

IBM Sterling Secure Proxy

Implementation Guide

Version 3.3



Copyright

This edition applies to the 3.3 version of IBM Sterling Secure Proxy and to all subsequent releases and modifications until otherwise indicated in new editions.

Before using this information and the product it supports, read the information in *Notices* on page 491.

Licensed Materials - Property of IBM

IBM Sterling Secure Proxy

© Copyright IBM Corp. 2005, 2010. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

What's New in This Release	19
Description of Support Requests Resolved for This Release	21
Special Considerations	23
Security Considerations	23
Adapter Considerations	23
Logging Considerations	23
Configuration Manager (CM) Considerations	24
Engine Considerations	25
HSM Considerations	26
Perimeter Server Considerations	26
Single Sign-On Considerations	27
Connect:Direct Select Version 1.2.01 Considerations	27
Known Restrictions	29
Sterling Secure Proxy Overview	31
General Proxy Terminology	31
About a Reverse Proxy	32
About a Forward Proxy	34
About SSL Session Break	34
About SSH Session Break	35
Using Digital Certificates	36
Configuration Overview	36
SSP Architecture	37
Authenticate Trading Partners in the DMZ	39
Certificate Authentication Options	39
User Authentication Options	41
IP Address Checking (Netmap Check)	42
Summary of Authentication	42
Authenticating SSP to the Trusted Zone Application	43
View a Finished SSP Configuration	45
Connect:Direct Reverse Proxy Diagrams	45
Connect:Direct Forward Proxy Diagrams	46
FTP Reverse Proxy	47
HTTP Reverse Proxy	48
SFTP Reverse Proxy	49
Plan Your SSP Configuration	51
Determine the Communications Protocol to Configure	51
Identify Secure Session Requirements for a Connect:Direct, HTTP, or FTP Environment	51
Identify Secure Session Requirements for an SSH (SFTP) Environment.	52

Determine Validation Requirements for Inbound Trading Partners (Inbound Nodes) to SSP	52
Determine Connection Requirements for the Connection to the Sterling Integrator Server or Connect:Direct Node (Outbound Node)	53
Set Up a Password Policy	53
Set Up User Accounts to Configure the SSP Environment	53
Set Up Users for Inbound Connections	53
Set Up Sterling Integrator/Connect:Direct Servers (Outbound Node Servers) in the Trusted Zone	54
Determine Security Requirements for Communications Sessions Between CM and the Engine	54
Configure a Sterling External Authentication Server	54
Configure a remote perimeter Server	54
Install or Upgrade SSP on UNIX or Linux	55
SSP Installation Checklist for UNIX or Linux	55
SSP Startup Worksheet for UNIX or Linux	56
Install or Upgrade the Engine on UNIX or Linux	57
Install or Upgrade CM on UNIX or Linux	58
Obtain a License Key File for UNIX or Linux	59
Create an Engine Definition	59
System Requirements	61
SSP UNIX and Linux System Requirements	61
SSP UNIX and Linux Host System Requirements	61
SSP UNIX or Linux Operating Systems Supported	62
Perimeter Server Requirements in UNIX or Linux	62
Hardware Accelerator Board	62
Hardware Security Module (HSM) Requirements	62
SSP Windows System Requirements	63
SSP Windows Host System Requirements	63
SSP-Supported Windows Operating Systems	63
Perimeter Server Requirements on Windows	63
Client Connections Supported	63
Web Browsers Supported by CM	64
Server Connections Supported	64
Sterling Security Products Supported	65
Cipher Suites Supported	65
Review Resources for UNIX or Linux	65
Install or Upgrade SSP on Windows	67
SSP Installation Checklist for Windows	67
SSP Startup Worksheet for Windows	68
Install or Upgrade the Engine on Windows	69
Install or Upgrade CM on Windows	69
Obtain and Install a License Key File on Windows	70
Create an Engine Definition	70
Install a Remote Perimeter Server	73
Perimeter Server Installation Prerequisites	73
Perimeter Server Installation Guidelines	74
Install remote perimeter Server in a More Secure Network on UNIX or Linux	74
Install a remote perimeter Server in a Less Secure Network on UNIX or Linux	75
Install remote perimeter Server in a More Secure Network in Windows	77
Install remote perimeter Server in a Less Secure Network in Windows	78
Upgrade Perimeter Server in Windows, UNIX, or Linux	79

Restrict the Policy for a remote perimeter Server	79
Upgrade SSP from Version 2.0.x to Version 3.x	83
Upgrade a Single SSP Node.	84
Single Node File Conversion Illustration.	85
Pre-Upgrade Checklist	86
Upgrade Tasks	86
Upgrade SSP Clustered Nodes	88
Cluster Nodes File Conversion Illustration	90
Cluster Nodes Upgrade Checklist.	92
Upgrade an SSP Loading Balancing Environment	93
Load Balancing Nodes File Conversion Illustration	94
Load Balancing Nodes Upgrade Checklist	96
Upgrade a Multiple SSP Nodes Configuration	97
Multiple Node Environment File Conversion Illustration	98
Load Balancing Multiple Node Upgrade Checklist	100
Start and Log On to SSP Version 2.0.x.	101
Export SSP Version 2.0.x Information.	101
Stop Perimeter Server Version 2.0	102
Stop SSP Version 2.0.x.	102
Back Up Version 3.x Configuration Files.	103
Convert Files from SSP Version 2.0.x to Version 3.x	103
Validate an Export File	103
Convert Version 2.0.x Files With New Engine If No Warnings Are Found	104
Convert Version 2.0.x Files With Existing Engine If No Warnings Are Found.	104
Convert Version 2.0.x Files and Ignore Warnings	105
Upgrade Script Options	106
Read the Upgrade Log File.	107
Copy an Adapter	109
Validate the Converted Components in SSP Version 3.x	109
Validate an Engine Definition	109
Validate an Adapter	110
Validate a PS Definition for a PS in a More Secure Zone	110
Validate a PS Definition for a PS in a Less Secure Zone.	110
Validate the Connection Between Engines and CM	111
Maintain Changes to HTTP Properties	111
New Properties in Version 3.x HTTP Adapter.	113
Maintain Changes to FTP Properties	113
New FTP Adapter Properties in Version 3.x	114
Implement Property Changes Made to a Connect:Direct Adapter	114
Change How Many Times a User Can Attempt to Log In Before a Lock Occurs	115
Move Key Certificates Created in SSP 2.0.02 on the HSM	115
Connect:Direct Proxy Configuration	117
Organization of the Connect:Direct Configuration Scenarios	117
Complete Scenario Worksheets	117
Complete and Test Configuration Scenarios	118
Create a Basic Connect:Direct Configuration	118
Basic Connect:Direct Configuration Worksheet	119
Create a Basic Connect:Direct Policy.	120
Create a Connect:Direct Netmap	120

Define the Connect:Direct Adapter Used for the Connection	121
What You Defined with the Basic Connect:Direct Configuration Scenario . .	121
Add SSL/TLS Support	122
SSL/TLS Support Worksheet	123
Secure the Connect:Direct Connection Using the SSL or TLS Protocol . . .	124
Variation on the Add SSL/TLS Support Configuration	124
Configure PNODE-Based Routing	125
PNODE-based Routing Worksheet	125
Configure PNODE-based Routing	126
Configure Mixed Routing	126
Mixed Routing Worksheet	126
Configure PNODE Specified and Then Standard (Mixed) Routing	126
Add Local User Authentication to a Connect:Direct Connection	127
Connect:Direct PNODE Connection (Local User Authentication) Worksheet	127
Add User Authentication to the Connect:Direct Inbound Connection	128
Add Credentials to the Local User Store	128
Copy Data or Run a Program Based on the Success or Failure of a Process Step (Step Injection)	129
Configure Step Injections Worksheet	129
Configure a Step Injection	131
Use Variables in a Step Injection Definition	132
Associate a Step Injection With a Connect:Direct Node	133
Block Connect:Direct Tasks Allowed on a Node	133
Strengthen User Authentication Using EA	134
Authenticate an Inbound Certificate or User Using EA	134
Authenticate a Certificate or User Using EA - Worksheet	135
Authenticate a Connect:Direct Certificate or User Using EA	135
Strengthen the Connection to the SNODE With User Mapping	136
Perform User Mapping Using EA - Worksheet	136
Perform User Mapping Using Information Stored in EA	137
Configure Certificate-Based Routing	138
Summary of Certificate-Based Routing	138
Configure Certificate-Based Routing in SSP	139
Test the Connect:Direct Connections	139
Additional Connect:Direct Configuration Options	140
Define Alternate Nodes for Failover Support	140
Configure IP Address Checking (Netmap Check)	141
Record an Error Message or Shut Down a Connection Based on Protocol Errors	142
FTP Reverse Proxy Configuration	143
Organization of the FTP Configuration Scenarios	143
Complete FTP Scenario Worksheets	143
Complete and Test FTP Configuration Scenarios	144
Create a Basic FTP Configuration	144
Basic FTP Configuration Worksheet	144
Create an FTP Policy	146
Create an FTP Netmap	146
Define the FTP Adapter Used for the Connection	147
What You Defined with the Basic FTP Configuration Scenario	147
Variations on the Basic FTP Configuration	148
Add SSL/TLS Support for an FTP Connection	150

SSL/TLS Support Worksheet	151
Secure the Inbound FTP Connection Using the TLS or SSL Protocol	152
Secure the Outbound FTP Connection Using the TLS or SSL Protocol.	153
Variations on the Add SSL/TLS Support on the Outbound Node	154
Enable a Clear Control Channel for an Outbound FTP Node Connection	154
Add Local User Authentication to the Inbound FTP Connection.	155
FTP Inbound Connection (Local User Authentication) - Worksheet.	156
Add Local User Authentication to the FTP Inbound Connection	156
Add Credentials to the Local User Store	157
Provide Sterling Integrator Credentials to the Outbound FTP Node Using the Netmap	157
Provide Credentials for the Outbound FTP Node Using the Netmap Worksheet.	158
Connect to the Outbound FTP Server Using Credentials from the Netmap.	158
Strengthen Authentication of an FTP Node Using EA	159
Authenticate an Inbound FTP Certificate or User Using EA.	160
Manage Connection Requirements to the Outbound FTP Server Using EA.	160
Authenticate an Inbound FTP Certificate or User Using EA Worksheet.	160
Authenticate the Inbound FTP Node Using EA	160
Connect to Outbound FTP Server Using EA Worksheet	161
Connect to the Outbound Node Using Information Stored in EA	162
Test the Inbound and Outbound FTP Connections.	162
Sample Inbound Node Log.	163
Sample Outbound Node Log	163
Additional FTP Configuration Options.	163
Route an Outbound FTP Connection to Alternate Sterling Integrator Servers	164
Define a Passive Data Outbound Port Range for an FTP Reverse Proxy Adapter.	164
Define a Passive NAT Address for an FTP Reverse Proxy Adapter	165
Define an Active Data Outbound Port Range for an FTP Reverse Proxy Adapter.	165
Use IP Address from a PASV Response For Outbound Data Connections	166
HTTP Reverse Proxy Configuration	167
Organization of the HTTP Configuration Scenarios	167
Complete Scenario Worksheets	167
Complete and Test HTTP Configuration Scenarios	168
Create a Basic HTTP Configuration	168
Basic HTTP Configuration Worksheet	169
Create an HTTP Policy.	170
Create an HTTP Netmap	171
Define the HTTP Adapter Used for the Connection	171
What You Defined with the Basic HTTP Configuration Scenario	172
Variations on the Basic HTTP Configuration	172
Add SSL/TLS Support for an HTTP Connection.	175
SSL/TLS Support for HTTP Worksheet	176
Secure the Inbound HTTP Connection Using the SSL or TLS Protocol.	178
Secure the Outbound HTTP Connection Using the SSL or TLS Protocol	179
Add Local User Authentication to the HTTP Connection	180

