUI and connect to the server for the first time, you must use the nonsecure port. To connect to the server using the secure listener port, you set up the certificates on the server and on the GUI and enable the secure listener port in the server. Refer to Chapter 5, *Create and Manage System Certificates*.

You can 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, where *install_dir* is the EA installation directory and type the following command:

```
./runGUI.sh
```

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

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. Refer to *Change the Admin Password* on page 34 for more information. You can also create additional user definitions.

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 computer running the server, and *port* is the port number for the servlet container (as specified during installation). The default port is 9080.
3. Click **Launch GUI**. The first time you run EA from a browser, Java Web Start dialog boxes inform you about the progress of 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

6. Click **Login**.

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. Refer to *Change the Admin Password* on page 34 for more information. You can also create additional user definitions.

Login Fields

Following is a description of the login fields:

Parameter	Description
Host	The host name or IP address of EA.
Port	The port number where EA is listening for connections. The default port is 61365.
User	The name of the user who is logging on. The default user is admin.
Password	The password assigned to the user. The default password is admin. After you log on for the first time, change this password.
SSL/TLS	This option is available only after you set up certificates on the GUI and the server, and configure the secure listener. Click the SSL/TLS check box to enable SSL/TLS for data channel encryption between the server and GUI. This check box is enabled when certificate files are imported. Refer to <i>Create and Manage System Certificates</i> on page 41.
Config	After setting up the appropriate certificate files, click Configure to specify the path for key store and truststore files. For more information, see <i>Configure SSL or TLS Between the GUI and the EA Server</i> on page 55.

Log Off

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 open when client applications need to connect. If you want to shut down the server, close all open GUI windows and then complete the following procedure:

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

Install and Start EA on Windows

Use the following procedures to verify installation resources, install EA, and start the server and the GUI.

Review Resources

EA provides certificate validation and user authentication over a network, in security environments that are configured based on organizational needs and policies. Before you install EA, review any network- and security-specific configuration details that are relevant for the server and GUI components of EA. You may need to consider details that are environment-specific.

Refer to the following list of resources as you plan the use of network and security related resources for installing and configuring EA:

Installation or Configuration Resource	EA Usage
TCP Ports	Use available port numbers, in appropriate port ranges, to set the secure listener, non-secure listener, and the servlet container port used to download the GUI.
Network Interface Addresses	Confirm the local bind address of a network interface you want to use for a connection.
LDAP Directory Information Tree	Apply related knowledge when selecting and specifying LDAP attribute query and assertion parameters for checking attributes in directory entries.
Requirements for data encryption	Set SSL/TLS-related parameters appropriately 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.

Installation or Configuration Resource	EA Usage
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 an organization's selected CAs.
Length of public keys	Set the Public key minimum key length appropriately in certificate validation definitions.

Installation and EA Startup Checklist

Use the following checklist to ensure that you complete al of the tasks to install and start EA:

Installation Task	Procedure to Complete
Install EA	<i>Install EA on Windows</i> on page 28
Start the EA server	<i>Start the EA Server on Windows</i> on page 29
Start the EA GUI	Use one of the following procedures: <ul style="list-style-type: none"> ◆ <i>Start the GUI from the Local Windows Computer</i> on page 30 ◆ <i>Start the GUI from a Remote Computer</i> on page 31
Change the password for the admin user	<i>Change the Admin Password</i> on page 34

Install EA on Windows

At installation, you define a passphrase for EA. A passphrase is six or more characters long and contains any combination of characters.

To install EA on Windows:

1. Navigate to the directory where the installation file is downloaded or mount the drive that contains the distribution media, and navigate to the Windows folder of the media to locate the installation file.
2. Double-click the SEASInstall.exe file.
3. Accept the default language (English) or select a language, and click **OK**.
4. Read the introductory information and click **Next**.
5. Accept the default installation directory or click **Choose** to navigate to the directory to use and click **Next**.

6. Accept the default value for the nonsecure listener port or specify a different port and click **Next**. The default port is 61365.
7. Type a passphrase that is 6 or more characters and click **Next**.
8. To configure the servlet container:
 - a. Accept the default value for the port number of the servlet container or specify a value.
 - b. Accept the default value or specify a value for the fully-qualified DNS name for the engine.
 - c. Click **Next**.
9. Review the installation details and click **Install**.
10. Click **Done**. Review the SEASInstall.log in the installation directory to review installation details.

Start the EA Server on Windows

When you install the EA server, you define a passphrase. The passphrase is required to start the server. You can use the default configuration and start the EA server with the encrypted passphrase that is stored on the server or you can require that a passphrase be typed at startup.

Storing the passphrase eliminates the need to supply it at startup. Determine which method to use and complete the procedure for the method you select.

Start the EA Server on Windows Using a Stored Passphrase

Two methods are available to start the EA server using a stored passphrase: from the Windows Start menu or from Window Services.

To start the EA server using a stored passphrase, from the Start menu:

Click **Start > Programs > Sterling External Authentication Server V2.2.00 > Run Sterling EA Server**.

The following messages are displayed when the server starts:

```
Sterling External Authentication Server Starting...
Waiting for bootstrap data...
Sterling External Authentication Server is ready for Service
```

To run EA as a Windows service:

1. Start **Administrative Tools** from **Control Panel**.
2. Double-click **Services**.
3. Double-click the **Sterling External Authentication Server V2.2.00** entry.
4. To configure the service to start automatically, 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 used to install EA.
2. From a Windows command prompt, navigate to the `install_dir/bin` directory and type the following command, where `passphrase` is the passphrase defined during installation:

```
runSeas.bat
```

3. Confirm that the following messages are displayed when the server starts:

```
Sterling External Authentication Server Starting...  
Waiting for bootstrap data...  
Sterling External Authentication Server is ready for Service
```

Start the EA GUI on Windows

When you start the GUI and connect to the server for the first time, you must use the nonsecure port. To prepare for connection using the secure listener port, you must first set up the certificates on the server and on the GUI and enable the secure listener port in the server. See Chapter 5, *Create and Manage System Certificates*. 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 V2.2.00 > Run Sterling EA GUI**.
2. Provide the following information:
 - ◆ Host
 - ◆ Port
 - ◆ User
 - ◆ Password
 - ◆ SSL/TLS
3. Click **Login**.

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. Refer to *Change the Admin Password* on page 34 for more information. You can also create additional user definitions.

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 host name of the computer running the server, and *port* is the port number for the servlet container (as specified during installation). The default port is 9080.
3. Click **Launch GUI**. The first time you run EA from a browser, Java Web Start dialog boxes inform you about the progress of 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
6. Click **Login**.

Login Fields

Following is a description of the login fields:

Parameter	Description
Host	The host name or IP address of EA.
Port	The port number where EA is listening for connections. The default port is 61365.
User	The name of the user who is logging on. The default user is admin.
Password	The password assigned to the user. The default password is admin. After you log on for the first time, change this password.
SSL/TLS	This option is available only after you set up certificates on the GUI and the server, and configure the secure listener. Click the SSL/TLS check box to enable SSL/TLS for data channel encryption between the server and GUI. This check box is enabled when certificate files are imported. Refer to <i>Create and Manage System Certificates</i> on page 41.
Config	After setting up the appropriate certificate files, click Configure to specify the path for key store and truststore files. For more information, see <i>Configure SSL or TLS Between the GUI and the EA Server</i> on page 55.

Log Off

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

Shut Down EA on Windows

If you close the EA GUI, EA continues to run. Keep the EA server open when client applications need to connect. To shut down EA:

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

Configure System Resources

After you install EA, configure the system for operation in your environment.

Modify the Non-Secure Listener Port

The non-secure listener defines how a client application connects to EA without requiring a certificate for an SSL or TLS handshake. The non-secure listener port is configured initially 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 following parameters:
 - ◆ IP Address
 - ◆ Port
 - ◆ Enabled
4. Click **OK**.

Disable the Non-Secure Listener Port

The non-secure listener defines how a client application connects to EA without requiring an SSL or TLS handshake. You must connect on the non-secure listener port the first time you login. 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 Non-Secure Listener. Deselect the Enabled box.
3. Click **OK**.

Change the Port Number of the Servlet Container

You may require the use of a different port number for the servlet container specified at installation. To change the port number, edit two XML files. XML files are in Unicode format with UTF-8 encoding (backwards compatible with ASCII). When you use a line editor such as Windows Notepad or UNIX vi to update these XML files, save them in the appropriate format.


To change the port number specified for the servlet container:

1. Open the XML document file, *install_dir/conf/jetty/JettyConfigDef.xml*, where *install_dir* is the installation directory.
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">`, where *servlet_port* is the servlet port you specified during installation, and *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, you should change the admin password.

To change the password for user admin:

1. From the **Manage** menu, select **Users**. The External Authentication User Definitions displays a list of user definitions.
2. Select the user definition for admin and click .
3. On the Update User dialog box, type the new password in the **Password** and **Confirm Password** fields.
4. Click **OK**.

Configure Logging Options

EA supports multiple levels of logging to help you 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 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 other logging details by editing the `log4j.properties` file in the `install_dir/conf` directory. Edit the `log4j.properties` file to change logging for the server and 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 38 for information.

To change the logging level written to a file, refer to *Change the Logging Level from the GUI* on page 35.

Use the following procedures to configure EA logging from the command line:

Turn Logging to the Console On or Off on page 36

Change the Maximum Log File Size on page 36

Change the Maximum Number of Archive Log Files on page 37

Change the Logging Level in a Logging Properties File on page 37

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. Following are the logging levels:

Level	Description
INFO	Errors, warnings, and informational messages are logged. This is the default.
WARN	Errors and warnings are logged.
DEBUG	Includes INFO and additional information useful for debugging.

Level	Description
ERROR	Only errors are logged.
TRACE	Details will be captured according to Connect:Direct Trace operation (logging in conjunction with the Trace command).
ALL	Logs all available information.
OFF	Turns off logging so that no server performance information is captured.

4. 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 the directory where EA is installed:
2. Open the *log4j.properties* file to change logging for the server or open the *guiolog4j.properties* file to change logging for the GUI.
3. Identify the logging standard output parameters, as shown in the following example from a *log4j.properties* file:

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

The line ending in *ConsoleAppender* turns on the console output. The line ending in *varia.NullAppender* suppresses console output.

4. Comment out the console logging option that you do not want to use by adding the pound symbol (#) at the beginning of the line. By default, the logging 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 Maximum Log File Size

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

1. Navigate to the *install_dir/conf* directory, where *install_dir* is the directory where EA is installed.
2. Open the *log4j.properties* file to change logging for the server or open the *guiolog4j.properties* file to change logging for the GUI.

3. Define, in kilobytes, the maximum size you want the log file to reach before archiving in the `MaxFileSize` parameter. The default is 1000 KB.

```
log4j.appender.R.MaxFileSize=1000KB
```

4. Save the logging properties file.

Change the Maximum 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 the directory where EA is installed:
2. Open the `log4j.properties` file to change logging for the server or the `gui-log4j.properties` file to change logging for the GUI.
3. Type the number of archive log files you want to keep in the `MaxBackupIndex` parameter. The default is 20.

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

4. Save the logging properties file.

Change the Logging Level in a Logging Properties File

You can configure the logging level to determine what server-related performance details are written in a log.

To change the logging level in EA log files:

1. Navigate to the `install_dir/conf` directory, where `install_dir` is the directory where EA is installed.
2. Open the `log4j.properties` file to change logging for the server or the `gui-log4j.properties` file to change logging for the GUI.
3. Define the logging level to report in the `LEVEL` parameter.

```
log4j.rootLogger=LEVEL,R,stdout
```

Following are the logging levels:

Level	Description
INFO	Errors, warnings, and informational messages are logged. This is the default.
WARN	Errors and warnings are logged.
DEBUG	INFO and additional information useful for debugging are logged.

Level	Description
ERROR	Only errors are logged.
TRACE	Details will be captured according to Connect:Direct Trace operation (logging in conjunction with the Trace command).
ALL	All available information is logged.
OFF	No server performance information is logged.

4. Save the logging properties file.

Refresh GUI Lists from the Server

More than one administrator can access the same EA GUI to configure EA. When more than one administrator changes EA definitions, you may need to update the GUI windows that list certificate revocation list, certificate validation, authentication, user definitions, and role definitions.

To update lists with updated configuration information from the server:

From the **Manage** menu, click **Refresh Lists**. When the progress message dialog box closes, all GUI windows that list EA configuration definitions display definitions added to the server since the last refresh.

Set Listener Connection Settings (Backlog and Timeout)

Leave the listener connection fields blank to accept the default connection settings for EA. To change the listener connection settings, you can specify the 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 listener port communications settings, set the following inbound and outbound connection parameters:
 - ◆ Listen Backlog
 - ◆ Accept Timeout
 - ◆ SSL Handshake Timeout
 - ◆ Connect Timeout
 - ◆ Read Timeout
4. Click **OK**.


Create a System-Wide LDAP or HTTP Connection Definition

Create one or more system-wide connection definitions for connecting as required to perform attribute queries or download certificate revocation lists. When you create a certificate validation definition, certificate revocation definition, or authentication definition in EA, you can select from the system-wide definitions that have been created.

Creating system-wide connection definitions before you create the definitions for certificate validation, certificate revocation lists, and authentication is helpful when the same connection information is required for multiple uses. For example, you save time by simply selecting a system-wide LDAP connection when several attribute query definitions require connection to the same LDAP server. In addition, by creating the system-wide connection definitions first, you ensure that when changes to connection details are required, they can be made in one place (instead of in multiple definitions), and that login credentials for authenticated connections are entered only once.

Specifying system-wide server connections saves time and reduces errors that could occur when parameters are entered manually. You can create an LDAP connection definition or an HTTP connection definition.

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 and name the connection definition.
 3. In the **Protocol** field, select the protocol as follows:
 - ◆ For an LDAP connection definition:
 - Specify `ldap://` to connect using the Lightweight Directory Access Protocol.
 - Specify `ldaps://` to connect using the Lightweight Directory Access Protocol over SSL/TLS.
 - ◆ For an HTTP connection definition:
 - Specify `http://` to connect using the HTTP protocol without using SSL/TLS.
 - Specify `https://` to connect using the HTTP protocol using SSL/TLS. When the protocol is `https://` you can specify a client key certificate alias to select a certificate from the system SSL key store for use with the SSL/TLS protocol.
 - Continue with step 5.
- Depending on the protocol you select, either the LDAP Connection Definition screen or the HTTP Connection Definition screen is displayed.
4. For an LDAP connection, specify the following parameters:
 - ◆ Name
 - ◆ Description
 - ◆ Host
 - ◆ Port
 - ◆ Authentication Method

- ◆ Principal Name
 - ◆ Principal Password
 - ◆ Client Key Certificate Alias
 - ◆ LDAP Version
 - ◆ Start TLS
 - ◆ Referral Action
 - ◆ Advanced options
5. For an HTTP connection, specify the following parameters and click **Next**.
- ◆ Name
 - ◆ Description
 - ◆ Host
 - ◆ Port
 - ◆ Client Key Certificate Alias
 - ◆ Advanced options
6. Click **Next** and click **Save**.

Create and Manage System Certificates

Before you can configure SSL or TLS secure connections, you must create, exchange, and store certificates for EA and the entities with which you communicate. Depending on your security policy, how you deploy EA, and the client applications that you communicate with, you may have to distribute your public key to entities and store certificates from client applications, 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 the secure listener port, configure access to the keystore and trust store, and configure a secure connection between a remote GUI and server.

Procedures to Generate and Use Certificates

Depending upon if you are using self-signed certificates or CA-issued certificates determines the procedures you complete to generate and use certificates. In addition, the connections you are securing determine what procedures to complete. Identify the type of certificate you are generating and the connection you are securing, and complete the procedures in one of the following tables.

Procedures to Generate a Self-Signed Certificate to Secure the Connection to the EA Server

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

Task	Procedure
Generate a self-signed-certificate for the EA server to use a self-signed certificate to authenticate EA to a client application.	<i>Generate a Self-Signed Certificate for the EA Server</i> on page 44.
When you generate the self-signed certificate, it is stored in the keystore. You must export it and send it to the entity with which you are communicating.	<i>Export a Self-Signed Certificate for the EA Server</i> on page 48.