HTTP Inbound Connection (Local User Authentication) Worksheet	181
Enable Local User Authentication to an HTTP Inbound Connection	181
Add Credentials to the Local User Store for an HTTP Connection.	181
Provide Credentials to the Outbound HTTP Node Using the Netmap	182
Connect to the Outbound HTTP Server Using Credentials from the Netmap Worksheet.	183
Configure Name and Password to Connect to the Outbound HTTP Server in the Netmap	183
Strengthen Authentication for an HTTP Connection Using EA.	184
Authenticate an Inbound HTTP Certificate or User Using EA	185
Manage Connection Requirements to the Outbound HTTP Server Using EA.	185
Authenticate an Inbound HTTP Certificate or User Using EA Worksheet . .	185
Authenticate the Inbound HTTP Node Using EA	185
Connect to the Outbound HTTP Server Using EA Worksheet	186
Connect to the Outbound HTTP Server Using Information Stored in LDAP. .	187
Test the Inbound and Outbound HTTP Connections	187
Sample Inbound Node Log.	188
Sample Outbound Node Log	188
Additional HTTP Configuration Options	188
Block Common Exploits	189
Change the Values to Block in a URL String	189
Map a URL in HTML Content from the Outbound Server.	189
Define Alternate Nodes for Failover Support for an Outbound HTTP Connection	191
SFTP Reverse Proxy Configuration.	193
Organization of the SFTP Configuration Scenarios	193
Complete SFTP Scenario Worksheets	193
Complete and Test SFTP Configuration Scenarios	193
Create a Basic SFTP Configuration	194
Basic SFTP Configuration Worksheet	194
Create an SFTP Policy.	197
Create an SFTP Netmap	197
Define the Adapter for the SFTP Connection.	199
What You Defined with the Basic SFTP Configuration Scenario	200
Variations on the Basic SFTP Configuration	200
Authenticate an Inbound SFTP Node Against Information Stored in the Local User Store	202
Add Local Authentication to an Inbound Node Worksheet.	203
Add Local Authentication to the Inbound Node Using Password Information	204
Authenticate an Inbound Node Using Key Information.	204
Authenticate an Inbound Node by Comparing Either a Password or a Key to the Local User Store	205
Authenticate an Inbound Node by Comparing Both a Password and a Key to the Local User Store	205
Provide User Mapping Using the Netmap.	206
Provide User Mapping Using the Netmap - Worksheet	206
Connect to the Outbound Server Using Credentials from the Netmap. . . .	207
Strengthen the SFTP User Authentication Using EA	207
Authenticate an Inbound SFTP User or Key Using EA.	208
Authenticate an Inbound SFTP User or Key Using EA Worksheet	208

Authenticate the Inbound User ID and Password Using EA	208
Authenticate the Inbound User ID and Key Using EA	209
Strengthen the Outbound SFTP Connection With EA User Mapping	209
Manage SFTP User Mapping Using EA	209
Perform User Mapping Using EA in an SFTP Environment Worksheet	210
Connect to the Outbound SFTP Node Using Information Stored in LDAP	210
Test the Inbound and Outbound Connections	210
Route an Outbound Connection to Alternate SFTP Servers	211
PeSIT Proxy Configuration	213
SSP and PeSIT Overview	213
Supported PeSIT Software	213
Organization of the PeSIT Configuration Scenarios	213
Complete Scenario Worksheets	214
Complete and Test Configuration Scenarios	214
Create a Basic PeSIT Configuration	214
Basic PeSIT Configuration Worksheet	215
Create a Basic PeSIT Policy	217
Create a PeSIT Netmap	217
Define the PeSIT Adapter Used for the Connection	218
What You Defined with the Basic PeSIT Configuration Scenario	218
Add SSL/TLS Support	219
SSL/TLS Support Worksheet	219
Secure the PeSIT Connection Using the SSL or TLS Protocol	220
Variation on the Add SSL/TLS Support Configuration	221
Configure PNODE-Based Routing	222
PNODE-based Routing Worksheet	222
Configure PNODE-based Routing	222
Configure Mixed Routing	222
Mixed Routing Worksheet	223
Configure PNODE Specified and Then Standard (Mixed) Routing	223
Add Local User Authentication to a PeSIT Connection	223
PeSIT PNODE Connection (Local LogonID Authentication) Worksheet	224
Add User Authentication to the PeSIT Inbound Connection	225
Add Credentials to the Local User Store	225
Provide Credentials to the Outbound PeSIT Node Using the Netmap	226
Provide Credentials for the Outbound PeSIT Node Using the Netmap - Worksheet	226
Connect to the Outbound PeSIT Server Using Credentials from the Netmap	226
Strengthen LogonID Authentication Using EA	228
Authenticate an Inbound Certificate or LogonID Using EA	228
Authenticate a Certificate or LogonID Using EA - Worksheet	228
Authenticate a PeSIT Certificate or LogonID Using EA	229
Strengthen the Connection to the SNODE With LogonID Mapping	230
Perform LogonID Mapping Using EA - Worksheet	230
Perform LogonID Mapping Using Information Stored in EA	231
Configure Certificate-Based Routing	231
Summary of Certificate-Based Routing	232
Configure Certificate-Based Routing in SSP	232
Test the PeSIT Connections	233
Additional PeSIT Configuration Options	234

Define Alternate Nodes for Failover Support	234
Record an Error Message or Shut Down a Connection Based on Protocol Errors	235
Block PeSIT Command from a PNODE	235
Change the Logging Levels	236
Use Perimeter Servers to Manage PeSIT Communications	236
Manage Your SSP Configuration	236
Modify Properties in an Adapter Definition	236
Copy a PeSIT Node	237
Manage Configuration Manager	239
Change the Password for a CM User	239
Change the CM Passphrase on UNIX or Linux	239
Change the CM Passphrase in Windows	239
Change the Listen Port on CM	240
Modify the Timeout Value for a CM Session	240
Modify the Listener Settings for CM	241
Modify Security Settings for CM	241
Modify Logging for Sessions Between CM and the Web Server	241
Modify Connection Settings for Sessions Between CM and the Web Server	242
Unlock a CM Component	242
Modify the Timeout Value for a CM Session	242
Uninstall CM from UNIX or Linux	242
Uninstall CM from Windows	243
Manage SSP Engines	245
View Configured Engines	245
Change the Engine Passphrase on UNIX or Linux	245
Change the Engine Passphrase on Windows	246
Configure the Refresh Interval Between CM and Engines	246
Update the Monitor Display of Engine Information	246
Manually Send a Configuration File to an Engine	247
Change the Listen Port for an Engine	247
Change the IP Address for an Engine	248
Change the Logging Level for an Engine	248
Uninstall the Engine from UNIX or Linux	249
Uninstall the Engine from Windows	249
Modify the Heap Size	251
Modify Engine Heap Size	251
Modify Engine Heap Size on UNIX or Linux	251
Modify Engine Heap Size on Windows	251
Modify Configuration Manager Heap Size	251
Modify the CM Heap Size on UNIX or Linux	251
Modify the CM Heap Size on Windows	252
Manage Adapters	253
Monitor Configured Adapters	253
Stop an Adapter from CM	253
Start an Adapter from CM	253
Manage User Accounts and Passwords	255
Manage Password Policies	255
Create a Password Policy	256
Edit a Password Policy	256
Copy a Password Policy	256
Delete a Password Policy	257

Manage CM User Accounts	257
Create a CM User Account	257
Edit a CM User Account	258
Copy a CM User Account	258
Delete a CM User Account	258
Manage User Stores and User Accounts	259
Create a User Store	259
Modify the User Account Locking Value in the User Store	260
Copy a User Store	260
Delete a User Store	260
Create an Engine User Account	261
Add SSH Keys to a User Account	261
Edit an Engine User Account	262
Copy an Engine User Account	262
Delete an Engine User Account	262
Configure Perimeter Servers to Manage SSP Communications	263
Typical Installation	264
Sample remote perimeter Server Configurations	264
Deployment Option Example—Two remote perimeter Servers on a Computer with Two NIC Cards	265
Deployment Option Example—From More Secure to Less Secure	266
Deployment Option Example—From Less Secure to More Secure	267
Deployment Option Example —External Authentication Perimeter Server	268
Define a remote perimeter Server for a Less Secure Environment	268
Configure a remote perimeter Server in a Less Secure Zone	268
Edit a remote perimeter Server in a Less Secure Zone Definition	269
Modify the Water Mark Values and Local Host Information of a remote perimeter Server in a Less Secure Zone	269
Configure and Edit a remote perimeter Server Definition When Installed in a More Secure Network	269
Configure a remote perimeter Server in a More Secure Zone	270
Edit A More Secure Zone remote perimeter Server Definition	270
Modify Water Mark Values and Local Host Information of a remote perimeter Server Installed in a More Secure Zone	270
Map Perimeter Servers	271
Modify Perimeter Server Properties	271
Configure SSP for Sterling External Authentication Server (EA)	273
EA Server Configuration - Worksheet	273
Configure an EA Server Connection	273
Specify Alternate EA Servers for Failover Support	274
Use a Perimeter Server to Connect to EA	274
Manage Certificates Between SSP Components	275
Use a Common Certificate for the Engine and CM	276
Replace the Factory Certificate with a Common Certificate on UNIX or Linux	276
Replace the Factory Certificate with a Common Certificate on Windows	277
Use Different Certificates for the Engine and CM	278
Replace the Factory Certificate with an Engine Certificate and CM Certificate on UNIX or Linux	279
Replace the Factory Certificate with an Engine and CM Certificate on Windows	280

Restore Factory Certificates	282
Restore the Factory Certificate on UNIX or Linux.	282
Restore the Factory Certificate on Windows	283
Change the Password of the CM Key Store and Trust	284
Change the Password of the CM Key Store and Trust Store on UNIX or Linux	284
Change the Password of the CM Key Store and Trust Store on Windows.	284
Change the Password of the Engine Key Store and Trust Store	284
Change the Password of the Engine Key Store and Trust Store on UNIX or Linux	285
Change the Password of the Engine Key Store and Trust Store on Windows	285
Configuration Utilities	286
Manage SSH Keys for SFTP Transactions	289
About SSH/SFTP	289
SSH Key Implementation Models Using SSP	290
Use Server Authentication for Inbound and Outbound Connections	290
Implement Public Key User Authentication for Inbound and Outbound Connections.	291
Manage Local Host Key Stores and Keys.	292
Create a Local Host Key Store and Import a Key.	292
Edit a Local Host Key	293
Copy a Local Host Key	293
Delete a Local Host Key	293
Copy a Local Host Key Store	294
Edit a Local Host Key Store	294
Delete a Local Host Key Store	294
Manage Authorized User Key Stores and Keys	294
Create an Authorized User Key Store and Import a Key	295
Edit an Authorized User Key.	295
Copy an Authorized User Key	295
Delete an Authorized User Key	296
Copy an Authorized User Key Store.	296
Edit an Authorized User Key Store.	296
Delete an Authorized User Key Store.	296
Manage Known Host Key Stores and Keys	297
Create a Known Host Key Store and Import a Key	297
Edit a Known Host Key.	297
Copy a Known Host Key.	298
Delete a Known Host Key.	298
Copy a Known Host Key Store	298
Edit a Known Host Key Store	299
Delete a Known Host Key Store	299
Manage Local User Key Stores and Keys	299
Create a Local User Key Store and Import a Key.	299
Edit a Local User Key.	300
Copy a Local User Key.	300
Delete a Local User Key.	301
Copy a Local User Key Store	301
Edit a Local User Key Store	301
Delete a Local User Key Store	301
Store System Certificates on a Hardware Security Module (HSM)	303

Enable and Disable the HSM Environment	303
Enable the HSM Environment	304
Disable the HSM Environment	305
Manage Key Certificates	305
Create Self-Signed Certificates	305
Import a Certificate	307
Export a Certificate	308
Obtain a Certificate from the HSM Device	309
Store a Certificate on the HSM Device	311
Copy a Certificate	312
Move a Certificate from One SSP System Certificate Store To Another Store	313
Rename a Certificate on the SSP System Certificate Store	313
Delete a Certificate	314
List Key Certificates on the SSP System Certificate Store	315
List Key Certificates on the HSM Device	315
Load References to Keys on the HSM into the SSP System Certificate Store	316
Update the HSM Password for HSM Key Certificates Stored in the SSP System Store	318
ManageCSRs	319
Create a CSR	319
Update a CSR	321
Delete a CSR	322
List CSRs on the CM Store	323
Retrieve a CSR to Send to a Certification Authority	323
Retrieve the CA-signed Certificate	323
Start and Stop Configuration Manager and the Engine	325
Start the Engine on UNIX or Linux	325
Start the Engine Using a Stored Passphrase	325
Start the Engine And Require a Passphrase	325
Stop the Engine from UNIX or Linux	325
Start the Engine on Windows	326
Start the Engine as a Console Application on Windows	326
Start SSP as an Automatic Windows Service	326
Set Up the Engine to Require a Passphrase Prompt at Startup on Windows	326
Set up the Engine to Start as a Windows Service	326
Stop the Engine from a Windows Console Application	327
Stop the Engine from CM	327
Stop the Engine from Windows Services	327
Run CM on UNIX or Linux	327
Start CM Without Providing a Passphrase at Startup on UNIX or Linux	328
Start CM and Require a Passphrase at Startup on UNIX or Linux	328
Log On to CM on UNIX or Linux	328
Stop CM on UNIX or Linux	328
Run CM on Windows	329
Start CM from Windows	329
Log on to CM from Windows	329
Stop the Engine from Windows	330
Change Log Settings	331

Audit Log	331
Audit Log Parameters	331
Enable SysLog Support in the Audit Log	332
CM Audit Log Events	332
Engine Audit Log Events	332
Secure Proxy Log	333
Secure Proxy Log Parameters	333
Secure Proxy File Output	334
Node Logs	334
Certicom Logs	334
Perimeter Server Log	335
SFTP Logs	336
Maverick Log	336
SFTP Adapter Log	336
Start and Stop a Remote Perimeter Server	339
Start and Stop a Remote Perimeter Server on UNIX or Linux	339
Start and Stop a Remote Perimeter Server on Windows	339
Configure a Single Sign-on Connection to an HTTP Server	341
Flow of Data for Single Sign-On Configuration Between SFG and SSP	341
Configuration Considerations	342
Organization of Single Sign-On Scenarios	342
Configure the Basic Scenario to Enable a Connection to the myFileGateway Application	343
Configure Advanced Features	343
Configure Optional Features	343
Worksheets	344
Field Definitions	344
Basic Single Sign-On Scenario for myFileGateway	344
Configure SSP for Basic Single Sign-On	344
Create an SSO Configuration	344
Create an HTTP Policy to Support a Single-Sign On Connection	345
Create an HTTP Netmap to Support a Single Sign-On Connection to myFileGateway	345
Define the HTTP Reverse Proxy Adapter Used for the Single Sign-On Connection	345
Configure Sterling EA to Support Single Sign-On	346
Prepare SFG to Support Single Sign-On	346
Prepare SFG to Support Single Sign-On on UNIX or Linux	346
Modify SFG to Support Single Sign-On on UNIX or Linux	347
Prepare SFG to Support Single Sign-On on Windows	348
Modify SFG to Support Single Sign-On on Windows	349
Change Sterling Integrator User Accounts for Single Sign-On	351
Verify That SFG is Configured for Single Sign-On	351
Verify the SSP Connections	352
Advanced Features of the Single Sign-On Configuration	353
Customize the SSO Cookie Attributes	353
Configure Sterling Integrator or SFG to use multiple EA servers	354
Allow a Third-Party Provider to Create Tokens	356
Configure EA to Enable a Third-Party Provider to Create Tokens	356
Customize SSP to Use a Login Portal of a Third-Party Application	357
Customize Token Definitions Created by EA	357
Add Single Sign-On Support for FileGateway	358

Configure Single Sign-On for FileGateway	358
Create an SSO Configuration for FileGateway	359
Create an HTTP Policy to Support SSO for FileGateway	359
Define the HTTP Netmap for FileGateway	359
Configure HTML Rewrite	359
Configure an HTTP Adapter for FileGateway	360
Create User Accounts in SFG	361
Verify the SSP Connections	362
Add Single Sign-On Support for Sterling Integrator Dashboard	362
Configure Single Sign-On for Dashboard	362
Create an SSO Configuration for Dashboard	362
Create an HTTP Policy to Support SSO for Dashboard	363
Define the HTTP Netmap for Dashboard	363
Configure HTML Rewrite	363
Configure an HTTP Adapter for Dashboard	364
Modify Sterling Integrator to Support Single Sign-On for Dashboard	364
Create User Accounts in Sterling Integrator	368
Verify the SSP Connections	369
Add Single Sign-On Support for Basic Authentication Applications on Sterling Integrator	370
Configure Single Sign-On for a Basic Authentication Application	370
Create an SSO Configuration for a Basic Authentication Application	370
Create an HTTP Policy to Support a Single-Sign On Connection	370
Define the HTTP Netmap for a Basic Authentication Application	371
Configure an HTTP Adapter for a Basic Authentication Application	371
Create Basic Authentication Application User Accounts in Sterling Integrator . .	371
Verify the SSP Connections	371
Customize the Logon Portal	371
Common User Tasks Managed by the Logon Portal	372
Workflow When the User Initiates a Password Change	372
Workflow When a User Password is Expired or Must Change	372
Workflow When the User Provides Invalid Logon Credentials	372
Workflow When a User Account is Locked	373
Workflow When the User Cannot Change Password	373
Configuration Considerations	373
Organization of Logon Portal Customization Scenarios	373
Customize the Login Page	374
Customize the Welcome Page	375
Configure SSP to Skip the Welcome Page	376
Customize the Change Password Page	377
Customize the Logout Page	378
Customize Password Policy Page	380
Customize User Messages	380
Configure the Forgot Your User ID or Password Page	380
Configure SSP to Use External Logon Portal	381
Configure a Single Sign-on Connection to a Connect:Direct Server	383
Flow of Data for Single Sign-On Configuration Between Sterling Integrator and SSP	383
Configuration Considerations	384
Organization of Single Sign-On Scenarios	384
Configure the Basic Scenario to Enable a Connection to	

Sterling Integrator	384
Configure Advanced Features	384
Configure Optional Features.	385
Worksheets.	385
Basic Single Sign-On Scenario	385
Configure SSP for Basic Single Sign-On	385
Create a Connect:Direct Policy to Support a Single-Sign On Connection . .	385
Create a Connect:Direct Netmap to Support a Single Sign-On	
Connection to Sterling Integrator	386
Define the Connect:Direct Adapter Used for the Single Sign-On	
Connection	386
Configure Sterling EA to Support Single Sign-On.	386
Prepare Sterling Integrator to Support Single Sign-On	386
Prepare Sterling Integrator to Support Single Sign-On on UNIX or Linux . .	387
Modify Sterling Integrator to Support Single Sign-On on UNIX or Linux . . .	388
Prepare Sterling Integrator to Support Single Sign-On on Windows	389
Modify Sterling Integrator to Support Single Sign-On on Windows	390
Change Sterling Integrator User Accounts for Single Sign-On.	391
Verify That Sterling Integrator is Configured for Single Sign-On	391
Verify the SSP Connections	392
Advanced Features of the Single Sign-On Configuration	393
Allow a Third-Party Provider to Create Tokens.	393
Configure EA to Enable a Third-Party Provider to Create Tokens	393
Customize Token Definitions Created by EA	394
Configure Sterling Integrator or SFG to use multiple EA servers	395
Configure Change Password Portal	396
Change Password Portal Page Flow	396
Configuration Considerations	397
Configuration Overview.	397
Create an SSO Configuration for Change Password Portal.	397
Define a Policy for Change Password Portal	398
Define a Netmap for Change Password Portal.	398
Define an HTTP adapter for Change Password Portal.	398
Organization of Change Password Portal Customization Scenarios	399
Customize the Login Page	399
Customize the Change Password Page	401
Customize the Logout Page	402
Customize Password Policy Page	404
Customize User Messages.	404
Configure the Forgot Your User ID or Password Page.	404
Configure SSP to Use External Logon Portal.	404
Configure a Single Sign-on Connection to an FTP Server	407
Flow of Data for Single Sign-On Configuration Between Sterling Integrator	
and SSP	407
Configuration Considerations	408
Organization of Single Sign-On Scenarios	408
Configure the Basic Scenario to Enable a Connection to	
Sterling Integrator.	408
Configure Advanced Features	408
Configure Optional Features.	409
Worksheets.	409
Basic Single Sign-On Scenario	409