Task	Procedure
The EA trust store must contain a copy of a root certificate for each secure server that EA connects to. Obtain a copy of the root certificate for each secure server and complete this procedure.	<i>Import a Certificate into the EA Server Trust Store on page 52.</i>
After you obtain a CA certificate and import it into the keystore, complete this procedure to allow EA to access the SSL keystore.	<i>Configure EA to Access the SSL Keystore on page 53.</i>
After you import client and secure server certificates to the server trust store, complete this procedure to enable the EA trust store.	<i>Configure EA to Access the SSL Trust Store on page 54.</i>
After you import client and secure certificates, complete this procedure to configure the secure listener.	<i>Configure the Secure Connection Listener on page 55.</i>

Procedures to Generate a CA-Issued Certificate to Secure the Connection to the EA Server

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

Task	Procedure
To use a CA certificate to authenticate EA to a client application, first generate the self-signed-certificate for the server. This procedure generates the information needed to create a Certificate Signing Request (CSR).	<i>Generate a Self-Signed Certificate for the EA Server on page 44.</i>
After you create a self-signed certificate, you are ready to create and send a certificate signing request.	<i>Create a PKCS#10 Certificate Signing Request for the Server on page 49.</i>
Obtain a copy of the root certificate from the CA. Distribute this information to the servers that require client authentication, the computer running the GUI, and client applications.	
When you receive the certificate from the CA, import the certificate into the EA keystore. This replaces the self-signed certificate.	<i>Import the CA-Issued Certificate into the Server Keystore on page 51.</i>
The EA trust store must contain a copy of a root certificate for each secure server that EA connects to. Obtain a copy of the root certificate for each secure server and complete this procedure.	<i>Import a Certificate into the EA Server Trust Store on page 52.</i>
After you obtain a CA certificate and import it into the keystore, complete this procedure to allow EA to access the SSL keystore.	<i>Configure EA to Access the SSL Keystore on page 53.</i>
After you import client and secure server certificates to the server trust store, complete this procedure to enable the EA trust store.	<i>Configure EA to Access the SSL Trust Store on page 54.</i>
After you import client and secure certificates, complete this procedure to configure the secure listener.	<i>Configure the Secure Connection Listener on page 55.</i>

Procedures to Generate a Self-Signed Certificate to Secure the Connection between the GUI and the EA Server

Complete the following procedures to configure self-signed certificates between the EA server and the GUI:

Task	Procedure
To use a self-signed certificate to authenticate the EA server to the EA GUI, generate a self-signed-certificate for the server.	<i>Generate a Self-Signed Certificate for the EA Server on page 44.</i>
To use a self-signed certificate to authenticate the connection between EA and the GUI, create a key certificate on the computer where the GUI is running.	<i>Generate a Self-Signed Certificate for the GUI on page 46.</i>
Export a copy of the self-signed certificate you created at the GUI.	<i>Export a Self-Signed Certificate for the GUI on page 48.</i>
At the EA server, import the certificate from the GUI into the EA server trust store.	<i>Import a Certificate into the EA Server Trust Store on page 52.</i>
The GUI trust store must contain a copy of the public keys from the EA server.	<i>Import the Server Certificate into the GUI Trust Store on page 53</i>

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

Complete the following procedures to configure CA-issued certificates to secure a connection between the EA server and the GUI:

Task	Procedure
To use a self-signed certificate to authenticate the EA server to the EA GUI, generate the self-signed-certificate for the server.	<i>Generate a Self-Signed Certificate for the EA Server on page 44.</i>
After you create a self-signed certificate, you create a certificate signing request for the server.	<i>Create a PKCS#10 Certificate Signing Request for the Server on page 49.</i>
When you receive the certificate from the CA, import the certificate into the EA keystore. This replaces the self-signed certificate.	<i>Import the CA-Issued Certificate into the Server Keystore on page 51.</i>
To use a CA certificate to authenticate the connection between EA and the GUI, generate a self-signed-certificate for the GUI.	<i>Generate a Self-Signed Certificate for the GUI on page 46.</i>
Create a CSR that contains information from the key and certificate at the GUI and send it to the CA.	<i>Create a PKCS#10 Certificate Signing Request for the GUI to Submit to a CA on page 49.</i>
Import the certificate you received from the CA into the GUI keystore.	<i>Import the CA-Issued Certificate to the GUI Keystore on page 51.</i>

Task	Procedure
The trust store at the server must contain a copy of the public key from the EA GUI.	<i>Import a Certificate into the EA Server Trust Store</i> on page 52.
Import a copy of the server certificate to the GUI trust store.	<i>Import the Server Certificate into the GUI Trust Store</i> on page 53.

Generate a Self-Signed Certificate

To configure the EA server for secure communications, you must first generate a self-signed certificate. To configure a secure connection between the GUI and the EA server, you must create a key certificate on the computer where the GUI is running.

A self-signed certificate is required whether you use a self-signed or CA-issued certificate to secure the connection between the EA server and clients or between the GUI and the EA server.

If you plan to use a self-signed certificate to authenticate EA to a client application, first generate the self-signed-certificate for the server. Then, export the certificate information and send it to the client application. If you use self-signed certificates, you are responsible for updating and maintaining them.

If you plan to use a CA-issued certificate to authenticate EA to a client application, first generate the self-signed certificate. Then, use this information to generate a certificate signing request (CSR) for a CA-issued digital certificate. After you obtain a CA certificate, import this information into the EA server keystore.

If you plan 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 information and send it to the EA server.

If you plan to use a CA-issued certificate to authenticate the GUI to the EA server, first generate the self-signed certificate at the GUI. Then, use the information from the self-signed certificate to generate a certificate signing request (CSR) for a CA-issued digital certificate. After you obtain a CA certificate, import this information into the GUI keystore.

Generate a Self-Signed Certificate for the EA Server

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 47 for a description of 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 do not use the `-dname` option to define the CN attribute, provide the following information:

- ◆ First and last name

Note: Information you provided in the **First and last name** field is used to create 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:
 - ◆ If you are using CA-issued certificates, continue to *Create a PKCS#10 Certificate Signing Request for the GUI to Submit to a CA* on page 49.
 - ◆ If you are using a self-signed certificate, complete the procedure, *Export a Self-Signed Certificate for the EA Server* on page 48.

Generate a Self-Signed Certificate for the GUI

To establish secure communications between the GUI and the EA server, you must create a key certificate on the computer where the GUI is running.

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

1. On the computer where the GUI is running, 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 creating a key certificate using the `-dname` option to control the attributes used 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 the **First and last name** field is used to create 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 password and the keystore password must be the same for EA to function properly.

5. Do one of the following:
- ◆ If you are using CA-signed certificates, complete the procedure *Create a PKCS#10 Certificate Signing Request for the GUI to Submit to a CA* on page 49.
 - ◆ If you are using a self-signed certificate, export a copy of the file. Refer to *Export a Self-Signed Certificate for the GUI* on page 48.

Parameters to Generate a Self-Signed Certificate

Following is a description 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 other options to view the help provided with the utility.
-genkey	Instructs the Keytool utility to generate a certificate and a private key.
-alias <i>alias_name</i>	Name of the certificate. This name is used to identify the certificate in the keystore.
-keyalg <i>alg_type</i>	Type of algorithm used to create the key. This value must be an RSA algorithm.
-keystore <i>keystore_path</i>	Path and file name of the keystore file. If you omit this parameter for the command, the keystore is created in your home directory with the file name .keystore .
-keysize <i>keysize</i>	Size of the key to create. Maximum key size is 2048.
-validity <i>validity_in_days</i>	Number of days that the certificate is valid for.
-storepass <i>password</i>	Password of the keystore file.
-dname	Controls attributes used to specify the distinguished name in your self-signed certificate or CSR. For example, you can 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 may need to send this information to the client with which you are communicating. Complete the procedure, *Export a Self-Signed Certificate for the EA Server* on page 48, to export the information.

After you create a self-signed certificate at the GUI, you need to export the certificate and send it to the EA server. Complete the procedure *Export a Self-Signed Certificate for the GUI* on page 48.

Export a Self-Signed Certificate for the EA Server

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 path. 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 48 for a description of the parameters.

2. Import the certificate into the EA trust store. Refer to *Import a Certificate into the EA Server Trust Store* on page 52.

Export a Self-Signed Certificate for the GUI

If you are using self-signed certificates, export a copy of the certificate to a file to import into the server's trust store.

To export the certificate:

1. On the computer where the GUI is running, type the following command, and 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 48 for a description of the parameters.

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

Commands to Export a Self-Signed Certificate

Following is a description of the parameters used to export a self-signed certificate at a server or a GUI:

Parameter	Description
keytool	Invokes the Keytool utility.
-export	Exports a copy of the certificate so you can distribute it to the servers with which you communicate as client, to the computer where the GUI resides, and to client applications.
-alias <i>alias_name</i>	Name of the certificate. It is used to identify the certificate in the keystore.

Parameter	Description
-keystore <i>keystore_path</i>	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 <i>password</i>	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 <i>cert_file_name.xxx</i>	Path and file name of the certificate to export.

Create Certificate Signing Request

After you create a self-signed certificate for the server or the GUI, you create a certificate signing request (CSR) that contains information from the key and the certificate. You submit the CSR to a Certificate Authority (CA) and request a digital certificate that is authenticated and digitally signed by the CA. This procedure does not apply to self-signed certificates.

Create a PKCS#10 Certificate Signing Request for the Server

To create a CSR for the certificate created for the EA server:

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

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

Refer to *Parameters to Create a CSR* on page 50 for a description of the parameters.

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

```
SEASkeycert install_dir
```

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-Issued Certificate into the Server Keystore* on page 51.

Create a PKCS#10 Certificate Signing Request for the GUI to Submit to a CA

Complete the following procedure to create a certificate signing request (CSR) that contains key and certificate information from a self-signed certificate. After you create the CSR, submit it to a CA to request a CA-issued digital certificate that is authenticated and digitally signed by the CA.

To create a CSR for the GUI to submit to a CA:

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

```

password          keystore_path      alias_name      CSR_file

```

Refer to *Parameters to Create a CSR* on page 50 for a description of the parameters.

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.

When you receive the certificate from the CA, perform the procedure *Import the CA-Issued Certificate to the GUI Keystore* on page 51.

Parameters to Create a CSR

Following are the parameters used to create a CSR:

Parameter	Description
keytool	Invokes the Keytool utility.
-certreq	Generates a certificate signing request (CSR).
-keystore <i>keystore_path</i>	The path to the keystore that contains the certificate to create the CSR for.
-alias <i>alias_name</i>	The alias name of the certificate to create the CSR for.
-file <i>CSR_file</i>	The path and file name of the CSR to create.
-storepass <i>password</i>	The password of the keystore.

Import the CA-Issued Certificate Keystore

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

Import the CA-Issued Certificate into the Server Keystore

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

1. Navigate to the `install_dir/jre/bin` directory on the EA server.
2. Type the following command and press **Enter**.

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

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

Following is a sample command to import a CA-issued 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. You distribute it to the servers that require client authentication, the remote computer running the EA GUI, and client applications.

Import the CA-Issued Certificate to the GUI Keystore

The following procedure replaces the self-signed certificate that was created in *Generate a Self-Signed Certificate for the GUI* on page 46 with the CA-issued certificate for the GUI.

To import the CA-issued certificate into the GUI keystore:

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

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

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

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

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

Parameters to Import the CA-Issued 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.
-keystore <i>keystore_path</i>	Path and file name of the keystore file.
-alias <i>alias_name</i>	Alias name to identify the certificate in the keystore. Use the same alias as you used to create the certificate in <i>Generate a Self-Signed Certificate for the GUI</i> on page 46.
-storepass <i>password</i>	Password of the keystore file.
-file <i>certificate</i>	Location of the CA-issued certificate to import.

Import Certificates into the Trust Store

Depending upon the connection you are securing, you import certificates into the trust store. To secure the connection between the EA server and client connections, the EA trust store must contain a copy of the root certificate for each secure server that EA connects to as well as from clients that initiate a connection to the EA server. To secure the connection between the GUI and the EA server, the GUI trust store must contain a copy of the public key of EA.

Import a Certificate into the EA Server Trust Store

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

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

```
store_path - password - certificate
```

The following example 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 computer where the GUI resides, type the following command and press **Enter**:

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

The following example 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 you are prompted with the message, Trust this certificate?, type **yes** and press **Enter**.

Parameters 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 <i>truststore_path</i>	Specifies the path and file name of the trust store file.
-storepass <i>password</i>	Password of the trust store file. Default=changeit.
-file <i>certificate</i>	Location of the public certificate to import.

Configure EA to Access the SSL Keystore

This procedure assumes that you have a self-signed or CA-issued certificate in the server keystore. See *Generate a Self-Signed Certificate for the EA Server* on page 44 or *Import the CA-Issued Certificate Keystore* on page 50 for more information.

The SSL keystore file stores the certificate used to connect to secure LDAP servers and to perform TLS/SSL negotiations with connecting client applications.

To configure EA to access the keystore:

1. From the **Manage** menu, select **System Settings**.
2. Click the **SSL** tab.

- Specify the following information:

Parameter	Description
Protocol	Protocol to use for all secure connections. System—Uses the default EA protocol. This is the default. SSLv3—Uses SSL version 3.0 (not valid for Linux). TLS—Uses TLS. JRE—Uses the protocol default of the JRE you are using for EA. Using the protocol default for the JRE is not valid for AIX.
Keystore File	Location of the keystore file on the computer that you are using to initiate a connection with the GUI.
Keystore Password	Password of the keystore file.

- 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 client applications, from EA to LDAP servers that it connects to, and to optionally validate signatures on CRLs and certificates.

This procedure assumes that you have imported client and secure server certificates to the server trust store. See *Import a Certificate into the EA Server Trust Store* on page 52 for more information.

To enable the EA trust store:

- From the **Manage** menu, select **System Settings**.
- Click the **Trusted Certificate** tab.
- Specify the following parameters:

Parameter	Description
Trust Store File	The location of the trust store file on the computer that you are using to initiate a connection with the GUI.
Trust Store Password	Password of the trust store file.

- 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:

Parameter	Description
IP Address	If you want a client connection to use an IP address to connect to EA, type the IP address.
Port	Type the number of the port that EA will listen on for connections. The default is 61366.
Keystore alias	Specifies the alias used in the keystore to identify the key certificate file associated with this listener.
Enabled	Enables secure listening on the port or IP address identified.

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 the EA server. Before you complete this procedure, have a certificate at the GUI and the EA server, import the GUI certificate into the EA trust store and import the root certificate of the EA server into the trust store of the GUI.

To configure SSL or TLS between the GUI and EA server connection:

1. From the Login screen, click **Config**.
2. Specify the Keystore File and password. Click **Next**. The fields are described below:

Parameter	Description
Keystore File	Location of the keystore file on the computer that you are using to initiate a connection with the GUI.
Keystore Password	Password of the keystore file.

3. On the Create SSL/TLS Trust Store Info screen, specify the following values and click **Next**.

Parameter	Description
Trust Store File	The location of the trust store file on the computer that you are using to initiate a connection with the GUI.
Trust Store Password	Password of the trust store file.

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 access information required to download published CRLs. Published CRLs validate certificates and determine if a certificate has been revoked. During certificate validation, EA checks any CRLs referenced in the CV definition to determine whether a certificate has been revoked.


EA supports the use of the system-wide connection definitions based on a specific local bind address and validation of certificates that include the CRL distribution points extension. To create CRL definitions that enable CRL access as required for some CV requests, you may need to create or reference system-wide connection definitions and variables that support advanced operation with certain client, server, or CA certificates. For more information about the CRL distribution points extension, see Chapter 15, *X.509 Extensions*. For information on variables, refer to Chapter 14, *CV and Authentication Definition Variables*. For more information about system-wide connection definitions, see Chapter 4, *Configure System Resources*.

Create a 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.

If you plan to use system-wide definitions to connect to the LDAP server, create the definition before you create a CRL definition. Refer to *Create a System-Wide LDAP or HTTP Connection Definition* on page 39.

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 following parameters and click **Next**:

Parameter	Description
CRL Definition Name	The name assigned to the CRL definition. If you copy another definition, change the CRL Definition name.
Description	An optional description of the CRL.
CRL Cache	Specify this option to save the CRL to a local cache and prevent the need to load the CRL for every validation request. Specify the following cache options: <ul style="list-style-type: none"> ◆ Refresh cache on CRL next update—The CRL stored in cache is refreshed on the next update as specified by the CRL publisher. ◆ Refresh cache at interval—The CRL stored in cache is refreshed on the interval you specify in the text box.
Refresh CRL on every check	Specify this option to refresh the CRL on each validation request.
Clock Tolerance	Number of seconds to allow for differences in clocks between the computer where EA is installed and the time specified in the CRL.
Reject expired CRL	Reject an expired CRL for the purpose of satisfying the CRL check required option in a CV definition. Regardless of this setting or the age of the CRL, a revoked certificate found in the CRL will result in failure of the certificate validation request.
Verify Signature	Verify the signature of the CRL. To be verified, the certificate of the CRL issuer must be included in the system trust store or must be one of the certificates in the certificate chain passed in the validation request from the client application.