Configure SSP for Basic Single Sign-On	409
Create an FTP Policy to Support a Single-Sign On Connection For FTP	409
Create an FTP Netmap to Support a Single Sign-On Connection to Sterling Integrator	410
Define the FTP Adapter Used for the Single Sign-On Connection	410
Configure Sterling EA to Support Single Sign-On	410
Prepare Sterling Integrator to Support Single Sign-On	410
Prepare Sterling Integrator to Support Single Sign-On on UNIX or Linux	411
Modify Sterling Integrator to Support Single Sign-On on UNIX or Linux	412
Prepare Sterling Integrator to Support Single Sign-On on Windows	413
Modify Sterling Integrator to Support Single Sign-On on Windows	414
Change Sterling Integrator User Accounts for Single Sign-On	415
Verify That Sterling Integrator is Configured for Single Sign-On	415
Verify the SSP Connections	416
Sample Inbound Node Log	417
Sample Outbound Node Log	417
Advanced Features of the Single Sign-On Configuration	417
Allow a Third-Party Provider to Create Tokens	417
Configure EA to Enable a Third-Party Provider to Create Tokens	418
Customize Token Definitions Created by EA	419
Configure Sterling Integrator or SFG to use multiple EA servers	419
Configure Change Password Portal	421
Change Password Portal Page Flow	421
Configuration Considerations	422
Configuration Overview	422
Create an SSO Configuration for Change Password Portal	422
Define a Policy for Change Password Portal	423
Define a Netmap for Change Password Portal	423
Define an HTTP adapter for Change Password Portal	423
Organization of Change Password Portal Customization Scenarios	424
Customize the Login Page	424
Customize the Change Password Page	425
Customize the Logout Page	427
Customize Password Policy Page	428
Customize User Messages	428
Configure the Forgot Your User ID or Password Page	429
Configure SSP to Use External Logon Portal	429
Configure a Single Sign-on Connection to an SFTP Server	431
Flow of Data for Single Sign-On Configuration Between Sterling Integrator and SSP	431
Configuration Considerations	432
Organization of Single Sign-On Scenarios	432
Configure the Basic Scenario to Enable a Connection to Sterling Integrator	432
Configure Advanced Features	432
Configure Optional Features	433
Worksheets	433
Basic Single Sign-On Scenario	433
Configure SSP for Basic Single Sign-On	433
Create an SFTP Policy to Support a Single-Sign On Connection For SFTP	433

Create an SFTP Netmap to Support a Single Sign-On Connection to Sterling Integrator	434
Define the SFTP Adapter Used for the Single Sign-On Connection.	434
Configure Sterling EA to Support Single Sign-On.	434
Prepare Sterling Integrator to Support Single Sign-On	434
Prepare Sterling Integrator to Support Single Sign-On on UNIX or Linux	434
Modify Sterling Integrator to Support Single Sign-On on UNIX or Linux	435
Prepare Sterling Integrator to Support Single Sign-On on Windows	437
Modify Sterling Integrator to Support Single Sign-On on Windows	438
Change Sterling Integrator User Accounts for Single Sign-On.	439
Verify That Sterling Integrator is Configured for Single Sign-On	439
Verify the SSP Connections	440
Advanced Features of the Single Sign-On Configuration	441
Allow a Third-Party Provider to Create Tokens.	441
Configure EA to Enable a Third-Party Provider to Create Tokens	441
Customize Token Definitions Created by EA	442
Configure Sterling Integrator or SFG to use multiple EA servers	443
Configure Change Password Portal	444
Change Password Portal Page Flow	444
Configuration Considerations	445
Configuration Overview.	445
Create an SSO Configuration for Change Password Portal.	445
Define a Policy for Change Password Portal	446
Define a Netmap for Change Password Portal.	446
Define an HTTP adapter for Change Password Portal.	446
Organization of Change Password Portal Customization Scenarios	447
Customize the Login Page	447
Customize the Change Password Page	449
Customize the Logout Page	450
Customize Password Policy Page	452
Customize User Messages.	452
Configure the Forgot Your User ID or Password Page.	452
Configure SSP to Use External Logon Portal.	452
Configure Failover Support.	455
Overview of Failover Support	455
Illustration of Failover Support in SSP	455
Components to Configure for Failover Support	457
About Failover Support.	458
Failover Support for an External Authentication Server	458
Failover For a Back-end Server	459
Overview of Failover Configuration	459
Configure the Load Balancer	459
Summary of Steps to Set Up a Load Balancer for an HTTP Connection	459
Configure the Health Check Monitor for FTP	460
Configure the Health Check Monitor for SFTP	460
Configure the Health Check Monitor for Connect:Direct.	461
Configure Failover Support for an HTTP Environment	461
Configure Failover Support for an FTP Environment	463
Configure Failover Support for an Connect:Direct Environment.	465
Configure Failover Support for an SFTP Environment	467
Configure Advanced Adapter Properties for Failover Support	469
Failover Support Properties	469

Change Failover Support Properties	469
Manage Certificates for SSL/TLS Transactions with Trading Partners	471
About Certificates	471
Certificate Implementation Models Using SSP	471
Implement Certificates that Use a Common Certificate Authority.	472
Implement Self-Signed Certificates	473
Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections	474
Configure a Secure Connection to Sterling External Authentication Server (EA).	475
Use Multiple Key Stores in SSP	476
Import a Public Certificate into a Trusted Certificate Store	477
Import Private Keys into a System Certificate Store	477
Create a New Trusted Certificate Store	477
Create a New System Certificate Store	478
Start and Stop remote perimeter Servers	479
Start a Perimeter Server on UNIX or Linux	479
Stop a Perimeter Server on UNIX or Linux	479
Start Perimeter Servers in a Windows Environment	479
Stop a Perimeter Server on Windows	479
Prepare for Production	481
Configure SSP to Interface with a Load Balancer.	481
Modify the Node-Level TCP Timeout Value in a Connect:Direct Node.	481
Manage Your SSP Configuration	483
Change the Logging Level for a Connect:Direct Node	483
Change the Logging Level for an Inbound Node	483
Change the Logging Level for an Outbound Node	484
Change the Logging Level for a Local Perimeter Server	484
Modify Properties in an Adapter Definition	484
Copy and Delete Engines, Adapters, Netmaps, Nodes, and Policies.	485
Copy an Engine, Adapter, Netmap, or Policy	485
Copy a Node	485
Copy a Connect:Direct Node	486
Delete an Engine, Adapter, Netmap, or Policy	486
Delete an Inbound Node or Outbound Node	486
Delete a Connect:Direct Node	486
Change the User Store Associated With an Engine	487
Filter a Node List	487
Configure Logon Portal or Change Password Portal.	489
Configure Logon Portal	489
Configure Change Password Portal	489

Notices 491

Trademarks.	493
---------------------	-----

What's New in This Release

SSP version 3.3.xx has the following features and enhancements:

Version	Feature or Enhancement
Version 3.3.01	Adds support for Safari 3.2 and 4.0 on Windows and Mac OS.
	Adds support for Firefox 3.5 on Windows and Mac OS.
	Adds 50 additional IP addresses to IP address checking for the Connect:Direct protocol.
	Adds support for 64-bit JREs for Red Hat 5, AIX 5.3, Solaris 10, SuSE SLES 10, HP-UX 11.23 (PA-RISC and HP Itanium), and Windows Server 2008 R2.
Version 3.3	Adds single sign-on (SSO) support for the SFTP, FTP and Connect:Direct protocols.
	Extended HTTP single sign-on (SSO) support allows trading partners to manage their passwords with a self-service mechanism. This change password functionality supports the recommended SSP, EA, and SSO configuration and does not utilize a 3rd party external portal for password management. It improves interoperability between LDAP and Active Directory.
	Allows trading partners to manage their passwords with a self-service mechanism. This change password functionality supports the recommended SSP, EA, and SSO configuration and does not utilize a 3rd party external portal for password management. It improves interoperability between LDAP and Active Directory.
	Improvements to the perimeter server installer.
	User documentation is now available on a live Sterling Web site. It provides improved search, using Google search.
	Adds support for JAAS and RSA SecurID, through Sterling External Authentication Server.

Description of Support Requests Resolved for This Release

No support requests were resolved for Sterling Secure Proxy version 3.3.01. For the history of issues resolved prior to this release, navigate to the Product Updates & Downloads site for your product and platform using the instructions in *Obtaining Product Updates* in the Release Notes PDF and review the Fix List.

Special Considerations

This section contains considerations in addition to the procedures contained in SSP documents.

Security Considerations

Refer to the following security considerations:

- ◆ If you use Sterling External Authentication Server, it uses strong, but limited, cryptography. To use stronger encryption, replace the default jurisdiction policy files with the Unlimited Strength Jurisdiction Policy Files 5.0, available from the JCE provider. Refer to Special Considerations on the Sterling External Authentication Server documentation library.
- ◆ If you use SSP with Connect:Direct UNIX version 3.7.00, you must use the TLS protocol for secure communications.

Adapter Considerations

Refer to the following adapter considerations when configuring or editing an adapter definition:

- ◆ If a Connect:Direct adapter is stopped while one or more sessions are active, the adapter is not restarted until the listen time-out interval expires. To determine if an adapter has active sessions, view the proxy log file.
- ◆ If you configure a Connect:Direct adapter and you do not provide a value for the Ping Response field, the message displayed when a user connects to the host or port has changed to Server Ready. The old message displayed was SecureProxy.
- ◆ If you change one of the following values on an SFTP adapter, you must restart the adapter before the change takes effect: listen port, local host key, selected cipher suites, selected MAC suites and selected key exchange algorithms, compression, maximum sessions, session time-out, inbound perimeter server, outbound perimeter server, or external authentication perimeter server.
- ◆ If you change the perimeter server mapped to an adapter, you must stop and restart the adapter and the perimeter server before the change is enabled.