4. On the Query General screen, select one of the following to identify how to connect to the server where the CRL is published and how to query for the list:

Parameter	Description
Use defined connection	Use a system-wide connection definition. Select the definition from the drop-down list.
Specify query parameters	Define the connection to the server as you specify query parameters.
Specify query as URL	Specify a URL to query for the server where the CRL is stored. Specifying the query as a URL is convenient when you can copy and paste the appropriate URL. Verify that the URL for an LDAP server or an HTTP server includes the appropriate parameters.

5. Do one of the following:


- ◆ On the LDAP Parameters screen, specify the following parameters:

Parameter	Description
Protocol	The protocol to use when connecting to the LDAP server. <ul style="list-style-type: none"> ◆ For a nonsecure connection, select ldap://. ◆ For a secure connection, select ldaps://.
Host	The hostname of the LDAP server.
Port	The port of the LDAP server.
Base DN	The starting point in the directory to begin the search.
Return Attributes	The attributes to return from the LDAP server. In a CRL query, only one attribute should be returned: default certificateRevocationList.
Scope	The scope of the LDAP search. Valid values are: <ul style="list-style-type: none"> ◆ OneLevel—Searches only the immediate descendants of Base DN. ◆ Base—Searches only Base DN. ◆ Subtree—Searches downward in the directory from Base DN.
Match Attributes	If Scope is OneLevel or Subtree , use this parameter to identify the location of the entry by matching the attributes specified to the attributes of the entry in the directory. Click <input type="button" value="..."/> to specify the attributes to match. You can match on any attribute stored on your LDAP server.
Query Timeout	The minutes and seconds (format MM:SS) that can elapse before the LDAP Attribute Query times out and processing ends.

- ◆ On the HTTP Parameters screen, specify the following parameters:

Parameter	Description
Protocol	The protocol to use when connecting to the HTTP server. <ul style="list-style-type: none"> ◆ For a nonsecure connection, select http://. ◆ For a secure connection, select https://.
Host	The hostname of the HTTP server.
Port	The port of the HTTP server.
Path	The path to use to access related resources.
Query	The query parameters.
Query Timeout	The minutes and seconds (format MM:SS) that can elapse before the query times out and processing ends.

6. On the LDAP Connection Settings screen, specify the authentication method used by the LDAP server, if required, and click **Next**:


Parameter	Description
Authentication Method	The authentication method to use when authenticating security principals. Valid values are: <ul style="list-style-type: none"> ◆ None—No authentication is performed. Default. ◆ Simple—Password is authenticated against the password found in the directory. ◆ Digest-MD5—SASL authentication method supported by most LDAP v3 servers. ◆ GSSAPI—SASL authentication method that negotiates Kerberos V authentication. This is the native authentication used in Active Directory. ◆ External—SASL authentication method that relies on the transport layer to perform authentication (TLS or SSL client certificate authentication). When SASL External is the authentication method, the LDAP bind principal is the subject of the certificate selected by the Client Key Certificate Alias.
Principal Name	If authentication is required, enter the name of the security principal requesting the LDAP search. The security principal is typically specified as a Distinguished Name, but may take other forms depending on the authentication method and the directory.
Principal Password	Password of the security principal.
Client Key Certificate Alias	This parameter selects the key/certificate to use from the system keystore during SSL or TLS negotiations with the LDAP server when the LDAP server configuration requires client authentication. This parameter is disabled unless the selected protocol is ldaps://, or the Start TLS option has been set to Yes.
LDAP Version	The version of LDAP you are authenticating against. Valid values are: <ul style="list-style-type: none"> ◆ 2—LDAP version 2 ◆ 3—LDAP version 3 (default)
Start TLS	Specifies whether to request TLS encryption using the LDAP v3 extended operation, Start TLS.
Referral Action	Specifies the action to take when an authentication request is referred by one LDAP server to another LDAP server. <ul style="list-style-type: none"> ◆ Follow—Follow the referral to the referred directory. ◆ Ignore—Ignore the referral. ◆ Throw—Ignore the referral and generate an exception.
JNDI Properties	Click  to specify JNDI property names and values if a JNDI service provider bundled with the JRE requires any special properties.

7. On the Confirm screen, verify the parameters and click **Save** and **Close**.

Edit a CRL Definition

You can edit a CRL definition from within the certificate validation definition it is associated with or from the **Manage** menu. For more information on the CRL fields, refer to *Create a CRL Definition* on page 57.


To edit a CRL:

1. Do one of the following:
 - ◆ To edit the CRL from the Certificate Validation Definitions window, double-click the definition that includes the CRL definition to modify. Click the **Referenced CRLs** tab.
Select the CRL definition to modify and click .
 - ◆ To edit a CRL from the EA menu, select **Manage > CRL Definitions**. Double-click the CRL to edit.
2. On the **General** tab, edit any parameters.
3. On the **Query General** tab, change the connection or define a new connection to the LDAP server.
4. Click the **LDAP Parameters** tab and modify any parameters.
5. Click the **LDAP Connection Settings** tab and modify the connection settings for the LDAP server.
6. Click the **Summary** tab and review all parameters. Click **OK**.

Copy a CRL Definition

You can 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. To open a CRL definition from the Certificate Validation Definitions window, double-click the CV definition that includes the CRL definition. Click the **Referenced CRLs** tab. Highlight the CRL definition to modify and click .
2. To open a CRL definition from the Manage menu, select **Manage > CRL Definitions** and double-click the CRL to copy.
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 from the Manage Menu

You can 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**.

Create and Manage Certificate Validation (CV) Definitions

Certificate Validation (CV) definitions specify how EA validates a certificate when a client application sends a certificate validation request. The request references a profile to specify the appropriate CV definition. The name of the definition that validates certificates must have exactly the same name as the profile referenced in the CV request.


When you create a CV definition, you can create definitions for LDAP attribute queries and attribute assertions, set up access to CRLs, and configure how X.509 supported extensions and X.509 custom extensions are allowed or required during validation of a certificate. 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. See *Specify General Details for a Query* on page 113 for more information.

After EA completes all certificate validation steps, it sends a response message to indicate the success or failure of certificate validation to the client application that sent the request. If validation is successful, EA can use the unique conversation ID associated with certificate validation results to access data from the certificate validation as it processes a related request for user authentication and authorization from the same client application.

Create a CV Definition

Certificate Validation (CV) definitions specify how EA validates a certificate when a client application sends a certificate validation request. The request references a profile to specify the appropriate CV definition. The name of the definition that validates certificates must have exactly the same name as the profile referenced in the CV request.

To create a CV definition:

1. From the Certificate Validation Definitions window, click .

2. On the General screen, specify the following parameters:

Parameter	Description
Name	Used by the client application to reference this definition. Up to 255 characters and includes alphanumeric character and space, underscore (_), and period (.).
Description	Summarize details and help administrators determine when to use this definition.
Clock tolerance	Number of minutes to allow for differences between the clock on EA and the time on the certificate. The default setting is 0 minutes.
Expiration grace period	How many hours after the expiration of a certificate that EA should continue to accept it. The default setting is 0 hours. This option allows you to accept an expired certificate for a specified amount of time after it expires and can prevent a shutdown. Note: If you do not continuously monitor the logs for expired certificates and have no policies dictating shutdown based on certificate expiration, consider using the default setting of 0 hours.
Expiration warning	How many days before the scheduled expiration to warn the user about an upcoming certificate expiration. The default setting is 14 days.
Validate using custom exits	Enables the use of custom validation by exiting to a Java class or a script or program run from an operating system command. If you need to add custom exits to a CV definition, see <i>Configure and Test a Custom Exit for a CV Definition</i> on page 65 for more information. Note: If you implement your custom exit in Java, review the detailed description of the interface and a sample class in standard javadoc at the following location: <i>install_dir/doc</i> where <i>install_dir</i> is the installation directory.
Public key minimum key length	Enforces a minimum key length for the public key of a certificate. Type the minimum key length for the public key. If a certificate has a key length that is shorter than the minimum, validation fails. For example, if you specify 2048 as the minimum key length and encounter a certificate with a public key length of 1024, then the certificate fails validation.

3. Specify how to validate certificates using the following table as a guide:

Parameter	Description
Validate to root	Validates the certificate chain to the root certificate. If the validation request does not include the complete certificate chain, EA searches the trust store for the missing issuer certificates.
Validate to Trust Anchor	Validates all certificates in the request. At least one of these certificates must also be found in the trust store. If not, the issuer of the last certificate in the request must be found in the trust store or the request fails.

Parameter	Description
Validate using custom exits	<p>Enables the use of custom validation by exiting to a Java class or a script or program run from an operating system command. To add custom exits to a certificate validation definition, see <i>Configure and Test a Custom Exit for a CV Definition</i> on page 65.</p> <p>Note: If you implement your custom exit in Java, review the detailed description of the interface and a sample class in standard javadoc at <i>install_dir/doc</i> where <i>install_dir</i> is the installation directory.</p>
Public key minimum key length	Enforces a minimum key length for the public key of a certificate. Type the minimum key length for the public key. If a certificate has a key length that is shorter than the minimum, validation fails. For example, if 2048 is the minimum key length specified, a certificate with a public key length of 1024 fails validation.

4. Perform one of the following actions:

- ◆ To set up how EA verifies a certificate subject in an LDAP attribute query definition, click **Next** and go to *Specify Certificate Subject Verification for an Attribute Query* on page 68.
- ◆ If you selected **Validate using custom exits**, go to *Configure and Test a Custom Exit for a CV Definition* on page 65.
- ◆ To create an LDAP attribute query without specifying how to verify the subject of a certificate in the query, go to Chapter 13, *Create and Manage Attribute Queries and Assertions*.

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.

You must 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 name you specify in the CV definition.

Be sure to perform the prerequisite tasks listed in *Prerequisites for Using a Custom Exit* on page 65 before you begin configuring a custom exit for a 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 to edit 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 class name. Be sure to enter the fully-qualified class name in the format *packageName.className* when you specify the custom exit class that implements SEASCustomExitInterface.
5. Click  next to **Properties**.
6. Type a name and a 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. Click **OK**.
7. Click **OK**.
8. Review the log for indications that the certificate was validated successfully by a 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 to edit 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 how to pass the certificate chain to the operating system command. Select one of the following:
 - ◆ 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 processing by the custom exit.

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}
```

- ◆ 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 program written for the custom exit is logged. All error output is logged in SEAS.log at the level specified for standard error log level. All standard (console) output is logged in the SEAS.log at the level specified for standard output log level.

8. Set the log levels required to meet your reporting needs. Following is a description of the logging levels:


Log Level	Definition
INFO	Logs errors, warnings, and informational messages. This is the default.
WARN	Logs errors and warnings.
DEBUG	INFO and additional information useful for debugging are logged.
ERROR	Only logs errors.
TRACE	Logs details logged according to the Connect:Direct Trace operation (logging in conjunction with the Trace command).
ALL	Logs all available information.
OFF	Turns off logging so that no server performance information is captured.

9. To redirect standard error and standard output from the custom exit to the response message that EA returns to the client, select one or both of the following parameters:
 - ◆ Select **Log output from stderr to response message** to send the standard error log output to the response message.
 - ◆ Select **Log output from stdout to response message** to send the standard log output to the response message.
10. Click **OK**.
11. Review the log to determine if the certificate was validated successfully by the custom exit.


Specify Certificate Subject Verification for an Attribute Query

When you create a CV definition, you can preconfigure parameters used to verify the subject of a certificate. Parameters on the Subject Verification Query screen give you the option of predefining part of an attribute query that locates a directory entry associated with the certificate subject. When you create an attribute query definition, EA uses the certificate attributes specified on the Subject Verification Query screen to automatically fill in related parameters. Create a CV definition before you complete this procedure. Refer to *Create a CV Definition* on page 63 for more information.


To specify verification of a certificate subject:

1. From the Certificate Validation Definitions window, click .
2. Provide a name for the CV definition in the Name field and click **Next**.

3. Define parameters to specify how an LDAP attribute query verifies a certificate subject. Following are the parameters:

Parameter	Description
Define query to verify certificate subject	Indicate that you wish to verify the subject of a certificate using an LDAP attribute query. Define one of the methods to use to verify the subject in the Search directory for certificate subject using these attributes field.
Search directory for certificate subject using these attributes	Search for a certificate that matches the certificate subject value. Define one or more values to match: <ul style="list-style-type: none"> ◆ CN—Common Name ◆ OU—Organization Unit ◆ O—Organization ◆ L—City/Locality ◆ ST—State/Province ◆ C—Country (2-character abbreviation) ◆ UID—User ID
Certificate subject is a valid DN in directory	Verify that the subject of the certificate matches a Distinguished Name in a directory.
Other	Select this option if none of the previous options correspond to the search criteria needed to verify your certificate subjects. Enter the query parameters manually on the Query Parameters screen.
Use defined connection	Specify a connection and select a system-wide server connection you defined previously or click  to create a server connection definition.
Verify certificate matches certificate in directory	Perform a comparison of the certificate to the certificate stored in the directory entry. The text in the Certificate Attribute field lists the name of the directory attribute where the certificate is stored. If the attribute name listed is not correct, change this name to correctly represent the certificate attribute of your directory entry.


4. Perform one of the following actions:

- ◆ To create LDAP attribute queries and assertions for the CV definition, click **Next**. Go to Chapter 13, *Create and Manage Attribute Queries and Assertions*.
- ◆ To skip creating attribute query definitions and attribute assertion definitions, and reference CRLs, click **Next** twice. Go to *Reference a CRL Definition* on page 70.
- ◆ To create a CRL for this CV definition, click  and go to Chapter 6, *Create and Manage Certificate Revocation Lists (CRL) Definitions*, and create the CRL.

Reference a CRL Definition

To create CRL definitions to reference in CV definitions, go to Chapter 6, *Create and Manage Certificate Revocation Lists (CRL) Definitions*.



To reference an existing CRL definition:

1. From the Certificate Validation Definitions window, select the CV definition where you want to reference the CRL and click .
2. Click the Referenced CRLs tab.
3. Select the CRL definition to use for this CV definition by moving it from the list on the left to the list on the right.
4. To configure a supported extension in the CV definition, click **Next**. Refer to *Configure Supported Extension Use for a CV Definition* on page 70 or *Create and Configure a Custom Extension for a CV Definition* on page 71 to configure extensions.
5. To complete the CV definition without configuring standard or custom extensions, click **Next** until the Confirm screen is displayed. Click **Save**. From the Finish screen, click **Close** to list the CV definition in the Certificate Validation Definitions pane of the GUI.

Configure Supported Extension Use for a CV Definition

The KeyUsage, BasicConstraints, and CRLDistributionPoints certificate extensions are supported in EA. These extensions are listed with the corresponding object identifiers (OID) and names on the Supported Extensions screen. For detailed information on supported extensions, see Chapter 15, *X.509 Extensions*.



To configure a supported extension:

1. From the Certificate Validation Definitions window, select the CV definition where you want to configure a supported extension and click .
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. Click **Allowed** or **Required** to indicate how each listed extension can be used in a certificate:
 - ◆ To allow an extension to be used in a certificate, whether you use the extension or not, enable **Allowed**. The validation does not fail if the extension is not in the certificate.
 - ◆ To require a certificate to have a specific extension, enable **Required**. The validation fails if the extension is not in the certificate.
 - ◆ Disable both **Allowed** and **Required** to reject an extension.
6. Click **OK**.

Create and Configure a Custom Extension for a CV Definition

On the Custom Extensions screen, review the numbered list of Custom Extensions. The custom extensions are identified by object identifier (OID) and Name. The key values used in custom extensions are configurable. Use the following procedure to edit the key values. For detailed information on custom extensions, see Chapter 15, *X.509 Extensions*.


To create and configure a custom extension:


1. Open a CV definition.
2. From the Custom Extensions screen, click .
3. Click inside the associated text boxes and type the OID (object identifier) and Name.
4. Click  to display the Properties screen.
5. Specify the **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**. The validation fails only if the value of the extension in the certificate does not match the value set on the properties screen, or if the value set on the properties screen is **false**.
 - ◆ To require a certificate to have a specific extension, enable **Required**. The validation fails if the extension is not present in the certificate, if the value of the extension in the certificate does not match the value set on the properties screen, or if the value set on the properties screen is **false**.
 - ◆ Disable both **Allowed** and **Required** to reject an extension.
7. Click **OK**.

Edit or Copy a CV Definition

Certificate validation requirements can change, and those changes can result in changes in the certificate validation requests from client applications. When changes occur, you can update, copy, and delete CV definitions and the definitions included in them. You can also create new CV definitions by copying, renaming, and updating parameters to save configuration time when creating a definition that is similar to an existing definition.

To edit or copy a CV definition:

1. To edit a CV definition, select the definition to edit and click . On the **General** tab, edit parameters as needed. Refer to *Create a CV Definition* on page 63 for more information on the parameters.

2. To copy a definition, select the certificate validation definition to copy and click . Specify a new name.
3. If necessary, edit a Java class custom exit. Refer to *Configure and Test a Custom Exit for a CV Definition* on page 65 for field descriptions.
4. If necessary, edit a Native OS command custom exit. Refer to *Specify an Operating System Command for a Custom Exit* on page 67.
5. If necessary, edit how standard error and standard output from the custom exit are applied to the response message that EA returns to the client.
6. Perform the following procedures as required to edit the components of the CV definition:
 - ◆ *Edit a CV Attribute Query Definition* on page 72
 - ◆ *Edit a CV Attribute Assertion Definition* on page 72
 - ◆ *Edit Supported Extensions in a CV Definition* on page 73
 - ◆ *Edit Custom Extensions in a CV Definition* on page 73
 - ◆ *Delete a Certificate Validation Definition* on page 74

Edit a CV Attribute Query Definition

You can only edit attribute queries from the certificate validation definition or authentication definition in which they are created. Fields that are dimmed cannot be modified.

To edit an attribute query definition:

1. From the Certificate Validation definitions list, double-click the Certificate Validation definition that contains the attribute query definition to modify.
2. From the Certificate Validation Definition Properties screen, click the **Attribute Query Definitions** tab and double-click the attribute query definition to edit.
3. On the **General** tab, modify the parameters as required.
4. Edit the query parameters as required.
5. Edit the LDAP connection parameters on the **LDAP Connection Settings** tab as necessary.
6. Click the **Summary** tab and review all parameters specified for the LDAP attribute query.
7. If all settings are accurate, click **OK**.

Edit a CV Attribute Assertion Definition

You can only edit an attribute assertion definition from the certificate validation definition or authentication definition in which it was created. Fields that are dimmed cannot be modified.

To edit an attribute assertion:


1. From the Certificate Validation Definitions list, double-click the CV definition that contains the attribute assertion to modify.

2. From the Certificate Validation Definition properties screen, click the **Attribute Assertion Definitions** tab and double-click the Attribute Assertion Definition to edit.
3. Modify the available parameters:
 - ◆ Assertion name
 - ◆ Description
 - ◆ Assertion
4. Click **OK**.
5. Click the **Summary** tab and review all parameters. If the settings are accurate, click **OK**.
6. Click the **Referenced CRLs** tab and change the referenced CRL by selecting it and either moving it to the right to reference it or by moving it to the left to stop referencing it.

Edit Supported Extensions in a CV Definition

Supported extensions can only be edited from the certificate validation definition in which they are used.


To edit supported extensions:

1. From the Certificate Validation Definitions list, double-click the certificate validation that contains the extensions to modify.
2. From the Certificate Validation Definition properties screen, 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.

Edit Custom Extensions in a CV Definition

Custom extensions can only be edited from the certificate validation definition in which they are used.


To edit a custom extension:

1. From the Certificate Validation Definitions list, double-click the certificate validation that contains the custom extension to modify.
2. From the Certificate Validation Definition properties screen, click the **Custom Extensions** tab, select the custom extension, and click .
3. Modify the values in the **Properties** dialog box and click **OK**.
4. Disable or enable the **Allowed** and **Required** options, as required.

Delete a Certificate Validation Definition

Deleting a certificate validation definition deletes all parameters and definitions included 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 certificate validation definition to delete and click .
2. Click **OK**.

Manage CV Attribute Query Definitions

Use the following procedures to copy and delete an attribute query that is part of a CV definition. To edit a CV attribute query, see *Edit a CV Attribute Query Definition* on page 72.



Copy a CV Attribute Query Definition

You can copy an attribute query definition when you need to create a new query definition with similar parameters and definitions. After copying the query definition, rename it and edit the settings to create the new one:

1. From the Certificate Validation Definitions window, select the certificate validation definition that contains the attribute query definition to copy and click .
2. From the Attribute Query Definitions window, select the attribute query definition to copy and click .
3. Specify a name for the attribute query definition and optional description and change the parameter values as required using the wizard.
4. On the Confirm screen, review the changes and click **Save**.

Delete a CV Attribute Query Definition

To delete an attribute query definition that is part of a CV definition:

1. From the Certificate Validation Definitions window, select the certificate validation definition that contains the attribute query definition to delete and click .
2. From the Attribute Query Definitions window, select the query to delete and click .
3. Click **OK** at the confirmation message.



Manage CV Attribute Assertion Definitions

Use the following procedures to copy and delete an attribute assertion that is part of a CV definition. To edit a CV attribute assertion, see *Edit a CV Attribute Assertion Definition* on page 72.

Copy a CV Attribute Assertion Definition



Copy an attribute assertion definition when you need to create a new assertion definition with similar parameters. After copying the assertion definition, rename and edit it to create the new one.

To copy a CV attribute assertion:

1. From the Certificate Validation Definitions window, select the certificate validation definition that contains the attribute assertion definition to copy and click .
2. From the Attribute Assertion Definitions window, select the attribute assertion definition to copy and click .
3. On the Add Assertion Definition screen, specify a name for the attribute assertion and optional description.
4. Define the assertion and click **OK**.

Delete a CV Attribute Assertion Definition

To delete a CV attribute assertion definition:

1. From the Certificate Validation Definitions window, select the certificate validation definition that contains the attribute assertion definition to delete and click .
2. From the Attribute Assertion Definitions window, select the attribute assertion definition to delete and click .
3. Click **OK** at the confirmation message.


Manage Custom Extensions in a Certificate Validation Definition

The following procedures explain how to manage custom extensions. You can only edit supported extensions. For instructions to edit supported and custom extensions, see *Edit Supported Extensions in a CV Definition* on page 73 and *Edit Custom Extensions in a CV Definition* on page 73, respectively.

Add a Custom Extension to a CV Definition


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 Chapter 15, *X.509 Extensions*.

To add a custom extension:

1. From the Certificate Validation Definitions list, double-click the certificate validation to modify.
2. From the Certificate Validation Definition properties screen, click the **Custom Extensions** tab and click .
3. Click inside the associated text box and type the OID (object identifier) and **Name**.
4. Enable or disable the **Allowed** and **Required** options:
 - ◆ To allow an extension to be used in a certificate, enable **Allowed**. The validation fails only if the value of the extension in the certificate does not match the value set on the properties screen, or if the value set on the properties screen is **false**.
 - ◆ To require a certificate to have a specific extension, enable **Required**. The validation fails if the extension is not present in the certificate, if value of the extension in the certificate does not match the value set on the properties screen, or if the value set on the properties screen is **false**.
 - ◆ Disable both **Allow** and **Require** to reject an extension.

Delete a Custom Extension in a CV Definition

To delete a custom extension:


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

Create and Manage LDAP Authentication Definitions


Authentication definitions specify how EA authenticates a security principal when a client application sends a request for authentication. Authentication definitions include parameters for connecting to a server, information needed to determine the authentication principal, and mechanisms used for authentication. Creating authentication definitions allows you to specify parameters that EA uses when accessing directories. The authentication definition can include definitions for any attribute queries, attribute assertions, and application-specific outputs required to perform authentication.

Create an LDAP Authentication Definition

To create an LDAP authentication definition:

1. From the Authentication Definitions window, click  to add an authentication definition.
2. On the LDAP Authentication screen, specify the following parameters and click **Next**.

Parameter	Description
Profile Name	Name for the authentication definition. It is included in the request from a client application. Profile names are up to 255 characters in length and can include alphanumeric character and the space, underscore (_), or period (.).
Description	Description of the authentication definition.
Authentication Type	Select LDAP as the authentication type.
Protocol	Select the protocol to use. For a clear text connection, select the ldap:// protocol. For LDAP over SSL, select ldaps://.
Host	Hostname for the LDAP server.
Port	Port number to connect to the LDAP server.

Parameter	Description
LDAP principal to bind	<p>Security principal to use to bind to the LDAP server. It is typically the DN (Distinguished Name) of the user entry containing the user ID. However, the appropriate entry depends on the LDAP server and authentication mechanism used.</p> <p>Select an option to indicate how to bind to the LDAP server:</p> <ul style="list-style-type: none"> ◆ User ID from request—Bind to the LDAP server using the user ID specified in the authentication request from the client application. Use this option if the client application presents the user ID in a form that can be used directly in the bind request to the LDAP server. Some Simple Authentication and Security Layer (SASL) implementations may accept or require a user ID rather than a DN. ◆ Search for user DN—Search for the directory entry that contains the user ID in the authentication request from the client application. This is usually in the CN or UID attribute of the directory entry, depending on the object class used for user entries by the directory. The DN of this entry is used as the security principal for the bind operation. By default the LDAP Attribute Query Definition Wizard will specify a search filter of CN={userId}, where {userId} is a variable representing the user ID from the authentication request. If the directory entry uses an attribute other than CN to store the user ID, then the appropriate attribute (for example, UID) will need to be specified in place of CN. <p>Click  to start the LDAP Attribute Query Definition Wizard. Refer to <i>Create and Manage Attribute Query Definitions</i> on page 113.</p> <ul style="list-style-type: none"> ◆ Specify User DN—Bind using the DN specified. This option can only be used if the name of the directory entry contains the user ID to be authenticated. Specifically, the attribute value of the RDN of the user entry must match the user ID received in the authentication request; typically, this is the Common Name (CN) attribute. <p>If you select this option, you must specify the DN for the principal in the two fields that follow. The information pre-populated in the fields indicates how to specify the DN. Use the format: cn={userId}, <base DN>, where:</p> <p><i>cn</i> represents the attribute used to name the entry.</p> <p><i>userId</i> represents the user ID from the authentication request.</p> <p><i>base DN</i> is a comma-delimited list of RDNs (Relative Distinguished Names) that represents the parent of the user entry.</p>
LDAP principal to bind	<ul style="list-style-type: none"> ◆ DN from Certificate Validation—Bind using the DN returned from the subject verification query. This option is valid when EA is authenticating in continuation of certificate validation and the certificate validation definition included a subject verification query. ◆ Other principal format—Bind to the LDAP server using the security principal specified by the expression you type in the text field. You can use EA Variables to specify this format.
<p>Note: If you define an authentication method that is a continuation of a certificate validation request, the EA variables set during certificate validation are available for the authentication service. Refer to Chapter 14, <i>CV and Authentication Definition Variables</i> for more information about variables.</p>	

3. On the LDAP Connection Settings screen, specify one or more of the following parameters as required to connect to the LDAP server.

Parameter	Description
Principal Name	Security principal to use in the bind operation to the LDAP server. Note: The principal name is displayed as a formula with appropriate details already filled in based on a previous entry or selection for LDAP authentication or an LDAP attribute query parameters.
Principal Password	Password to use for the security principal. The principal password displays as a series of asterisks. The actual password used is the password from the authentication request.
Authentication Method	Method to use to authenticate the security principal. <ul style="list-style-type: none"> ◆ None—This option is not allowed; do not use it. ◆ Simple—Authenticate the password against the password found in the directory. ◆ Digest-MD5—Modify the password to Digest-MD5 and then authenticate against the Digest-MD5 encrypted password in the directory. This method transmits message digests over the network instead of a clear text password. ◆ CRAM-MD5—Modify the password to CRAM-MD5 and then authenticate against the CRAM-MD5 encrypted password in the directory. This method transmits message digests over the network instead of a clear text password. ◆ GSSAP—Use Kerberos V authentication. This is the native authentication used in Active Directory. ◆ External—This option is not allowed; do not use it.
Client Key Certificate Alias	Selects the key/certificate to use from the system keystore during SSL or TLS negotiations with the LDAP server when the LDAP server configuration requires client authentication. This parameter is disabled unless the selected protocol is ldaps://, or the Start TLS option is set to Yes.
LDAP Version	LDAP protocol version. Select one of the following values from the list: <ul style="list-style-type: none"> ◆ 2—Use LDAP version 2. ◆ 3—Use LDAP version 3.
Start TLS	Specifies whether to request TLS encryption using the LDAP v3 extended operation, Start TLS.
Referral Action	Action to take when an authentication request is referred by one LDAP server to another LDAP server. <ul style="list-style-type: none"> Follow—Follow the referral to the referred directory. Ignore—Ignore the referral. Throw—Ignore the referral and generate an exception.

Parameter	Description
Advanced options	<p>Click <input type="button" value="..."/> to specify advanced connection options:</p> <ul style="list-style-type: none"> ◆ Local bind address—Type the IP address for the network interface to associate with the LDAP connection. ◆ JNDI Properties—Click <input type="button" value="..."/> to specify any JNDI property names and values that the JNDI service provider bundled with the JRE requires for special properties.

4. At the Attribute Query Definitions screen, do one of the following:
 - ◆ To skip defining attribute queries and assertions click **Next** twice.
 - ◆ To create attribute queries and assertions, go to *Create and Manage Attribute Queries and Assertions* on page 113.

Create an Application Outputs Definition for an LDAP Authentication Definition

Create an Application Output definition for an authentication definition when you need to perform an LDAP query and return login credentials for user authentication to the client application. Lookup Login Credentials is an application feature that 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. Sterling Commerce defines a directory schema object for storing login credentials to support this feature. 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 directory where EA is installed.

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 create and populate directory entries as described in the following procedures, the creation of an application output definition from the Application Outputs Definition screen is mostly automated. Alternatively, you can use any arbitrary directory object that can store 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 all details of the attribute query that can store 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 all details of the attribute query that obtains the credentials for the application outputs definition. Then you can use controls on the Application Outputs Definition screen 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.

For other LDAP servers, you must 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 your OpenLDAP installation. Schema files are normally in the following subdirectory: */etc/openldap/schema*.
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 */usr/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 *ibmslapd.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 loginPwD 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 82 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 scenario suggested in the preceding example, 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, click the **Application Feature** drop-down arrow to select the method to use for returning 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, perform the following actions:
 - 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 82, 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 *Create and Manage Attribute Query Definitions* on page 113 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, perform the following actions:
 - a. Construct an attribute query to return the user ID and password from your directory object. See *Create and Manage Attribute Query Definitions* on page 113 for instructions.
 - b. After the Attribute Query wizard closes, you must manually map the return attributes to the respective output names.
 - c. In the left pane, click the Output Name, mappedUid. Selecting the output highlights it.
 - d. 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.
 - e. 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.

Edit or Copy an LDAP Authentication Definition

When the authentication requirements or the authorization requirements for an organization's destination services change, the authentication requests and authorization requests from client applications can change also. 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. You can also copy, edit, and delete the attribute query definitions, attribute assertion definitions, LDAP server connections, and application output definitions that an authentication definitions includes. Also, when you need to create a new authentication definition that requires multiple parameters and definitions that are already configured in an existing authentication definition, you can save time and reduce errors by copying, renaming, and editing a similar authentication definition to create the new one.

To copy and edit an LDAP authentication definition:


1. From the Authentication Definitions window, perform one of the following actions:
 - ◆ To make a copy of an authentication definition, select the definition you want to copy and click .
 - ◆ To edit an authentication definition, double-click the definition you want to edit.
2. Type a unique **Profile Name** if you are copying an authentication definition.
3. For LDAP authentication, update the parameters as required. Refer to *Create an LDAP Authentication Definition* on page 77 for a description of the parameters.

Note: If defining an authentication that is a continuation of a certificate validation request, the EA variables set during certificate validation are available for the authentication service. Refer to *CV and Authentication Definition Variables* on page 121 for more information about variables.

4. To change LDAP connection settings, click the **LDAP Connection Settings** tab. Change the parameters as required.
5. To edit attribute queries and assertions, see Chapter 13, *Create and Manage Attribute Queries and Assertions*.
6. To edit application outputs specified for an authentication definition, perform the following actions:
 - a. If you have changed or added the entry for storing credentials in your directory, change the application feature selection. On the Application Outputs screen, click the **Application Feature** drop-down arrow. Choose the application feature that indicates the method used to add an entry for credentials to the directory:
 - To use the standard method of returning attributes in EA, select **Lookup loginCredentials (Sterling)**.
 - To use a custom method for returning attributes in to a client application, select **Lookup loginCredentials (Custom)**.
 - b. To create a new LDAP attribute query for the credentials to be returned, click **Query**. Refer to Chapter 13, *Create and Manage Attribute Queries and Assertions* for more information about creating a query definition.
 - c. To adjust the mapping of return attributes to outputs, select both the return attribute (right) and the output name (left) and click **Map** or **Unmap** to change the mapping.
7. 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**.

Create and Manage SSH Key Authentication and Mapping Definitions

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

A client application such as SSP sends a request to EA. The request contains a profile name, user ID, and SSH public key. EA uses information in the profile to bind to an LDAP directory, look up the SSH keys assigned to the user, and perform an attribute assertion to match the key provided by the user to the list of keys stored at the LDAP server. EA notifies the client if the key sent by the client matches one of the keys stored at the LDAP server.

The credentials of the principal used to bind to the directory are defined in the SSH key authentication profile configuration. Unlike regular user authentication requests, the `userid` from the SSH key authentication request cannot be used to bind to the directory because the password for the user is not available on the key authentication request. The credentials used to bind to the directory are normally the administrator of the directory and are configured in a global LDAP connection definition.

The query to lookup the SSH keys assigned to a user is defined in the profile according to your directory layout. If you use the `openssh` schema provided with EA, the query returns all `sshPublicKey` attributes for the user. If you use a customized schema, be sure to modify the query to ensure that the query returns the attributes associated with the customized schema.

An assertion definition matches the public key from the request against the keys returned by the SSH public key lookup query. A pre-configured assertion is included with EA. It uses the `openssh` schema to store the public keys. If you do not use this schema, edit the assertion definition to use the appropriate schema.

To use SSH key mapping, you must define another query to return a reference to the mapped key. The existing `MapSSHCredentials` query provided with EA returns the new `routingKeyName` attribute of the `loginCredentials` record, and assigns it to the `mappedRoutingKeyName` application output. The application uses the value of the `mappedRoutingKeyName` output to locate a public/private key pair to use as the mapped key for the user.

Create an SSH Key Authentication Definition


Create an SSH key authentication definition to identify how to authenticate an SSH user connecting to SSP and using EA.

Before you create an SSH key authentication definition, define a global connection setting for the LDAP server. Refer to *Create a System-Wide LDAP or HTTP Connection Definition* on page 39.

The protocol, host, and port fields for connection to the LDAP server are automatically populated when you select a globally defined connection.

Select the assertion definition to use with the definition. It matches the public key from the request against the keys returned by the SSH public key lookup query. A preconfigured assertion called `VerifySSHPublicKey` is provided with EA. It uses the `openssh` schema to store the public keys. You can use this SSH assertion definition or define your own. If you do not use the `openssh` schema, you must edit the assertion definition to reference the schema used.

To create an SSH key authentication definition:

1. From the Authentication Definitions window, click  to add an authentication definition.
2. Select `SSHKEY` as the authentication type. The SSH Key Authentication screen is displayed.
3. Define a profile name in the Profile Name field. It is included in the request for the client application. It can be up to 255 alphanumeric characters and include spaces, underscores (`_`), and periods (`.`). Click **Next**.
4. Identify the following information and click **Next**.
 - ◆ The name is automatically populated with `sshPublicKeyQuery`.
 - ◆ Select **Use globally defined connection** as the connection method.
 - ◆ Select the global connection definition that you defined for the LDAP server.
 - ◆ Select **Specify Query Parameters**.
5. On the Query Parameters screen, define the Base DN information. Click **Next**. Base DN information identifies the starting point in the directory to begin the search.
6. Click **Save** to save the definition.
7. At the Attribute Assertion Definitions screen, do one of the following:
 - ◆ The `VerifySSHPublicKey` assertion is prepopulated. Double-click this assertion to review the definition. Click **OK**.
 - ◆ Click **Next** to go to the Application Output Definition page.

Create an SSH Application Output Definition for an SSH Authentication Definition

Create an SSH Application Output definition when you need to perform an SSH user and key query and return login credentials for user authentication to the client application. 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.

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 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 **Next**.
4. The Query Parameters screen is populated with the appropriate parameters. Edit the Base DN field, the starting point in the directory to begin the search. 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 the mapping of return attributes to outputs has been performed 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 *Create and Manage Attribute Query Definitions* on page 113 for instructions.

After the Attribute Query wizard closes, you must manually map the return attributes to the respective 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 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.

Edit or Copy an SSH Key Authentication Definition

When authentication requirements for destination services change, the authentication requests from client applications can change. Changes in requests require that you make related changes in the authentication definitions.

You can change how EA authenticates SSH keys and users by copying, editing, and deleting SSH key authentication definitions. To create a new SSH key authentication definition that requires multiple parameters and definitions that are already configured in an existing authentication definition, you can save time and reduce errors by copying, renaming, and editing a similar definition to create the new one.


To copy and 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, 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. For SSH key authentication, update the parameters as required. Refer to *Create an SSH Key Authentication Definition* on page 86 for a description of the parameters.
4. To change LDAP connection settings, click the **LDAP Connection Settings** tab. Change the parameters as required.
5. To edit attribute queries and assertions, see Chapter 13, *Create and Manage Attribute Queries and Assertions*.
6. To edit application output specified for an authentication definition, perform the following actions:
 - a. If you change or add the entry for storing credentials in your directory, change the application feature selection. On the Application Output screen, click the **Application Feature** drop-down arrow. Choose the application feature that indicates the method used to add an entry for credentials to the directory:
 - To use the standard method of returning attributes in EA, select **Lookup loginCredentials (sterling)**.
 - To use a custom method for returning attributes in to a client application, select **Lookup loginCredentials (custom)**.

- b. To create a new LDAP attribute query for the credentials to be returned, click **Query**. Refer to Chapter 13, *Create and Manage Attribute Queries and Assertions* for more information about creating a query definition.
 - c. To adjust the mapping of return attributes to outputs, select both the return attribute (right) and the output name (left) and click **Map** or **Unmap** to change the mapping.
7. 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**.

Prepare the Directory to Store Keys, a User ID, and Password for an SSH User

Before you can store SSH keys in a directory to perform user authentication and login credentials mapping, you must update the directory schema.

Use the schema files provided by EA to directly extend your schema for OpenLDAP and IBM Tivoli Directory Server directories, or as a reference when you manually extend other directories. If you use these directory extensions and create and populate directory entries, the creation of an application output definition is automated. 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, you must define all details of the attribute query that can store 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 all details of the attribute query that fetches the credentials for the application output definition. Then, use controls on the Application Output Definition screen to manually map attributes returned by the query to outputs that a client application can access.

If you implemented a custom SSH schema you do not need to 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 your OpenLDAP installation.
2. Copy the `sci.schema` file from `install_dir/schema/` to the schema subdirectory of your OpenLDAP installation.
3. Edit the `slapd.conf` file to add an include statement with the added schema references. The file is located in the `/etc/openldap` directory.