Logging Considerations

Refer to the following consideration when viewing log files:

- ◆ To check the CM or engine log file on Windows, use the Notepad application. WordPad will not open the log file because the file is locked by the application. If you want to use WordPad, copy the file before opening it.
- ◆ A connection time-out with a remote node causes a listen time-out exception in an SSP log.
- ◆ CM uses the log4j tool to log messages. The configuration file that controls the CM log is called log4j.properties. By default, the logging level is set to INFO and includes warning, error, and informational messages. You can change the log level to DEBUG but this is not recommended because DEBUG generates a large amount of logs and may affect system performance.
 1. To configure the CM log to write debug messages, open the log4j.properties file located in *install_dir/conf* where *install_dir* is the location of the CM installation.

2. To write debug messages for the Jetty web server, add the following statements:

```
# GUI log4j
log4j.logger.sspdashboard=DEBUG,R,stdout
log4j.logger.sspjsf=DEBUG,R,stdout
log4j.logger.com.sterlingcommerce.sspgui.web=DEBUG,R,stdout
log4j.logger.com.sterlingcommerce.csp.gui.web=DEBUG,R,stdout
log4j.logger.com.sterlingcommerce.hadrian.client.gui=DEBUG,R,stdout
```

3. To write debug messages for CM, add the following statements:

```
# CM log4j
log4j.logger.com.sterlingcommerce.component.configurator=DEBUG,R,stdout
log4j.logger.com.sterlingcommerce.hadrian.system=DEBUG,R,stdout
log4j.logger.com.sterlingcommerce.component.accepter.csap.impl.AccepterImpl=INFO,R,stdout
log4j.logger.com.sterlingcommerce.component.dispatcher.XmlConversionFilter=INFO,R,stdout
```

Configuration Manager (CM) Considerations

Refer to the following considerations when configuring CM:

Configure an engine with only one CM. If you use more than one CM, the engine accepts only the CM configuration with a higher version number than its own and with the engine name defined at the first connection.

If you configure CM to use a port already in use by another application, CM will not start.

If you upgrade CM from version 3.2 to version 3.3.01, information about Configuration Manager from version 3.2 is maintained in the new installation to allow the CM from version 3.2 to continue to work for both engines. If you want to change the url used to connect to the server in version 3.3.01, click the Advanced menu. Then select the SSO Configuration to modify. Modify the value in the *Default Landing Page* field. You must also delete the default.app.url property defined in the adapter on the Properties tab; otherwise, the property overrides the value defined in the SSO Configuration.

When you access CM with a browser, a session time-out occurs, if the session is idle for 30 minutes. The Monitoring screen is periodically refreshed, which prevents a session time-out. If the Monitoring screen is displayed as the active application and is not minimized, it does not time out. If you select Monitoring and then minimize the application, a time-out occurs. After a time-out, you may have to login to CM again. Enforce desktop security to prevent unauthorized access to CM.

When you configure a node in a netmap, complete all required fields before you copy a policy or add a new policy. Otherwise, an error message is generated.

When you configure an adapter, complete all required fields before you copy or create a netmap. Otherwise, an error message is generated.

If you are using Firefox, do not open two login sessions on a computer; otherwise, the second login page is blank.

If you use Firefox as your browser and you clear the cache while using CM, restart the browser. Otherwise, it may display unpredictable results.

If you do not store the certificate in the trusted store and you are using Firefox, you receive an error message. Adding the CM certificate to the trusted store prevents this error.

Engine Considerations

Refer to the following considerations when configuring or editing an engine definition:

When you configure an engine, you identify either the host name or the IP address in the definition. Define only one engine definition for each engine you install.

To run two engines that use different network interface cards (NICs) on one computer, install the engine and use the `configureAcceptor` script to change the IP address and port to listen on. Create an engine definition for each NIC card and make sure that the IP address in the engine definition is the same as the one specified in the script. If you want the engine's local perimeter server to use a specific NIC for adapter traffic, you must edit the `perimeter.properties` file and change the `localmode.interface` parameter to the desired IP address. Refer to installation instructions.

When the engine is installed on Solaris 10 and you use SSL or TLS, the AES_256 and DES40 ciphers fail. To enable these ciphers, remove the PKCS11. To remove the PKCS11 provider:

1. Open the `install_dir/jre/lib/security/java.security` file where `install_dir` is the location of the SSP installation.

The first two entries in the file are displayed below:

```
security.provider.1=sun.security.pkcs11.SunPKCS11 ${java.home}/lib/security/sunpkcs11-solaris.cfg
security.provider.2=sun.security.provider.Sun
```

2. Comment out the first entry for `security.provider.1`.
3. Make a copy of the second entry and renumber it to `security.provider.1`.

Leave the second entry as `security.provider.2=sun.security.provider.Sun`. Although the first two entries are the same, you do not have to renumber all security provider lines.

Following is a sample of the new line:

```
security.provider.1=sun.security.provider.Sun
```

4. Save and close the `java.security` file.

If you get the following out of memory error, the engine does not have enough memory to create new threads. This error could result in an abnormal termination.

```
java.lang.OutOfMemoryError: unable to create new native thread
at java.lang.Thread.start0(Native Method)
at java.lang.Thread.start(Thread.java:574)
```

Take one or more of the following actions:

- ◆ If you have more physical memory on the SSP server, install and configure another engine on the server and move some adapters to the new engine. *Install or Upgrade SSP on UNIX or Linux* on page 55 or *Install or Upgrade SSP on Windows* on page 67.
- ◆ Decrease the size of the thread pool defined for SFTP adapters. For each adapter definition, change the following parameters to the same value:
 - `sftp_acceptthreads`—How many threads are available to accept inbound client connections. The default value is 50.
 - `sftp_connectthreads`—How many threads are available for permanent connect threads. When existing SSH connections make socket connections through port forwarding,

these threads manage the asynchronous connection process. Default=50.

- `sftp_xferthreadpools`—How many threads are available for permanent transfers. This thread asynchronously performs the IO for the socket. Default=50.

Refer to *SFTP Adapter* on page 196 for instructions on changing these values.

- ◆ Make more memory available for a Java process by increasing the maximum number of threads allowed. Refer to your operating system documentation.

The default session limit is 20 and allows only about six users with browser sessions. If you use an application such as myfileGateway, FileGateway, or Dashboard, change this limit.

HSM Considerations

Refer to the following considerations when configuring or installing an HSM:

If you run the `manageKeyCerts -list` command, as documented in the and it takes a long time to run or appears to lock, perform the following steps to correct the problem:

1. Open the `install_dir/bin/security.properties` file on the CM computer, where `install_dir` is the location of the CM installation.
2. Insert the following line: `DEF_KEYSTORE_TYPE=JKS`
3. Save the file.

If you upgrade from SSP version 3.1.0 and HSM was enabled, you must run the `setupHSM` script after the upgrade to reenab HSM.

Perimeter Server Considerations

Refer to the following considerations when configuring or editing a perimeter server definition:

If you change the perimeter server associated with an adapter, stop and restart the adapter to implement the change.

If you change a more secure perimeter server configuration, you may need to restart the engine that uses the perimeter server before the changes are enabled.

For a perimeter server installed in a less secure zone, the value of the parameter called `restricted` in the `remote_perimeter.properties` is set to `false` by default. Do not change it.

Before changing a remote perimeter server configuration, first stop all adapters that are using that perimeter server. If you save changes to a perimeter server definition without stopping the adapters that use the perimeter server, errors may occur, the adapters are stopped, and any sessions that are active are stopped. You will be unable to restart these adapters. First stop and restart the remote perimeter server that is used by the adapter and then restart the adapters.

To change the listen port, outbound port range, or perimeter server that a Connect:Direct adapter uses, stop the adapter, make the necessary changes, and enable the adapter.

If you experience a connection failure, refer to the perimeter server log for additional error information.

Some configuration issues exist when using two NIC cards configured with one remote perimeter server. When configuring client software, be sure to identify the correct IP address based on the definition of the external network interface.

When configuring the client software, make sure to use the IP address defined for the external network interface. When using the host name, make sure the host name refers to the IP address specified during the network interface configuration. If not, use the IP address only.

If you change the value of an External Authentication Perimeter Server in an adapter from local to a more secure configuration, restart the perimeter server or the SSP engine.

Single Sign-On Considerations

Refer to the following considerations when configuring single sign-on for Sterling File Gateway:

Sterling File Gateway (SFG) does not support Firefox.

To prevent a token created for one application from accessing a different application, define a unique EA server for each application. For example, if you configure myFileGateway and FileGateway, configure one server for FileGateway and another one for myFileGateway. Each external user in SFG must then be assigned to their respective Authentication Host (EA Server).

If you configure SSO in SSP version 3.2, a Signon directory is created to store SSO files. When you upgrade to version 3.3.01, the Signon directory is backed up. A message is displayed at the end of the installation indicating the where the information is backed up. Compare the contents of the backup Signon directory and the newly-installed Signon directory to ensure that all changes are applied to the new Signon directory.

If you upgrade from version 3.2 to version 3.3.01 and you configured single sign-on, you must update the **Fully Qualified host name the Trading Partner connects to** field in the SSO Configurations object. This field did not exist in version 3.2 but it is required in version 3.3.01.

When you connect through SSP to the Sterling Integrator Dashboard or the Sterling Integrator B2B console applications, you may experience some UI presentation issues, including Manage Layout functionality and some Calendar pop-ups. In addition, Webstart applications will not function correctly through SSP.

Connect:Direct Select Version 1.2.01 Considerations

Refer to the following when configuring Connect:Direct Select version 1.2.01 with SSP:

Connect:Direct Select version 1.2.01 processes zero length messages in a different way than version 1.2.00 and is reported as an issue in the SSP testing database. Therefore, it will not work as configured with Sterling Secure Proxy version 3.1.x or later. To use Connect:Direct Select version 1.2.01 with SSP, turn off empty SSL records.