```
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 your Tivoli installation. Schemas are often located in the `/usr/ldap/etc` directory.
2. Copy the file `install_dir/schema/V3.sci` to the schema subdirectory of your Tivoli installation. Schema files are normally located in the `/usr/ldap/etc` directory.
3. Restart the LDAP server.

Create Entries for SSH Public Keys

For each user SSH key, define an `sshPublicKey` attribute and set it to the value of the public SSH key for the user. If a user has multiple SSH 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. Copy the content of a public key to an editor. Remove the BEGIN and END comments from the file and delete all 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 with an LDAP user entry that 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: :
c3NoLXJzYSBBQUFBQjNOemFDMX1jMkVBQUBQk13QUFBSUVBbkRUN09VYWROZmNXdH
pzV0QveFIzWXBYd2VmS3FLbVhaQnRsenlIWVRXTjhoOXZtaHdiY1N1NWVtYWZFeVh1eGJr  eXBHRDFMK
0Y1aStVbUZadE1nSUTyblIwQ1hZazhwYmlxeXBSclJ40XBEQWR5QzRrekZaTEJnQzR
2R3N1bjRHTStTZUN
XTVA0Zy9oazRGNFRvWWx6Y0VENTBnaDgzTXVwcl1dhOWZaRko4PSBxYXRlc3RAcWFzbGVzOAO=
sshPublicKey: :
c3NoLWRzcyBBQUFBQjNOemFDMWtjM01BQUFDQkFLU3gyRGoyRmgyZjY5b0hNU2o2Ufo
va3U2ZUJoZlA1enE5UHhUeHBadExXWj1xNFh6NwtKOVfmdzFuZTVNbDhhOHFBSmN2YmFwQStBRG50U2J
0bhZQVfH5MXdObnB2OTUxRjFaYU1Md0ZieJBLUzkkUGJ1aE5ZOE9JbEdJTEY1Q0JraWc2aFFPMXBu
SFJWR1VMMEx0a31odnIOeG5CYtdqTmtKSm1hQUJpZkzJBQUFBRLFEWDC4R1hVdDjPn052QjN4aTRXdG1Nb
UZ6OEZ3QUFBSUFIT1JuUE5sdC9qa25mTW4zZWt1Q3ZHbEVrZjdEQ1hIR1E4UGdEcmNpNWh0US9NekpjR0
tCb2FXRUVNQNGLzBrVV1CdJZkVWZwZTR2dVM5VmZnRzFDV01vMjV6N1BDM2FvQ1lmK0VGUXFRWtuL1B
FV1M1UUU1N1B6S29ueXBma3ZLdFFkS3VtbVNFsFBCR1owbUVWT21EbJhsdTzBb1Z0L28rMmZXZkxvS1FB
QUFJQXdoMHJXRUSUMXVFZFUxV1hOL2hBdmcRtkV1Vy94SnkvQUpXeTnrMGxLajM4MvdnekdiODRneTFDL
2FMam40bWo4Q29ublhPeHVxZnBiL3Q4Q0c1U2xUV1UwaUUxYWpDR0o2ODNVT20wc2xNeT13S1hYU3BJcW
dnU25zTnJaQjJ6Y0lIS29NTDNITHF4WEF4RXZnMndhaTzReHBGd1d3Q0UwOVM4eHBwBm4zdz09IHfhdGV
zdEBxYXJoYXMyMQo=
userPassword:: e1NIQX1rZC9aM2JRWml2L0Z3W1ROak9iVE9QM2tjt0k9
```

Create Entries for Login Credentials

After you add the schema to your directory, you can create `loginCredentials` entries to define attributes for the user ID, password, and SSH key. The supported directory structure creates separate `loginCredentials` entries in the authenticated user's directory entry: one for each destination service.

Set the `loginId` and `loggingPwd` 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 suggested in the preceding example, 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, password, and routing key name 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
```


Create Generic Authentication Definitions

Authentication definitions specify how EA authenticates a security principal when a client application sends a request for authentication. Generic authentication definitions enable configuration of custom authentication using a custom exit, attribute queries, and attribute assertions.

Create a Generic Authentication Definition

To create a generic authentication definition:

1. From the Authentication Definitions window, click .
2. On the LDAP Authentication screen, specify the following parameters and click **Next**:

Parameter	Description
Profile name	Name for the authentication definition. It is included in the request from a client application, can be up to 255 characters, include alphanumeric characters, and a space, underscore (_), and period (.).
Description	Description of the authentication definition.
Authentication type	Select Generic authentication type. The LDAP Authentication dialog changes to Generic Authentication.
User ID required	The authentication request must include a user ID.
Password required	The authentication request must include a password.
Authenticate using custom exits	Authentication will be performed using a custom exit. Click  to specify the Java class or OS command that will perform the authentication function and any other configuration details associated with the exit.

Note: If you are defining an authentication as a continuation of a certificate validation request, the EA variables set during certificate validation are available for the authentication service. Refer to Chapter 14, *CV and Authentication Definition Variables* for more information about variables.

3. At the Attribute Query Definitions screen, do one of the following:
 - ◆ To skip defining attribute queries and assertions click **Next** twice.
 - ◆ To create attribute queries and assertions, go to *Create and Manage Attribute Query Definitions* on page 113.

Configure and Test a Custom Exit for a Generic Authentication Definition

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:

Before you create a generic authentication definition that includes a custom exit to a Java class, review the files in the `install_dir/doc` and `install_dir/samples` subdirectories, where `install_dir` is the directory where EA is installed.

Create a generic authentication definition that includes a custom exit to a script or program. Define the functionality required by writing the code that runs from the operating system command line.

For Java classes created for a custom exit, 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 processing 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/SampleletAuthenticationExit.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.





You must 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 name you specify in the generic authentication definition.

Be sure to perform the prerequisite tasks listed in *Prerequisites for Using a Custom Exit* on page 94 before you begin configuring a custom exit for a 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. If necessary, 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 your generic authentication definition is used to process 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. If necessary, open 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 using a native 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 one of the following methods to determine how to pass the certificate chain to the operating system command.

Note: If certificates will be processed, a certificate validation request to EA must be performed before you can pass a certificate chain.

- ◆ Enable **Certificate file** to pass the certificate chain as a certificate file. Define the following parameters:
 - a. Type the 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 file after the custom exit is complete, enable **Delete file after exit**.
 - d. Click **Standard input (PEM format)** to pass the certificate chain through the standard input stream.
- ◆ Specify the timing for running the custom exit and for authenticating as configured in the generic authentication definition:
 - a. Select **Run default validator after exit** to continue processing the authentication validation definition after the custom exit.
 - b. Select **Run custom exit synchronously** to enable synchronous use of this custom exit. If you select this option, and if a client application sends an authentication 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.
6. Specify **standard error log level** and **standard output log level** to control how output from the program written for the custom exit is logged. All error output is logged in SEAS.log at the level specified for standard error log level. All standard (console) output is logged in the SEAS.log at the level specified for standard output log level.

Set the log level as required to meet your reporting needs. Refer to the following table for a description of the log levels.

Log Level	Definition
INFO	Errors, warnings, and informational messages are logged. Default.
WARN	Errors and warnings are logged.

Log Level	Definition
DEBUG	Includes INFO and additional information useful for debugging.
ERROR	Only errors are logged.
TRACE	Details will be captured according to Connect:Direct Trace operation (logging in conjunction with the Trace command).
OFF	Turns off logging so that no server performance information is captured.

7. To redirect standard error and standard output from the custom exit to the response message that EA returns to the client, select one or both of the following parameters:
 - ◆ Select **Log output from stderr to response message** to send the standard error log output to the response message.
 - ◆ Select **Log output from stdout to response message** to send the standard log output to the response message.
8. Review the log for indications that the authentication occurred successfully using the custom exit.

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 application-specific outputs defined to map log-in credentials for Sterling Secure Proxy.

When Sterling Secure Proxy (SSP) is configured to use this feature, a user logs in to SSP with a set of credentials. EA authenticates the credentials and then returns a different set of credentials to use to log in to the service in the trusted zone. This feature allows you to protect your internal systems because the user IDs and passwords of internal systems are not provided to external users. The external users are only able to log in through the Sterling Secure Proxy.

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 Chapter 14, *CV and Authentication Definition Variables* for information about variables and their use.

Refer to Chapter 8, *Create and Manage LDAP Authentication Definitions* for information about authenticating users using LDAP. When you are creating 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 you are 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 to look up the credentials, as described in Chapter 8, *Create and Manage LDAP Authentication Definitions*. 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 Sterling Commerce loginCredentials entry as described in Chapter 8, *Create and Manage LDAP Authentication Definitions*, 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.

Perform Gentran Integration System (GIS) User Authentication through an EA Custom Exit

Sterling External Authentication (EA) provides a custom user authentication exit to validate a trading partner user ID and password against the GIS user store. Refer to *Create Generic Authentication Definitions* on page 93 for instructions on creating a user authentication.

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

No changes are required to the EA user interface.

Prepare the Certificates for Authentication in the GIS User Store

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

- Configure the HTTP Server Adapter Certificate
- Export the System Certificate from GIS
- 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 GIS CA Certificate Store

Configure the HTTP Server Adapter Certificate

Decide which system certificate the you will use for the HTTP server adapter. You can use the default certificates provided by GIS, or you can import your own. For security reasons, you should use your own certificates.