In a Windows installation, first determine how the SSP engine is running and then modify the appropriate file. To turn off empty SSL records in a Windows installation running as a Windows service:

1. Open the SSPEngine\$.lax file in the *install_dir*\bin directory, where *install_dir* is the location of the SSP engine.
2. Add the parameter, `-DDisableSSLEmptyRecords=true`, to the following section, as shown:

```
lax.nl.java.option.additional=-DDisableSSLEmptyRecords=true -server
-Dorg.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
-Dcom.sterlingcommerce.cspssh.logging.SSHLogger.logger=logrp
-Dcom.sterlingcommerce.cspssh.stats=false -DvendorFile=vendor.properties
-DPlatformFactory=com.sterlingcommerce.csp.perimeter.platform.SSPPlatformFactor
y -Dhadrian.root.dir=.. -Djava.net.preferIPv4Stack=true
```

To turn off empty SSL records in a Windows installation running as a console application:

1. Open the startEngine.bat file.
2. Add the parameter, qq=-DDisableSSLEmptyRecords=true, to the following section, as shown:

```
set qq=-DDisableSSLEmptyRecords=true
"C:\installSSP\gabuild\jre\bin\java.exe" %qq% -server ...
```

To turn off empty SSL records in a UNIX or Linux installation:

1. Open the startEngine.sh file.
2. Add the parameter, QQ=-DDisableSSLEmptyRecords=true, to the following section, as illustrated:

```
QQ=-DDisableSSLEmptyRecords=true
"if test ! -s "${DIST_DIR}/conf/system/sb.enc"
then
"${JAVA_HOME}/bin/java" ${QQ} -server -Xmx${MAXHEAP} -cp ${CLASSPATH} ${F} ${B}
${C} -DvendorFile=vendor.properties
-DPlatformFactory=com.sterlingcommerce.csp.perimeter.platform.SSPPlatformFactor
y -Dhadrian.root.dir=${DIST_DIR} -Djava.net.preferIPv4Stack=true
com.sterlingcommerce.hadrian.Main
else
nohup "${JAVA_HOME}/bin/java" ${QQ} -server -Xmx${MAXHEAP} -cp ${CLASSPATH} ${F}
${B} ${C} -DvendorFile=vendor.properties
-DPlatformFactory=com.sterlingcommerce.csp.perimeter.platform.SSPPlatformFactor
y -Dhadrian.root.dir=${DIST_DIR} -Djava.net.preferIPv4Stack=true
com.sterlingcommerce.hadrian.Main >startEngine.out &
```

Known Restrictions

SSP version 3.3.01 has the following restrictions:

The AES cipher suites do not work with Internet Explorer.

If you use Safari 4.0.4 with SSP, you must add the following line to the startEngine script: "-Dcom.certicom.tls.record.maximumPaddingLength=0".

The only cipher suites supported with Safari 4.0.4 are: TLS_RSA_WITH_RC4_128_SHA, TLS_RSA_WITH_RC4_128_MD5, TLS_RSA_WITH_DES_CBC_SHA, TLS_RSA_EXPORT_WITH_RC4_40_MD5, and TLS_RSA_WITH_3DES_EDE_CBC_SHA.

If you configure an SFTP adapter to use JCE with the property `sftp_jce_enable=true`, only the AES256-CBC, AES192-CBC, AES128-CBC, ARCFOUR, ARCFOUR-128, ARCFOUR-256, 3DES-CBC, and BLOWFISH-CBC ciphers and the HMAC-SHA1 and HMAC-MD5 MACs are supported.

The arcfour SSH ciphers ARCFOUR, ARCFOUR-128, and ARCFOUR-256 only work with JCE.

If you modify the property called `sftp_jce_enable` in an SFTP adapter, you must restart the engine before the change is enabled.

Using JCE for SFTP must be consistent across all SFTP adapters running on an engine. You cannot mix JCE and non-JCE SFTP adapters.

The only cipher suites supported with SSLv2 are: TLS_RSA_WITH_RC4_128_MD5 and TLS_RSA_EXPORT_WITH_RC4_40_MD5.

When using WS_FTP Professional, only version 2007 or later is supported.

If problems occur with WS_FTP Professional version 2007, upgrade to version 2007.11.12 or later.

When connecting to SSP using WS_FTP Professional for SSH-SFTP, if the password and public key are both required for authentication, configure WS_FTP Professional to present the public key first or the connection will fail.

If you use Firefox with SSP, do not open multiple tabs. Opening multiple tabs may cause unexpected results.

If you use Safari and Internet Explorer on the same workstation to access Sterling File Gateway, you must clear the browser cache on both browsers when you switch from one browser to the other.

Enhanced failover support, RSA SecurID, single sign-on, and logon portal functionality do not apply to the PeSIT protocol.

Sterling Secure Proxy Overview

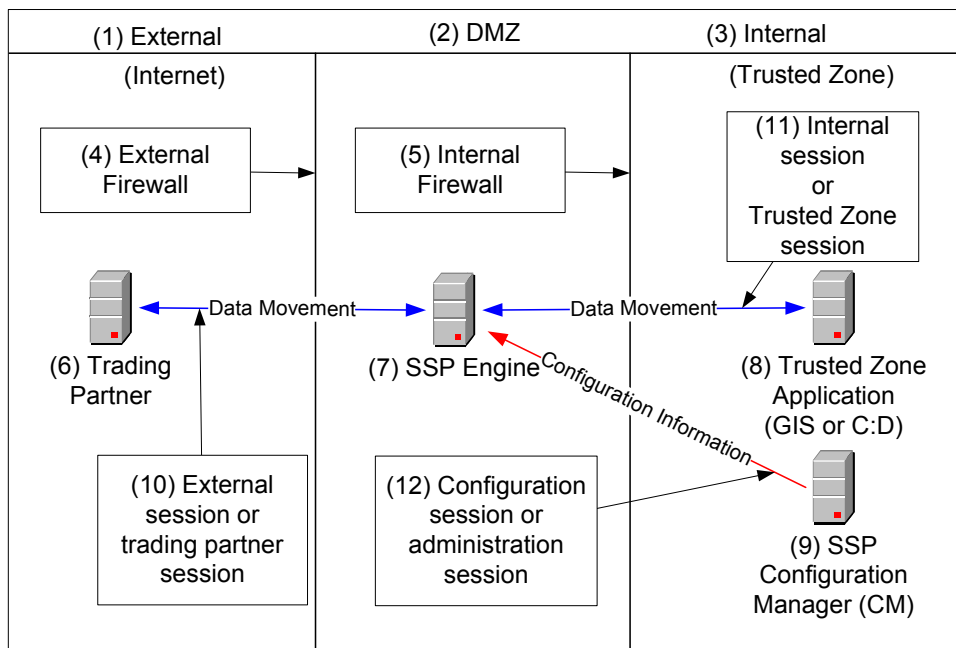
Sterling Secure Proxy (SSP) acts as an application proxy between Connect:Direct nodes or between a client application and a Sterling Integrator (SI) server. It provides a high level of data protection between external connections and your internal network. Define an inbound node definition for each trading partner connection from outside the company and an outbound node definition for every company server to which SSP will connect.

Click one of the following topics to view more information:

- General Proxy Terminology
- About a Reverse Proxy
- About a Forward Proxy
- About SSL Session Break
- About SSH Session Break
- Configuration Overview
- Authenticate Trading Partners in the DMZ
- Summary of Authentication
- Authenticating SSP to the Trusted Zone Application

General Proxy Terminology

Following is an illustration of the SSP general proxy environment:



The following table describes the terminology used in the illustration:

#	Term	Description
1	External Network (Internet)	Network providing connectivity for trading partners to your network. This network is usually the Internet and is called the Internet in this documentation. The external network can also be a private network.
2	DMZ (Demilitarized Zone)	The part of the network that is neither the internal network nor the Internet. It is a network between the two networks. SSP is deployed in the DMZ and provides authentication before a trading partner can access information in the trusted zone.
3	Internal Network (Trusted Network or Trusted Zone)	The internal network behind the internal firewall and secure from outside networks.
4	External Firewall (Outer Firewall)	The firewall between the public network (Internet) and DMZ.
5	Internal Firewall (Inner Firewall)	The firewall between the DMZ and trusted zone.
6	Trading Partner	The external entity that you do business with. Trading partner may also be referred to as remote trading partner, external trading partner, or remote client.
7	SSP Engine	SSP includes two parts: an SSP engine and Configuration Manager (CM). The engine is deployed in the DMZ. It authenticates trading partners and information that is transmitted between the trading partner and trusted zone.
8	Trusted Zone Application	The application in the trusted zone with which the trading partners exchanges information. It is either Sterling Integrator or a Connect:Direct server. It is also called the end point, destination node, or internal application.
9	SSP Configuration Manager (CM)	SSP includes two components: an engine and Configuration Manager. Configuration Manager resides in the trusted zone and configures the engine to perform its duties.
10	External Session (Trading Partner session)	The session between the remote trading partner and SSP.
11	Internal Session (Trusted Zone session)	The session between SSP and the trusted zone application.
12	Configuration Session (Administration session)	The session between CM and the engine that CM is configuring. This session is used by CM to push the configuration to the engine.

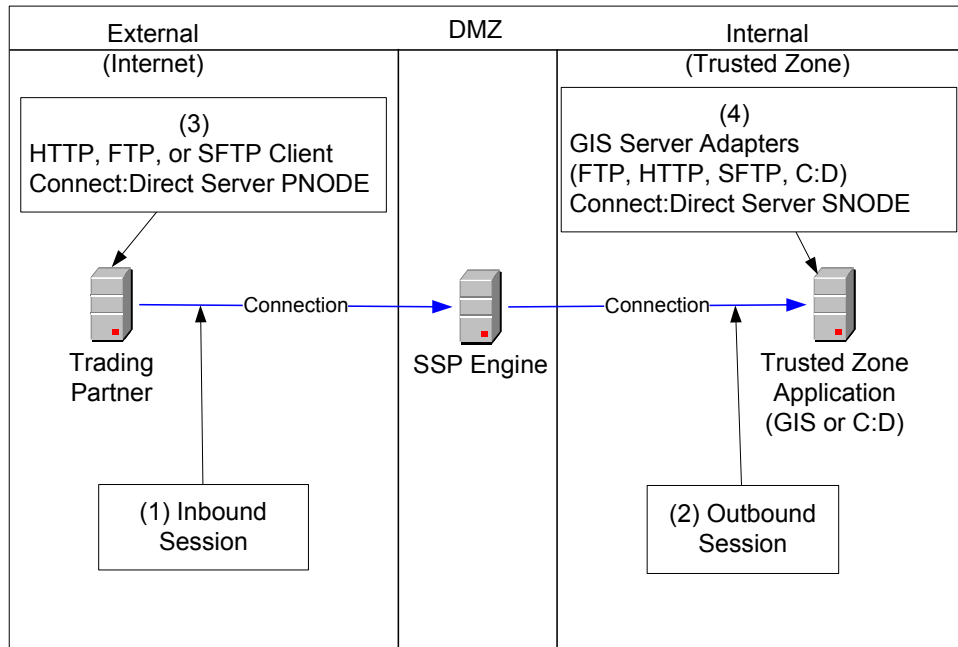
About a Reverse Proxy

A reverse proxy acts on behalf of a trusted zone application. The trading partner or remote client initiates a connection to a trusted zone application and is connected to a reverse proxy.

SSP provides reverse proxy services for Sterling Integrator when the trading partners initiate FTP, HTTP, SFTP, and Connect:Direct sessions to the Sterling Integrator server in the trusted zone.

SSP provides reverse proxy services for Connect:Direct servers when the trading partners initiate Connect:Direct sessions to Connect:Direct servers in the trusted zone.

Following is an illustration of SSP, labeled with reverse proxy terminology.