Note: PEM key certificates must have a txt extension. If your key certificate file 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 GIS certificate store:

1. On the GIS dashboard, select Trading Partners > Digital Certificates > System.
2. Click **Go!** in the check in section for the type of key you are checking in: either a 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** on the Validate When Used screen.
5. Click **Finish** on the Confirm screen.

Export the System Certificate from GIS

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 GIS dashboard, select Trading Partners > Digital Certificates > System.
2. Do one of the following:
 - ◆ Type the name of the system certificate 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 you want 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. The EA trust store is located, by default, 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 GIS 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.
-alias alias_name	The alias name of the EA client certificate in the keystore. Use the same alias as you used to create the certificate.
-storepass password	The password of the keystore file.

Parameter	Description
-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 GIS CA Certificate Store

After the EA client certificate is exported, it must be imported into the GIS CA certificate store.

To import the certificate into the GIS CA certificate store:

1. On the GIS 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 GIS HTTP Server Adapter for EA Support

To configure a GIS server adapter to support GIS user authentication through an EA custom exit:

1. On the GIS 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 the *Configure the HTTP Server Adapter Certificate* on page 100 for information on how to check in a system certificate for the HTTP adapter to use.

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 GIS CA Certificate Store* on page 102 for information on how to check in the CA certificate for EA. If no certificate is selected, the HTTP adapter will not request a certificate from EA. (Client authentication is disabled.)

8. Press **Next**.
9. Click the + button 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, a 500 error will be 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 GIS 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** on the Properties dialog.
10. Click **Next** to move through the definition pages.
11. Click **Save** on the Confirm page.

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 that are written in the secureproxy log file, EA log file, and GIS log file.

Sterling Secure Proxy Messages

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

Message	Sample
Success Authentication	08 Aug 2009 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 2009 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 2009 13:08:41,986 730209 [Pool Worker - 4] INFO com.sterlingcommerce.component.authentication.impl.CommonAuthenticator - AUTH073I iikon1 successfully authenticated.
Failed Authentication	F08 Aug 2009 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).

GIS Authentication Log Messages

Following are the messages written to the GIS log file:

Message	Sample
Failed Authentication	<p>[2009-08-14 13:02:32.931] DEBUG 000000000000 GLOBAL_SCOPE SecurityManager user:gvega attempting to log in (SSO:false)</p> <p>[2008-08-14 13:02:32.931] DEBUG 000000000000 GLOBAL_SCOPE GISAuthentication user:gvega is identified as a LOCAL GIS User</p> <p>[2008-08-14 13:02:32.931] ALL 000000000000 GLOBAL_SCOPE SecurityManager user:gvega authorization FAILED (SSO:false)</p>
Successful Authentication	<p>[2009-08-14 13:03:35.9] DEBUG 000000000000 GLOBAL_SCOPE SecurityManager user:gvega attempting to log in (SSO:false)</p> <p>[2008-08-14 13:03:35.9] DEBUG 000000000000 GLOBAL_SCOPE GISAuthentication user:gvega is identified as a LOCAL GIS User</p> <p>[2008-08-14 13:03:35.9] ALL 000000000000 GLOBAL_SCOPE SecurityManager user:gvega authorization SUCCEEDED (SSO:false)</p>

Create and Manage Tivoli Access Manager (TAM) Authentication Definitions

Create Tivoli Access Manager (TAM) authentication definitions to specify parameters that EA uses when accessing Tivoli Access Manager resources.

Refer to the following procedures to create a Tivoli Access Manager authentication definition:

Create a Tivoli Access Manager Authentication Definition on page 108

Create an Application Output Definition for TAM on page 110

Authenticating with Tivoli Access Manager

EA provides an authentication service for interfacing with Tivoli Access Manager (TAM). The TAM authentication service provides user ID/password authentication and/or user DN authentication through Tivoli Access Manager. DN authentication allows you to authenticate the subject of a certificate received during certificate validation. The TAM authentication service can also provide application-level authorization for accessing a destination service specified in the authentication request and provide credential lookup for logging in to the destination service.

Prerequisites for Tivoli Access Manager Authentication

Each TAM authentication definition (policy) that you create in EA must be configured to securely communicate with the TAM Authorization server and the TAM Policy server. Before you begin the procedure to create a TAM authentication definition, review *Sterling External Authentication Server Version 2.2.00 Release Notes* to ensure that your system meets the requirements for authenticating with TAM and that you have performed the prerequisite tasks described in the release notes.


Logging Information for Tivoli Access Manager Authentication

Because the child process running the TAM API communicates over standard I/O streams to the parent process (the CV process), it is critical that the logger not be configured to use the console appender for output. By default, both processes share `conf/log4j.properties` for configuring logging output as well as the active log file in the logs directory. You can create a separate `log4j.properties`

file for the child process (the TAM API process) if desired, to allow the parent process to log to the console. The child process looks for its own log4j.properties file in the lib/sterling/retro14 directory. If the child process does not find its own properties file, the parent log4j.properties file is used.

Create a Tivoli Access Manager Authentication Definition

Variables supported with authentication definitions are described in *Variables for Authentication Requests* on page 126. To create a Tivoli Access Manager (TAM) authentication definition:

1. From the Authentication Definitions window, click  to display the LDAP Authentication screen.
2. In the **Authentication type** field, select **TAM** to display the Tivoli Access Manager Authentication screen.

Note: When TAM authentication is used, the first line of the log4j.properties file should remain commented out (as it is by default). The TAM authenticator will not function if console output is enabled.

3. On the Tivoli Access Manager Authentication screen, specify the following parameters and click **Next**.

Parameter	Description
Profile Name	The name for the Tivoli Access Manager authentication definition. Profile names can be up to 255 characters and include any alphanumeric character. You can also include the special characters of space, underscore (_), and period (.).
Description	A description of the authentication definition.
TAM Config File URL	The URL to use to locate the TAM configuration file that contains an SSL key and other configuration data used to establish secure communications with the authorization server you are accessing. For example, the following URL specifies a local configuration file: file:///home/SeasAdmin/tam/config_file.conf
Target JRE location	The path to the Java Runtime Environment where the TAM API has been installed. For example, specify the location /home /SeasAdmin/java/j2sdk1.4.2_12 for a JRE location. Note: TAM requires the use of JRE version 1.4.2.

Parameter	Description
TAM User to Authenticate	<p>Enable one of the following to determine which TAM user to bind for authentication:</p> <ul style="list-style-type: none"> ◆ User ID from request—The user ID passed in the authentication request must be a user in the TAM user registry. ◆ Certificate Subject DN—A follow-up authentication request to a previous certificate validation request. The certificate subject DN from the validation request is the security principal passed to the TAM API for authentication. When this option is selected the password and/or user ID are only required if specified by the corresponding options. Doing so provides a form of multi-factor authentication. The user ID and password from the request must match those associated with the user entry corresponding to the certificate subject DN. ◆ Other—Use to support 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. <p>For example, consider the following variable expression, where VerifyCertSubject is the name of the attribute query executed during certificate validation for locating the subject of the certificate in the directory:</p> <pre>{attr[VerifyCertSubject].dn}</pre> <p>When this option is selected, the password and/or user ID are only required if specified by the corresponding options. Doing so provides a form of multi-factor authentication. The user ID and password from the request must match those associated with the user entry corresponding to the security principal specified.</p>
User ID required	The authentication request must specify a user ID. Enable to require both a certificate and a user ID/password as a form of multi-factor authentication.
Password required	The authentication request must specify a password. Enable to require both a certificate and a user ID/password as a form of multi-factor authentication.
Authorize Access to Destination Service	<p>Select this parameter to authorize the specified user to access the destination service requested. Specify the TAM resource representing the destination service and the requested access permissions to be authorized.</p> <p>The default resource is <code>/SEAS/profileName/{dstsvc}</code>, where <i>profileName</i> is the profile the client application uses to access the TAM authentication definition, and <i>dstsvc</i> is a variable described in <i>Variables for Authentication Requests</i> on page 126. The default resource suggests the setup of a protected object space, SEAS in the TAM secure domain specified in the TAM configuration file for the API. Default resource <code>/SEAS/profileName/{dstsvc}</code> further suggests the setup of a protected object corresponding to the TAM authentication profile and a separate protected resource to correspond to each destination service that the client application can request access to.</p> <p>The <code>/SEAS/profileName/{dstsvc}</code> default resource is a suggested convention; you can specify any valid resource name in the configured secure domain. The default requested permissions are Traverse and view (Tv). Any valid permissions can be specified here, but they should reflect the access anticipated by the client application for this resource.</p>

4. At the Attribute Query Definitions screen, do one of the following:
 - ◆ To skip defining attribute queries or assertions, click **Next** twice and continue with *Create an Application Output Definition for TAM* on page 110.
 - ◆ To create an attribute query or assertion definition, go to Chapter 13, *Create and Manage Attribute Queries and Assertions*.

Create an Application Output Definition for TAM

The application output for TAM authentication is implemented using TAM GSO resource credentials.

To create an application output definition for TAM:

1. On the Application Output screen, specify outputs you want the authentication definition to return to the client application.

Parameter	Description
Return TAM Credentials	<p>When EA requests authentication for a given security principal, the TAM API returns a credential object that applications may need for additional interaction with TAM.</p> <p>For example, if single signon solutions are implemented, a related credential is required for access. Select Return TAM Credentials to return this credential to the client application.</p>
Return Destination Service Login Credentials	<p>Select this option to return the credentials for logging in to a destination service.</p> <ul style="list-style-type: none"> ◆ Look Up GSO Resource—Look up the Global Sign On (GSO) resource from the TAM user registry. Specify the resource in the text field that follows. The default value is the variable, {destination service}. The variable {destination service} resolves to the destination service name passed by the client application in the authentication request. ◆ Return User ID and Password from Request—Return the UserID and Password received in the authentication request. ◆ Return User ID from Request and the Following Password— Return the UserID from the request and the password you specify in the text field that follows. ◆ Other—Return some other credential that you specify by typing the mapped user ID (Uid) and mapped password (Pwd) in the fields that follow. This option supports the use of LDAP lookup of credentials that are identical in functionality to those provided in the application output wizard of the LDAP authenticator. See <i>Create an Application Outputs Definition for an LDAP Authentication Definition</i> on page 80 for more information.


2. Click **Next** to review the parameters and click **Save** when all parameters are set as required.

Edit or Copy a TAM Authentication Definition

When the authentication requirements or the authorization requirements for an organization's destination services change, the authentication requests and authorization requests from client applications can change also. 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. You can also copy, edit, and delete the attribute query definitions, attribute assertion definitions, TAM server connections, and application output definitions that an authentication definition includes. Also, when you need to create a new authentication definition that requires multiple parameters and definitions that are already configured in an existing authentication definition, you can save time and reduce errors by copying, renaming, and editing a similar authentication definition to create the new one.


To copy or edit a TAM authentication definition:

1. To make a copy of a TAM authentication definition, select the TAM authentication definition to copy and click . Type a new **Profile Name**.
2. To edit a TAM authentication definition, double-click the definition to edit.
3. To change TAM authentication, update parameters as required. Refer to *Create a Tivoli Access Manager Authentication Definition* on page 108 for more information.
4. See Chapter 13, *Create and Manage Attribute Queries and Assertions*, to edit attribute queries and attribute assertions for the TAM authentication definition.
5. To edit application outputs for a TAM authentication definition, click the **Application Output** tab and update the parameters as required. Refer to *Create an Application Output Definition for TAM* on page 110.

Delete an Authentication Definition

Delete any authentication definition that is no longer needed.

To delete an authentication definitions:

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


Create and Manage Attribute Queries and Assertions

The certificate validation definitions and authentication definitions you create can include LDAP attribute queries for finding and checking specified data from a request against entries in a directory.

Create and Manage Attribute Query Definitions

Create or manage attribute query definitions by performing one of the following actions on the Attribute Query Definitions screen:

To create an attribute query definition, click  and go to *Specify General Details for a Query* on page 113.

To copy an attribute query definition, click . When the LDAP Attribute Query Definition Wizard starts, type a unique name for the query you copied. Go to *Specify General Details for a Query* on page 113 for more information about specifying parameters on the General screen.

To edit an attribute query definition, click  and go to *Edit an Attribute Query Definition* on page 118.

To delete an attribute query definition, click . Click **OK** to confirm the deletion. The attribute query definition name has been removed and no longer displays in the list.

Specify General Details for a Query

On the LDAP Attribute Query Definition Wizard-General screen, complete the following steps to define general details for a query:

1. On the Attribute Query Definitions screen, click .

2. Specify a name and description for the query.

Note: If you are creating an attribute query definition for a CV request that requires certificate-based routing, the CV definition must include an LDAP attribute query called Routing Names. Construct the Routing Names query to use the subject from the CV request to look up a corresponding attribute (group name) that can be returned to the client application for use in determining the appropriate connection for routing.

3. In the Connection Specification section, specify one of the following connection methods to the LDAP server:

- ◆ Use globally defined connection—Select this option to use a connection definition that you have already created. The protocol, host, and port fields for connection to the LDAP server are automatically populated.
- ◆ Use authenticated user's connection—This option is only available for attribute queries within LDAP authentication definitions. If you select this option, the query is submitted over the bound session created when the user specified in the authentication request is authenticated. This prevents the need to perform an additional bind operation to the LDAP server, or to specify the login credentials or other parameters required to perform the bind. For the query to succeed with this connection option selected, the user must have read permission over the scope of the search specified by this query definition.
- ◆ Define connection info with query—Select this option if you want to specify protocol, host, and port information on the Query Parameters screen.

4. In the Query specification section, specify how to perform the LDAP attribute query after connection:

- ◆ Specify query parameters—Select this option to query for attributes using parameters specified on the Query Parameters screen.
- ◆ Specify query as URL—Specify a valid URL to use to perform the LDAP attribute query. The following example shows a valid LDAP URL format:

```
ldap://host:port/BaseDN?Attributes?Scope?SearchFilter
```

Paste the URL in the text box and confirm that the URL includes the appropriate elements to perform the query as required URL elements are described below:

URL Element	Description
Protocol	The protocol to use to connect to the LDAP server. Choose ldap:// (nonsecure) or ldaps://(secure) as the protocol.
Host	The host name of the LDAP server.
Port	The port number to use to connect to the LDAP server.

URL Element	Description
Scope	<p>The starting point used when performing the search. Specify one of the following options:</p> <ul style="list-style-type: none"> ◆ BaseDN—Search at the level of the Base DN. This is typically used for retrieving data from a known entry in the directory. You can specify EA variables to represent this element. Refer to Variables for more information. ◆ One Level—Search only the level immediately below the Base DN. ◆ Sub Tree—Search the entire sub-tree below the Base DN.
Search Filter	<p>The search filter (match attributes) used to determine which directory entries in the scope of the search are a match. The search filter (see RFC 2254) can be very complex, but is typically just one or two attribute names and their expected values. You can specify EA variables to represent this element. Refer to Chapter 14, <i>CV and Authentication Definition Variables</i>, for more information.</p>
Attributes	<p>The names of the attribute types to return (return attributes) from the entries that match.</p>

5. Specify query parameters. Refer to *Specify Query Parameters* on page 115.


Specify Query Parameters

The protocol, host, and port fields for connection to the LDAP server are automatically populated if you select a globally defined connection. If you chose the **Define connection info with query** parameter, you must specify protocol, host, and port information.

To specify the protocol, host, and port:

1. On the Query Parameters screen, define the parameters to use to perform the attribute query:

Parameter	Description
Protocol	<p>The protocol to use when connecting to the LDAP server.</p> <ul style="list-style-type: none"> ◆ For a nonsecure connection, select ldap://. ◆ For a secure connection, select ldaps://.
Host	<p>The host name of the LDAP server.</p>
Port	<p>The port of the LDAP server.</p>
Base DN	<p>The starting point in the directory to begin the search. See Chapter 13, <i>Create and Manage Attribute Queries and Assertions</i>, for more information about using variables supported for certificate validation or authentication.</p>

Parameter	Description
Return Attributes	The attributes to return from the LDAP server. Specify multiple return attributes in a comma-separated list, for example, dn,o,uid. See Chapter 13, <i>Create and Manage Attribute Queries and Assertions</i> for more information about using variables supported for certificate validation or authentication.
Scope	The scope of the LDAP search. Valid values are: <ul style="list-style-type: none"> ◆ One Level —Searches only the immediate descendants of Base DN. ◆ Base—Searches only Base DN entry. ◆ Subtree—Searches downward in the directory from Base DN. See Chapter 13, <i>Create and Manage Attribute Queries and Assertions</i> for more information about using variables supported for certificate validation or authentication.
Match Attributes	If Scope is OneLevel or Subtree , use this parameter to specifically identify the location of the entry by matching the attributes specified to the attributes of the entry in the directory. Click  to specify the attributes to match. You can match on any attribute that is stored on your LDAP server. The entry is found only when all of the attributes specified match. See Chapter 13, <i>Create and Manage Attribute Queries and Assertions</i> for more information about using variables supported for certificate validation or authentication.
Query Timeout	The minutes and seconds (format MM:SS) that can elapse before the LDAP Attribute Query times out and processing ends.

2. On the Confirm screen, review the parameters. Click **Save** and **Close**.
3. From the Attribute Query Definitions screen:
 - ◆ Repeat the previous steps to create another attribute query definition.
 - ◆ To create an attribute assertion definition, click **Next** to go the Attribute Assertion Definitions screen.
 - ◆ To create the certificate validation definition without an attribute assertion, click **Next** and from the Attribute Assertion Definitions screen, click **Next** and continue.

Specify Match Attributes

Match attributes are used to specify how search filters are used to find entries in a directory. Specify attributes that you want to compare to entries in a directory by performing the following steps:

1. On the Match Attributes dialog box, click **Name** and type the name of the value that will be used for matching.
2. Click **Value** and type a valid string to specify the value to use for matching. See Chapter 14, *CV and Authentication Definition Variables*, for more information about using variables supported for certificate validation or authentication.
3. Repeat step 1 and step 2 to specify more match attributes as required.

- Click **OK** to save the match attributes.


The attributes you specified are displayed as *Name=Value* in **Match Attributes** on the Query Parameters screen.

Specify LDAP Connection Settings

The LDAP Connection Settings screen indicates how EA connects to the LDAP server for the directory to query.

To specify LDAP connection settings, specify the following parameters on the LDAP Connection Settings screen:

Parameter	Description
Principal Name	If EA is required to authenticate to the LDAP server, type the name of the security principal for the LDAP operation. The security principal is typically specified as a Distinguished Name, but may take other forms depending on the authentication method and the directory.
Principal Password	The password for the security principal specified.
Client Key Certificate Alias	This parameter selects the key/certificate to use from the system keystore during SSL or TLS negotiations with the LDAP server when the LDAP server configuration requires client authentication. This parameter is disabled unless the selected protocol is ldaps://, or the Start TLS option has been set to Yes.
LDAP Version	The version of LDAP you are authenticating against. Valid values are: 2—Specifies LDAP version 2 3—Specifies LDAP version 3 (default)
Authentication Method	The authentication method to use when authenticating security principals. Valid values are: None—No authentication is performed. Default. Simple—Password is authenticated against the password found in the directory. Digest-MD5—SASL authentication method supported by most LDAP v3 servers. GSSAPI—SASL authentication method that negotiates Kerberos V authentication. This is the native authentication used in Active Directory. External—Rely on the transport layer to provide authentication. External authentication is provided through SSL/TLS client authentication. If the LDAP server supports External as an authentication mechanism, the bind principal identified in the client certificate that EA presents during SSL/TLS negotiation is used.
Start TLS	To request TLS encryption using the LDAP v3 extended operation, Start TLS.
Referral Action	The action to take when an authentication request is referred by one LDAP server to another LDAP server. Follow—Follow the referral to the referred directory. Ignore—Ignore the referral. Throw—Ignore the referral and throw an exception.

Parameter	Description
JNDI Properties	JNDI property names and values if the JNDI service provider bundled with the JRE requires any special properties. Click  to specify property names and values.

Specify JNDI Properties for a Connection

If you use a JNDI (Java Naming Directory Interface) service provider that requires special properties, you can assign the appropriate names and values for the custom JNDI properties to associate with a server connection.

To specify JNDI properties for a connection:


1. On the JNDI Properties dialog box, click **Name** and type the value to use.
2. Click **Value** and type a valid string to specify the value.
3. Repeat step 1 and step 2 to specify as many JNDI properties as required.
4. Click **OK** to save the properties.


Attributes specified display as *Name=Value* in **JNDI Properties** on the LDAP Connection Settings screen.

Edit an Attribute Query Definition

To edit 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 an attribute query definition:


1. Click the **General** tab and make the following changes:
 - ◆ Update or enter a description for the query definition in **Description**.
 - ◆ In the Connection specification section, choose a system-wide connection to the directory server or use connection parameters specified on the **Query Parameters** tab.
 - ◆ In the Query specification section, choose a parameter to indicate whether parameters on the **Query Parameters** tab or a URL you type is used to specify query parameters.
2. Click the **Query Parameters** tab and make the following changes:
 - ◆ Specify or change the **Protocol**, **Host**, and **Port** for the connection to the directory server. If another source (such as a system-wide connection) for the query parameters is indicated on the **General** tab, then you cannot change the protocol, host, and port. See *Specify Query Parameters* on page 115 for more information.
 - ◆ Specify or change the Base DN, Return Attributes, and Scope to use for the attribute query. See *Specify Query Parameters* on page 115 for more information.
 - ◆ Click  to change the Match Attributes used to perform the query. See *Specify Match Attributes* on page 116 for more information.
 - ◆ In **Query Timeout**, type the number minutes and seconds (MM:SS) that can elapse with no response from the directory before a query times out.

3. Click the **LDAP Connection Settings** tab and change the following parameters:
 - ◆ Specify the authentication method to use when authenticating security principals in **Authentication Method**.
 - ◆ Change the **Principal Name** and **Password** to use when EA must authenticate to the directory server.
 - ◆ Specify the alias in the keystore associated with the SASL External authentication method in **Client Key Certificate Alias**. This parameter selects the key/certificate to use from the system keystore during SSL or TLS negotiations with the LDAP server when the LDAP server configuration requires client authentication. This parameter is disabled unless the selected protocol is ldaps://, or the Start TLS option has been set to Yes.
 - ◆ Choose **2** or **3** as the **LDAP Version**. Some EA functionality requires LDAP Version 3, so choose this option if the directory supports it.
 - ◆ Choose **Yes** or **No** for **Start TLS** to encrypt the data channel for this connection.
 - ◆ Specify whether EA **Follows**, **Ignores**, or **Throws** (ignores the referral and throws an exception) when processing a referral to another directory.
 - ◆ Click  to specify **JNDI Properties** for the connection. See *Specify JNDI Properties for a Connection* on page 118 for more information.
4. Review the updated parameters on the **Summary** tab, or click the tabs to review changes made. Click **OK**.

Create and Manage Attribute Assertion Definitions

Create or manage attribute assertion definitions by performing one of the following actions on the Attribute Assertion Definitions screen:


To create an attribute assertion definition, click  and go to *Create an Attribute Assertion Definition* on page 119.

To copy an attribute assertion definition, click . On the Add Assertion Definition screen, type a unique name for the assertion you copied in **Name**. Go to *Create an Attribute Assertion Definition* on page 119 for more information.

Create 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.

To define an attribute assertion:


1. From the Attribute Assertion Definition screen, click  to display the Add Assertion Definition dialog box. Specify the following parameters:

Name	Description
Name	Name to assign to the attribute, up to 255 characters and can include alphanumeric characters and special characters, space, underscore (_), and period (.).
Description	A description of the assertion you are defining.
Assertion	Variables and expressions to create the assertion you want to evaluate. String data must be enclosed in variables. Use the following operators: == Equality > Greater than >= Greater than or equal != Inequality < Less than <= Less than or equal && Logical and Logical OR ! Logical NOT () Parenthesis for grouping true Boolean TRUE (case sensitive) false Boolean FALSE (case sensitive)

2. Click **OK**. The attribute assertion definition you created is displayed in the list on the Attribute Assertion Definitions screen.

Edit an Attribute Assertion Definition

You can modify an attribute assertion definition by performing the following steps:

1. On the Attribute Assertion Definitions screen, select the attribute assertion definition you want to edit.
2. Click .
3. Edit the summary for the assertion definition in **Description** or edit the assertion statement in **Assertion**.

CV and Authentication Definition Variables

EA supports using variables in certificate validation and authentication definitions. Variables are resolved at runtime by data from the certificate validation request or authentication request, from the entity's certificate that you are validating, and from data returned from attribute queries you have configured.

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=pki  
User
```

Variables that represent a single element are typically 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`”.

Variables for Certificate Validation Requests

This section describes the variables you can use in definitions associated with certificate validation requests. These variables represent data from the certificate validation request as well as the results of various operations performed during the certificate validation process.

The following table lists the variables used in certificate validation requests:

Variable	Description
Attr	Root node representing the results of all attribute queries.
Cert	Contains the raw data of the end entity X.509 certificate received in the certificate validation request. Also contains the root node of all certificate variables, such as subject and issuer. This variable can be referenced in an attribute assertion statement to perform a binary compare of the certificate received in a request with the certificate returned from an attribute query.
ClientID	Represents the client ID passed in the request. This variable depends on the client application. For example, if the client application is Connect:Direct, the client ID is generally the node name. If the client application is Sterling Secure Proxy, the client ID passed is the adapter name.
Exit	Root node containing any output variables set by a custom exit.

Variable	Description
Ext	Represents the X.509 V3 extensions of the end entity certificate, serving as the parent node of each extension variable. See Chapter 8, <i>Using X.509 Extensions</i> , for details.
ipAddress or IP	<p>Represents the IP address specified in the request, formatted in dotted decimal notation, with leading zeros omitted. Include this variable to use the IP address from the client application as an authentication factor.</p> <ul style="list-style-type: none"> ◆ v—Represents the IP version (either 4 or 6 in the request). ◆ x—The hexadecimal representation of the IP address. For example, if IP = 10.20.30.40, then IP.x=0x0a141e28. ◆ 0-3—Indices for individual nodes of the dotted decimal. For example, if IP = 10.20.30.40, then IP[0]=10, IP[1]=20, IP[2]=30, and IP[3]=40.
Issuer	<p>Represents the certificate issuer field of the end entity certificate, serving as the parent node of the following issuer attribute variables:</p> <ul style="list-style-type: none"> ◆ CN—Common Name ◆ L—Locality Name ◆ ST—State or Province Name ◆ O—Organization Name ◆ OU—Organizational Unit Name ◆ C—Country Name ◆ STREET—Street Address ◆ DC—Domain Component ◆ UID—User ID
Subject	<p>Represents the certificate subject field of the end entity certificate, serving as the parent node of the following subject attribute variables:</p> <ul style="list-style-type: none"> ◆ CN—Common Name ◆ L—Locality Name ◆ ST—State or Province Name ◆ O—Organization Name ◆ OU—Organizational Unit Name ◆ C—Country Name ◆ STREE —Street Address ◆ DC—Domain Component ◆ UID—User ID

Variable	Description
ssl	<p>Represents variables associated with the SSL session this certificate validation request is authenticating. These variables include Server and Client.</p> <ul style="list-style-type: none"> ◆ Server—Indicates whether the certificate belongs to the server in the SSL session. This is a Boolean variable that is set to true or false. ◆ Client—Indicates whether the certificate belongs to the client in the SSL session. This is a Boolean variable that is set to true or false.

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
```

Variables for Authentication Requests

The variables described in this section are valid when a client application sends an authentication request. If the authentication request is the continuation of a certificate validation request, then the variables set during certificate validation are also available to the authentication service. When multiple factors are checked for authentication, this feature allows correlation of the different factors, for instance, to verify that the certificate subject is the same as the LDAP user.

The following table describes variables used in authentication requests.

Variable	Description
UserID	Represents the user ID received in the authentication request.
Password	Represents the password received in the authentication request.
DestinationService or dstSvc	Represents the destination service name received in the authentication request.
Principal	Represents the authentication principal to be bound to the directory, after it is determined in the LDAP authenticator. This variable may be passed directly in the request or may be the result of a directory search.
ipAddress or IP	Represents the IP address specified in the request formatted in dotted decimal notation, with leading zeros omitted. Include this variable to use the IP address from the client application as an authentication factor. You can use the full variable name, ipAddress for this variable.

Variable	Description
ipAddress or IP	<p>Represents the IP address specified in the request, formatted in dotted decimal notation, with leading zeros omitted. Include this variable to use the IP address from the client application as an authentication factor.</p> <ul style="list-style-type: none">◆ v—Represents the IP version (either 4 or 6 in the request).◆ x—The hexadecimal representation of the IP address. For example, if IP = 10.20.30.40, then IP.x=0x0a141e28.◆ 0-3—Indices for individual nodes of the dotted decimal. For example, if IP = 10.20.30.40, then IP[0]=10, IP[1]=20, IP[2]=30, and IP[3]=40.
ClientID	<p>Represents the client ID passed in the request. This variable depends on the client application. For example, if the client application is Sterling Secure Proxy client, the client ID passed is the adapter name.</p>
Exit	<p>Root node containing any output variables set by a custom exit.</p>

X.509 Extensions

Certificate extensions were introduced in version 3 of the X.509 standard for certificates. These v3 extensions allow certificates to be customized to applications by supporting the addition of arbitrary fields in the certificate. In this manner, 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, which includes a broad range of 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	Identifies whether the subject of a certificate is a Certificate Authority (CA) and describes the maximum depth of a valid path for the certificate.
CRL Distribution Points	Identifies 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 address this requirement and to prevent interoperability issues, EA provides the following mechanisms for extension support:

- Allow and require settings to explicitly allow and disallow specific extension as a means of preventing 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 a particular extension to appear in a certificate. If an extension that is disallowed appears in a certificate, the Certificate Validation Request fails. Similarly, it provides support for an application to explicitly require that particular extensions appear in certificates. If a required extension is not included in a certificate, then the Certificate Validation Request fails.

Boolean Expressions for Extension Properties

EA also provides a mechanism for an application to have very specific control of the interpretation and enforcement of a particular extension. The general model for extension handling is that 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 to each. 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**.

For each property, there is a default Boolean expression that you can modify or replace. If the expression does not evaluate to true, 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 certain keywords such as true and false. Variables available for use in these expressions include all the variables derived from the certificate, such as subject and issuer attributes, plus additional 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*. However, depending on where it is referenced, 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, the 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 simply state that it is not legal to set both the encipherOnly and decipherOnly bits in the same certificate. The third rule adds to this 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. Using CRL Definitions you create and the CRL information included in some certificates, EA can locate the appropriate directory and CRL.

When the CRLDistributionPoints extension references a CRL definition, the CRL definition provides all 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 specified 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

The properties configured for the CRLDistributionPoints extension deviate from the general “Client|Server|CA-*ExtensionName*” properties discussed so far. Instead, 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.

These properties are discussed more in the following sections.

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 in the following situations:

You do not use a CRL definition.

A single CRL definition is configured to support all distribution points.

The definition is referenced directly by name, without the use of variables.

Variables are necessary in the following situations:

You use multiple CRL definitions to access multiple directories.

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-CrIDef” Host=“ldap1”

Name=“DP2-CrIDef” Host=“ldap2”

Then set the CrIDistributionPoints properties as follows:

Ignore CRL Distribution Point: false

Referenced CRL Definition: {distributionPoint.cn}-CrIDef

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, you should 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.

Manage Users and Roles

EA can be used by many administrators who have different network administration and configuration tasks to perform. Use the following procedures to customize the user definitions and role definitions:

Manage Users

Manage Roles

Manage Users

User definitions identify users in EA. When you define users in the system, you specify a user name and password and assign the user a 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 141 to create additional roles that enable you to allow only the required permissions for users.

To manage definitions, complete the following procedures:

Create a User Definition on page 139


Change a User Definition on page 140

Delete a User Definition on page 140

Create a User Definition

To create or copy a user definition:

1. From the **Manage** menu, select **Users**.


- On the External Authentication User Definitions window, click  and specify the following parameters:

Parameter	Description
Name	Name used to login.
Password	The password associated with the user name.
Confirm Password	Reenter the password to confirm it.
Role	Choose the role that the user will sign on as. By default, the admin role is the only one available for assignment.
Description	A description of the user and role.
Properties	This parameter is not used.

- Click **Save**.


Change a User Definition

To change a user definition.

- From the **Manage** menu, select **Users**.
- Select the user definition to edit and click .
- On the **Update User** dialog box, update the user definition by making the required changes:
 - To change the password, type the new password in the **Password** and **Confirm Password** fields.
 - To change the user role, click the **Role** drop-down arrow to select the role to assign to the user definition.
- Click **Save**.