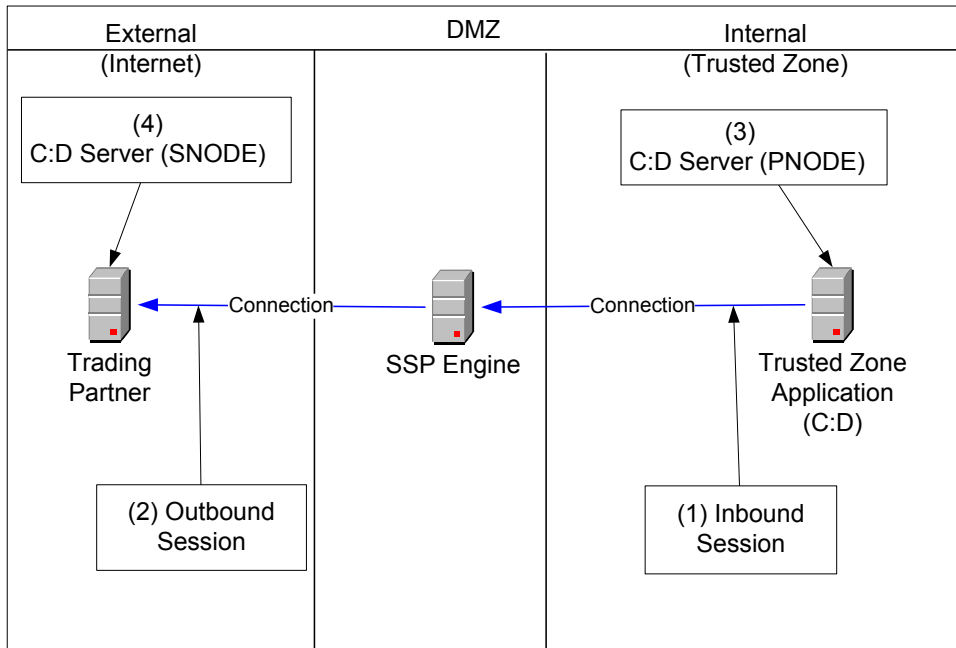
The following table explains the reverse proxy terminology used in the illustration.

#	Term	Description
1	Inbound Session	The session between the remote trading partner and SSP. It is inbound to SSP.
2	Outbound Session	The session between SSP and the Sterling Integrator or Connect:Direct application in the trusted zone. It is outbound from SSP.
3	Client or Connect:Direct PNODE	The trading partner. It can be an HTTP, FTP, SFTP, or Connect:Direct client. For Connect:Direct implementations, the client is the PNODE or the initiating node.
4	Server Adapters or Connect:Direct SNODE	The Sterling Integrator server in the trusted zone. The adapters at Sterling Integrator are the HTTP server adapter, FTP server adapter, SFTP server adapter, and the Connect:Direct server adapter (SNODE). For Connect:Direct implementations, the trusted zone server is the SNODE.

About a Forward Proxy

A forward proxy participates in connections that originate from the trusted zone. The client in the trusted zone connects to the forward proxy in the DMZ and the forward proxy sends connection information to the destination application at the remote trading partner. SSP provides forward proxy services for Connect:Direct servers when the node in the trusted zone initiates a session to a server at a remote trading partner.

Following is an illustration of SSP, labeled with forward proxy terminology.



The following table describes forward proxy terminology used in the illustration.

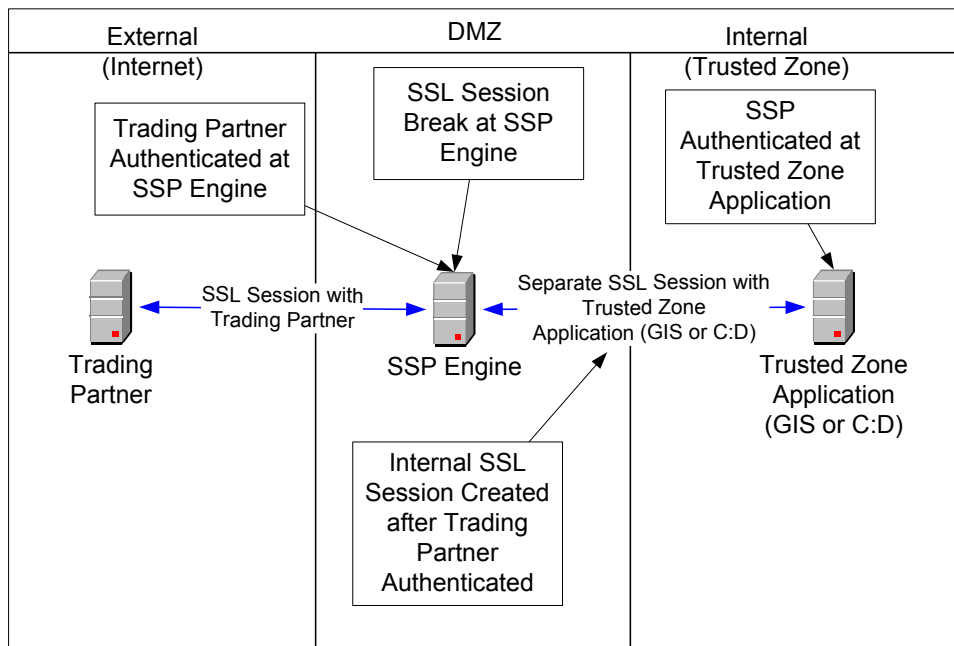
Term	Description
1 Inbound Session	The session between the Sterling Integrator or Connect:Direct application in the trusted zone and SSP and is inbound to SSP.
2 Outbound Session	The session between SSP and the remote trading partner. It is outbound from SSP.
3 Connect:Direct PNODE	The Connect:Direct node in the trusted zone that initiates the session.
4 Connect:Direct SNODE	The Connect:Direct server at the trading partner.

About SSL Session Break

The SSL session break is a primary SSP security feature. SSP authenticates a remote trading partner in the DMZ, before creating a separate SSL session into the trusted zone. This allows you to create firewall rules to prevent trading partners from obtaining direct access to your application in the trusted zone. It also allows you to keep sensitive data out of the DMZ.

The SSL session break occurs because the trading partner connects to SSP in the DMZ and not to the application in the trusted zone. The trading partner is unaware that SSP is deployed and believes it is connecting to your backend system. SSP negotiates an SSL session with the remote trading partner and authenticates the trading partner's certificate, if SSL client authentication is configured. SSP then enforces user authentication to validate that the trading partner uses a valid user ID and password. After the SSL session is established and the user ID and password is authenticated, SSP initiates a separate SSL session to the application in the trusted zone. After the application in the trusted zone authenticates SSP via SSL client authentication and user ID and password authentication, SSP communicates messages between the trading partner and trusted zone application connection to allow the remote trading partner to move data into and out of the trusted zone application.

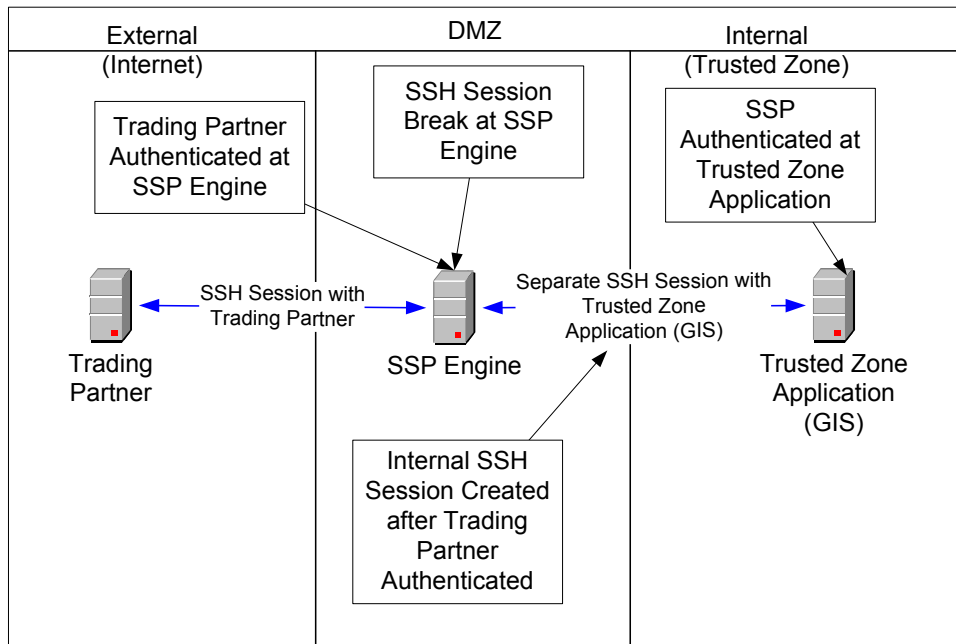
Following is a sample SSL session break flow:



About SSH Session Break

Just as SSP creates an SSL session break for the HTTP, FTP, and Connect:Direct protocols, it creates an SSH session break when using the SFTP protocol. The SSH session break occurs because the trading partner connects to SSP in the DMZ and not to the application in the trusted zone. The trading partner is unaware that SSP is deployed and believes it is connecting to your backend system. SSP negotiates an SSH session with the remote trading partner and authenticates the trading partner's key and/or password as part of the SSH negotiation. After the SSH session is established, SSP initiates a separate SSH session to the application in the trusted zone. After the application in the trusted zone authenticates SSP using key and/or password authentication, SSP relays messages between the trading partner connection and the trusted zone application connection to allow the remote trading partner to move data into and out of the trusted zone application.

Following is a sample flow of an SSH session break:



Using Digital Certificates

SSP uses X.509 digital certificates for secure data transport. Before you set up trading partner information, you must obtain and check in any digital certificates. Certificates can be stored in the SSP store or on a Hardware Security Module (HSM). An HSM is a hardware-based security device that generates, stores, and protects cryptographic keys. SSP provides support for the Safenet and Thales HSMs.

After you store system certificates on the HSM and import information about the system certificates stored on the HSM to the SSP store, all system certificates, including those in the store and on an HSM, are displayed and available when you configure SSP.