Delete a User Definition

Complete the following procedure to delete a user definition. You cannot delete a user that is currently logged in.

- From the Manage menu, select **Users**. The User Definitions screen is displayed with a list of the current user definitions.
- Select the user definition to delete and click .
- Click **OK** to confirm the deletion.

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:

Parameter	Description
Role name	Type the name you want to use to identify the role. You select this role name when you assign a role during creation or modification of a user definition.
Description	Type a description of the role to help administrators determine when to assign the role.
Permissions	Under each permission category, select a check box to allow the Create, Read, Update, Delete, and Execute permissions for configurable functionality as is appropriate for the role. Clear selected check boxes to disable a permission for the role.
Select All	Select this check box to turn on all permissions in all permission categories. Clear the check box to disable all permissions for all categories. Tip: To save time when creating a role that needs several permissions in multiple permission categories, choose Select All . You can then customize the role by clicking to clear the check boxes for any unnecessary permissions in the Cert Validation, Cert Revocation, Authentication, Acceptor, User, Role and System categories.
Cert Validation	Identify the certificate validation permissions allowed for the role: <ul style="list-style-type: none"> ◆ Create—Allows users assigned the role to create certificate validation definitions. ◆ Read—Allows users assigned the role to read certificate validation definitions. ◆ Update—Allows users assigned the role to update certificate validation definitions. ◆ Delete—Allows users assigned the role to delete certificate validation definitions. ◆ Execute—Allows users assigned the role to execute certificate validation definitions.


Parameter	Description
Cert Revocation	<p>Identify which certificate validation permissions are allowed for the role:</p> <ul style="list-style-type: none"> ◆ Create—Allows users assigned the role to create certificate revocation list definitions. ◆ Read—Allows users assigned the role to read certificate revocation list definitions. ◆ Update—Allows users assigned the role to update certificate revocation list definitions. ◆ Delete—Allows users assigned the role to delete certificate revocation list definitions. ◆ Execute—Allows users assigned the role to execute certificate revocation list definitions.
Authentication	<p>Identify which authentication permissions are allowed for the role:</p> <ul style="list-style-type: none"> ◆ Create— Allows users assigned the role to create authentication definitions. ◆ Read— Allows users assigned the role to read authentication definitions. ◆ Update— Allows users assigned the role to update authentication definitions. ◆ Delete— Allows users assigned the role to delete authentication definitions. ◆ Execute—Allows users assigned the role to execute authentication definitions.
Acceptor	<p>Identify which acceptor permissions are allowed for the role:</p> <ul style="list-style-type: none"> ◆ Create— Allows users assigned the role to create accepters. ◆ Read— Allows users assigned the role to read accepters. ◆ Update— Allows users assigned the role to update accepters. ◆ Delete— Allows users assigned the role to delete accepters. ◆ Execute— Allows users assigned the role to execute accepters.
User	<p>Identify which user permissions are allowed for the role:</p> <ul style="list-style-type: none"> ◆ Create— Allows users assigned the role to create user definitions. ◆ Read— Allows users assigned the role to read users definitions. ◆ Update— Allows users assigned the role to update user definitions. ◆ Delete— Allows users assigned the role to delete user definitions. ◆ Execute— Allows users assigned the role to execute user definitions.

Parameter	Description
Role	<p>Identify which role permissions are allowed for the role:</p> <ul style="list-style-type: none"> ◆ Create—Allows users assigned the role to create role definitions. ◆ Read— Allows users assigned the role to read role definitions. ◆ Update— Allows users assigned the role to update role definitions. ◆ Delete— Allows users assigned the role to delete role definitions. ◆ Execute— Allows users assigned the role to execute role definitions.
System	<ul style="list-style-type: none"> ◆ Create—Allows users assigned the role to create system-wide connections and add other parameters from System Settings on the Manage menu. ◆ Read— Allows users assigned the role to read system settings. ◆ Update— Allows users assigned the role to update system settings. ◆ Delete— Allows users assigned the role to delete system settings. ◆ Execute— Allows users assigned the role to execute system settings.

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. On the External Authentication Role Definitions screen, select the role definition to edit and click .
3. On the **Update Role** dialog box, update the role definition by making the required changes:
4. Click **OK** to save the changes.

Delete a Role Definition

Complete the following steps to delete a role definition. You cannot delete a role that is assigned to a user who is currently logged in.

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** to confirm the deletion.

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 to the layout, highlight the column name to be added in the **Available Columns** list and click >.
3. To remove a column from the layout, highlight the column name to be removed in the **Selected Columns** list and click <.
4. To rearrange the order of columns, highlight 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. Highlight 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 any column and select **Manage Layouts**.
2. Highlight the layout you want to remove and click **Remove**.
3. Click **Close**.

A

- About Sterling External Authentication Server 9
- Add CV definition custom extension 76
- alias alias_name, field definition 47
- Allow certificate extension display 130
- Application output definition
 - create 82, 87
 - create for TAM 110
 - create for Tivoli Access Manager 110
 - create LDAP definition 80, 87
- Attribute assertion definition
 - create 119
 - CV, copy 75
 - CV, delete 75
 - defined 17
- Attribute query definition
 - copy 113
 - Create 113
 - CV, copy 74
 - CV, delete 74
 - CV, manage 74
 - defined 17
 - edit 118
 - manage 113
 - subject verification query parameters 69
- Attribute Query Definition Wizard-General details 113
- Attribute query, FindCert 124
- Attributes
 - specify match 116
- Authentication
 - create TAM definition 108
 - supported variables 126
 - TAM logging information 107
 - Tivoli Access Manager 107
- Authentication definition
 - copy TAM 111
 - create application output for LDAP 80, 87

- create for TAM 107
- create generic 93
- create LDAP 77, 86
- defined 16
- delete 84, 89, 111
- edit application output 84, 88
- edit LDAP 84, 88
- edit TAM 111
- elements 18

- Authentication interaction diagram 11

B

- BasicConstraints extension 132
- Boolean operators 131

C

- CA-signed certificate, defined 15
- Cert variable 124
- Certificate
 - Boolean expression for extensions 130
 - create CV definition 63
 - create revocation lists 70
 - create self-signed certificate for GUI 46
 - define subject verification query parameters 69
 - delete validation definition 74
 - disallow extension display 130
 - display extension 130
 - extension variables 131
 - extension, configure 130
 - import to server keystore 51
 - Java class files, specify 66
 - manage validation definitions 71
 - preconfigure subject parameters 68
 - system, manage 41
- Certificate Authority (CA), defined 15
- Certificate chain, defined 15
- Certificate issuer, variables 124
- Certificate Revocation List (CRL) 57, 70, 129

Index

- Certificate Revocation List (CRL) definition, defined 17
- Certificate Revocation List (CRL), defined 15
- Certificate Signing Request (CSR), defined 15
- Certificate subject
 - abbreviated 124
 - variables 124
- Certificate validation (CV) 63
- Certificate Validation definition
 - defined 16
- Certificate validation definition
 - defined 63
- Certificate validation definitions 63
- Certificate validation steps 10
- Change
 - admin password 34
 - log file size 36
 - logging level 37
 - maximum number of archived log files 37
 - user definition 140
 - user role definition 143
- Column
 - delete layout 147
 - hide 145
 - manage 146
 - rearrange 146
 - rename layout 147
 - resize 146
 - restore 145
 - save layout 146
 - select layout 146
- Configure
 - access to keystore 53
 - access to SSL trust store 54
 - certificate extension 130
 - custom exit for CV definition 65, 94
 - logging from command line 35
 - non-secure listener 33
 - secure listener ports 55
- Connect, GUI to EA 55
- Console, turn logging on or off 36
- Copy
 - attribute query definition 113
 - CRL definition 61
 - CRL definition from CV definition 61
 - CV attribute assertion definition 75
 - CV attribute query definition 74
 - CV definition 71
 - TAM authentication definition 111
 - user definition 139
- Create
 - application output definition 82, 87
 - application output for TAM 110
 - attribute assertion definition 119
 - CRL definition 57
 - CV definition 63
 - key certificate, using -dname option 46
 - LDAP attribute query 84, 89
 - self-signed certificate for GUI 46
 - system certificates 41
 - Tivoli Access Manager 107
 - user definition 139
 - user role definition 141
- create
 - CRL definition 70
- CRL
 - copy definition 61
 - copy definition from CV definition 61
 - create definition 57
 - delete definition 62
 - Distribution Points 133
 - edit definition 61
- CRLDistributionPoints extension 132
- Custom exit
 - configure 65, 94
 - defined 17
 - Java class files, specify 66
 - OS command, specify 67, 95
 - prerequisites 65, 94
 - redirect standard error 68, 97
 - redirect standard output 68, 97
 - timing 67, 96
- Custom extension
 - add for CV definition 76
 - defined 17
 - delete for CV definition 76
- Customize layout views 145
- CV attribute assertion definition
 - copy 75
 - delete 75
 - edit 72
- CV attribute query definition
 - copy 74
 - Delete 74

- edit 72
- CV definition
 - configure custom exit 65, 94
 - copy 71
 - copy CRL 61
 - edit 71
 - edit components 72
 - edit custom extensions 73
 - edit extensions 73
 - elements 17

D

- Define
 - LDAP server for query 117
- Definitions
 - attribute query 113
 - authentication, delete 111
 - authentication, manage 83, 88, 111
 - create system-wide connections 39
 - CV attribute assertion, edit 72
 - CV attribute query, edit 72
 - CV custom, edit 73
 - edit CV components 72
 - edit CV extensions 73
 - roles, change 143
 - roles, create 141
 - roles, delete 143
 - security terms 14
 - TAM authentication, copy 111
 - user, change 140
 - user, copy 139
 - user, create 139
 - user, delete 140
 - users 139
 - validation 71
- Delete
 - authentication definition 84, 89, 111
 - certificate validation definition 74
 - column layout view 147
 - CRL definition 62
 - CV attribute assertion definition 75
 - CV attribute query definition 74
 - CV definition custom extension 76
 - user definition 140
 - user role definition 143
- Directory object, loginCredentials 81

- DirectoryName distribution point 133
- Disallow certificate extension display 130
- Distinguished Name (DN) 124
 - defined 15
 - referencing 125
- Distribution point
 - CRL 133
 - DirectoryName 133
 - URI 134
- dname option
 - specify distinguished name for self-signed GUI certificate 46
- dname, field description 47

E

- EA
 - install on Windows 28
 - shutdown from Windows 32
 - shutdown on UNIX 26
- Edit
 - application output 84, 88
 - attribute query definition 118
 - CRL definition 61
 - custom CV definition extensions 73
 - CV attribute assertion definition 72
 - CV attribute query definition 72
 - CV definition 71
 - CV definition components 72
 - Java class 72
 - LDAP authentication definition 84, 88
 - LDAP connection parameters 72
 - listener connections 38
 - Native OS command 72
 - port number of servlet container 34
 - supported CV definition extensions 73
 - TAM authentication definition 111
- Enable, SSL trust store 54
- Enabled, field definition 55
- Expressions
 - keywords 130
- Extension
 - allow certificate display 130
 - BasicConstraints 132
 - CRLDistributionPoints 132

- custom expression 130
- disallow display 130
- support mechanism 130
- variables 131

F

Field definition

- alias alias_name 47
- dname 47
- enabled 55
- genkey 47
- IP Address 55
- keyalg alg_type 47
- keysize keysize 47
- keystore alias 55
- keystore keystore_path 47
- keytool 47
- port 55
- storepass password 47
- trust store file 54
- validity validity_in_days 47

Field definition, trust store password 54

File naming guidelines 13

FindCert, Attribute Assertion statement 124

G

Generic authentication definition, create 93

-genkey, field definition 47

GUI

- change logging level 35
- connect to server 55
- create self-signed certificate 46
- download to remote computer 25
- refresh lists 38
- start from remote computer 31
- start from Windows 30

H

Hide

- columns 145

I

Import

- CA certificate to GUI keystore 51
- CA certificate to server keystore 51

Install

- EA on Windows 28
- UNIX using console installation 23

Interaction with Sterling Secure Proxy and Connect
Direct Secure+ Option 12

IP Address, field definition 55

J

Java

- class files for custom exit 66
- class, edit 72
- JNDI properties, specify 118

JRE, TAM 108

K

-keyalg alg_type, field definition 47

-keysize keysize, field definition 47

Keystore

- configure access 53
- defined 16, 50

Keystore alias field definition 55

-keystore keystore_path, field definition 47

keytool, field definition 47

KeyUsage extension

- extension
KeyUsage 131

L

Layout, select column 146

LDAP

- connection, specify 117
- create application output 80, 87
- create attribute query 84, 89
- create authentication definition 77, 86
- defined 16
- edit connection parameters 72
- server bind options 78

Listener connection

- change settings 38

Listener ports

- configure secure 55

Log

- change logging level 37

- change maximum number of archived files 37
- change maximum size 36
- turn on or off to console 36
- Login to EA GUI 30
- loginCredentials
 - create directory object 81
 - store SSH keys in directory 89
- Logs
 - change logging level from GUI 35
 - configure files 35
- Lookup Login Credentials (loginCredentials) 80, 87

M

- Manage
 - system certificates 41
- Match attributes
 - specify 116

N

- Native OS command
 - edit 72
- Non-secure listener
 - configure 33

O

- Object identifiers (OID) 70
- OID
 - object identifiers 70
- OS command
 - for custom exit, specify 67, 95

P

- Parameters
 - define for Subject Verification Query 69
- Password
 - change for admin 34
- Port, field definition 55
- Prerequisites
 - secure connection 18
- Principal, defined 16
- Private key, defined 15

- Public key, defined 15

Q

- Query parameters
 - specify for LDAP 114, 115

R

- Rearrange
 - columns 146
- Redirect
 - standard error 68, 97
 - standard output 68, 97
- Reference, DN attributes 125
- Refresh GUI lists from the server 38
- Relative Distinguished Name (RDN) 125
 - multiple attributes 125
- Rename
 - column layout 147
- Replace, self-signed keystore 51
- Resize
 - columns 146
- Restore columns 145
- RFC 3280 129
- Roles
 - manage 141
 - parameters 141

S

- Save column layout 146
- sci.schema 80, 87
- Secure connection listener
 - configure 55
- Secure connection prerequisites 18
- Security terms, defined 14
- Self-signed certificate defined 14
- Server bind options for LDAP 78
- Servlet container
 - port number, edit 34
- Session key, defined 15
- Shutdown

Index

- EA from Windows 32
- EA on UNIX 26
- Simple Authentication, defined 15
- SSH key
 - schema to store 89
- SSL keystore
 - defined 53
- SSL trust store
 - enable access 54
- Start
 - GUI from remote computer 31
 - GUI on Windows 30
- storepass password, field description 47
- Subject verification query
 - define parameters 69
- Supported extensions, defined 17
- Syntax
 - variables 121

T

- TAM
 - authentication 107
 - copy authentication definition 111
 - create application output 110
 - create authentication definition 107, 108
 - edit authentication definition 111
 - JRE 108
 - logging information 107
 - output, specify 110
- Timing
 - custom exit, specify 67, 96
- Tivoli Access Manager (TAM) 107
 - Application output, create 110
 - authentication 107
 - authentication definition, create 108
 - create 107
 - logging information 107
- Trust store file, parameter definition 54
- Trust Store password
 - parameter definition 54
- Trust store, defined 16
- Trusted root key defined 15

U

- URI distribution point 134
- User
 - change definition 140
 - change role definition 143
 - copy definition 139
 - create definition 139
 - definitions 139
 - definitions parameters 140
 - delete 140
 - delete role definition 143
 - roles, create 141

V

- Validation request
 - supported variables 121
 - variables 122
- validity validity_in_days, field definition 47
- Variables
 - authentication requests 126
 - cert 124
 - certificate validation 121
 - certificate validation definitions 124
 - certificate validation requests 122
 - syntax 121
- View
 - customizing display 145
 - delete column layout 147
 - rearrange column 146
 - rename column layout 147
 - resize column 146
 - restore column 145

W

- Window display
 - Customize 145
 - delete column layout 147
 - hide columns 145
 - Manage columns 146
 - rearrange columns 146
 - rename column layout 147
 - resize columns 146
 - restore columns 145
 - save column layout 146
 - select column layout 146

X

X.509 v2 129

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 fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

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

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2012.

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

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

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

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

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

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

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

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