Configuration Overview

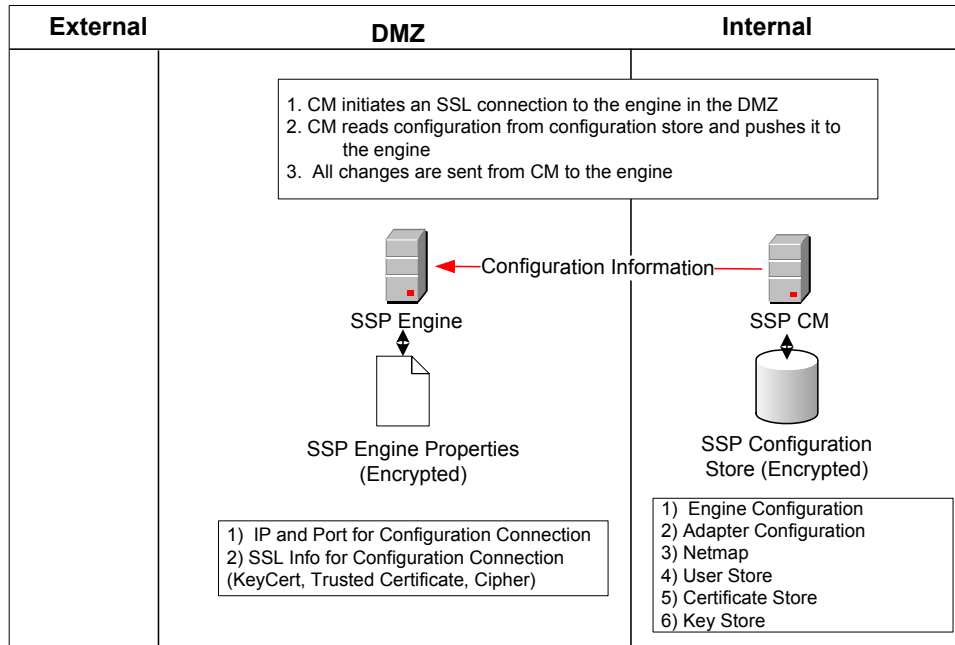
The SSP architecture requires that only the minimum amount of configuration information be stored in the DMZ. It includes two components: the Configuration Manager (CM) and an engine. Configuration data is stored at CM, is encrypted, and does not require a database. CM is installed on the internal or trusted network.

The engine resides in the DMZ and receives configuration data from CM. The engine stores engine properties on disk in the DMZ, and the files are encrypted. The engine properties contain the minimum information required to accept and secure a connection from CM. It includes the IP address and port that the SSP engine listens on for the CM connection. It also includes the SSL key certificate, trusted certificate, and encryption cipher that will be used to secure the connection with CM.

When the engine is first started, it does not have configuration information. It listens on the configured IP address and port for a connection from CM, which tries to connect to the engine at a configurable interval. When CM connects to the engine, they negotiate an SSL session and secure the connection.

After the channel is secure, CM pushes the configuration to the engine. The engine reads the configuration and starts the appropriate proxy services. When you update a configuration in CM, CM transfers the updates to the engine.

Following is an illustration of the SSP flow of a configuration push:



SSP Architecture

SSP architecture is described below:

SSP Engine—the engine resides in the DMZ and contains the minimum components necessary to manage communications sessions. The engine configuration (SSP engine properties) is created at CM and pushed to the engine. It is stored in active memory and is never stored on disk in the DMZ. No web services or UI ports are open in the DMZ.

Configuration Manager (SSP CM)—Configuration Manager is installed in the trusted zone. Use this tool to configure your environment. When you save a configuration definition (SSP configuration store) at CM, it is pushed to an engine, using an SSL session. Configuration files are encrypted and stored on the computer where CM is installed.

Note: Only one Configuration Manager should update an engine definition.

SSP configuration store—This file is encrypted on disk and contains the following information:

- The user store with information on user credentials
- The system certificate store with the certificates used for SSL/TLS sessions
- The key store with the SSH keys
- The engine configuration store with all configuration information for the engine

SSP engine properties file—These files are encrypted and contain the following information:

- The IP and port number to listen on for connections from Configuration Manager
- SSL key certificate, trusted certificate, and encryption cipher used for the connection from Configuration Manager

Web server—Configuration Manager is installed with a web server. You open a browser and access CM through a web page to configure SSP and monitor the engine activity. The web server is installed when you install Configuration Manager.

Adapter—an adapter identifies the protocol allowed for connections from trading partners. You can accept connections from clients that use different protocols; however, you must define a different adapter for each protocol. A single engine can run multiple adapters. In an adapter definition, you identify the port on which to listen for connections, the netmap to use with the adapter, the security policy, and the routing method to use. If you are using External Authentication, you identify the EA server to use in the adapter definition. If you are using a remote perimeter server, you identify the perimeter server to use in the adapter definition.

Netmap—define a netmap to identify the trading partners authorized to communicate through SSP and the company servers where connections are made.

For a Connect:Direct netmap, create a node definition for all Connect:Direct nodes that will communicate through SSP. The node definition identifies the IP address and port to be used by the node and the policy to associate with the node. If SSL or TLS security is required for the connection, configure the protocol options in the node definition. You can also enable node-level logging in the node definition.

For HTTP and FTP netmaps, define an inbound node definition for trading partner connections from outside the company. The inbound node definition identifies the IP address or address pattern to allow for the connection and the policy to associate with the node. If SSL or TLS security is required, configure the protocol options in the node definition. You can also enable node-level logging in the inbound node definition.

For HTTP and FTP netmaps, define an outbound node for every company server to which SSP will connect. An outbound node definition identifies the address and port used to connect to the company server and enables SSL or TLS if this is required. You can also enable node-level logging and failover support in the outbound node definition.

For SFTP netmaps, define an inbound node definition for trading partner connections from outside the company. The inbound node definition identifies the IP address or address pattern to allow for the connection and the policy to associate with the node. You can also enable node-level logging in the inbound node definition.

For SFTP netmaps, define an outbound node for every company server to which SSP will connect. An outbound node definition identifies the address and port used to connect to the company server, the known host key that is used to authenticate the company server to SSP, and the cipher suites and MACs used to secure the connection. You can also enable node-level logging and failover support in the outbound node definition.

Policy—define a policy to identify the security features to implement for an inbound node definition or a Connect:Direct node definition.

In all protocol policies, you can enable the capability to authenticate the inbound connection and identify what user ID and password to use to connect to the secure company server.

For FTP, HTTP, and Connect:Direct policies, you can enable the capability to authenticate certificate information using EA.

In an HTTP policy, you can enable the capability to block commonly occurring HTTP exploits.

In a Connect:Direct policy, you can enable the capability to send a warning message or stop a session if a protocol error occurs, as well as prevent a Connect:Direct node from performing a runtask, runjob, copystep, or submit step function.

In an SFTP policy, you identify the method required to authenticate the inbound connection. Authentication methods supported are key, password, password or key, and password and key.

Sterling External Authentication Server (EA)—a separately installed feature of SSP, EA allows you to validate digital certificates passed by the client or trading partner during SSL/TLS session requests. You can also validate certificates against one or more certificate revocation lists (CRLs), and validate certificates based on a valid date range. See the Sterling External Authentication Server documentation for more information.

EA can be configured to validate certificates and authenticate users. The functions performed by EA are defined in an EA definition. EA performs one or more of the following functions:

- Certificate Validation

- Certificate Revocation List (CRL)—certificate revocation checking using a certificate revocation list (CRL)

- Multi-factor Authentication

- Certificate Policy Enforcement

- LDAP Authentication

- User ID mapping—remote trading partners can be given IDs and passwords that do not provide access to internal systems. The ID and password presented by the trading partner is mapped to an ID and password that can then access the internal system

- TAM (Tivoli Access Manager) Authentication

- Generic Authentication

Before you can use EA with SSP, you must configure EA server definitions in SSP. Then, when configuring policies and protocol adapters, you select these server definitions. You can also select security features available in EA such as certificate authentication, user authentication, and user mapping. Refer to the Sterling External Authentication Server documentation library for more information.

Authenticate Trading Partners in the DMZ

SSP allows you to select an authentication method to meet your security requirements. The authentication mechanisms can be used together to enforce multi-factor authentication.

Authentication options include certificate authentication, user authentication, and IP address checking.

Certificate Authentication Options

You can authenticate a remote trading partner using certificate authentication. Certificate authentication uses SSL client authentication and is optional. Three methods of certificate authentication are available to allow you the flexibility to choose how you want to authenticate

trading partners using x.509 certificates. Certificate authentication options include no authentication, local authentication, or authentication using EA. Authentication using EA provides the highest level of security.

Option	Description
Additional Certificate Authentication Using EA (Recommended)	<p>This method provides the most secure method of certificate authentication. Configure SSL client authentication to use EA to perform additional authentication on the certificate. EA can perform the following authentications:</p> <ul style="list-style-type: none"> ◆ Certificate Revocation List (CRL) checking—validates that the certificate has not been revoked. ◆ Common name check or subject name lookup— validates that the certificate is issued to a trusted trading partner by looking up the name at your LDAP server. ◆ Binary comparison—compares the certificate received to a public certificate. ◆ Bind certificate to an IP address—validates that the certificate and IP address are associated and that the certificate is presented by the IP address identified. ◆ Custom Exit—transmit the certificate to your java program to interface with internal certificate validation routines. <p>Choose this option to enforce the following security policy requirements:</p> <ul style="list-style-type: none"> ◆ Enforce multiple factors of authentication in the DMZ and authenticate the trading partner connection using SSL client authentication and user authentication. ◆ Enforce a single factor of authentication in the DMZ and you plan to authenticate the trading partner connection using SSL client authentication. ◆ To further authenticate the client certificate using a mechanism external to SSP.
Local Certificate Authentication	<p>If SSL client authentication is configured, SSP requests a valid certificate from the trading partner. The certificate is validated against the trusted root.</p> <p>Choose this option to enforce the following security policy requirements:</p> <ul style="list-style-type: none"> ◆ Enforce multiple factors of authentication in the DMZ and authenticate using SSL client authentication and user authentication. ◆ Enforce a single factor of authentication in the DMZ and authenticate using SSL client authentication. ◆ Authenticate using SSL client authentication and do not use EA to provide certificate validation.
No Certificate Authentication	<p>You can configure SSP so that the remote trading partner certificate is not authenticated. Either disable SSL security or turn on SSL security but do not enforce SSL client authentication. In both configurations, SSP will not require the client to send a certificate for authentication.</p> <p>Choose this option to enforce the following security policy requirements:</p> <ul style="list-style-type: none"> ◆ Enforce a single factor of authentication in the DMZ and authenticate a trading partner using user authentication. ◆ You require an SSL session break in the DMZ but you do not want to authenticate the trading partner. In this case, you do not enforce SSL client authentication to SSP nor do you authenticate the user. ◆ The session with the remote trading partner is not secure and does not use SSL. The trading partner does not present a certificate.

User Authentication Options

Three methods of user authentication allow the flexibility to choose how to authenticate users: no user authentication, authenticate users locally or authenticate users using EA. Authenticate using EA is the most secure option. Following is a description of the user authentication methods:

Option	Description
Authenticate Users With EA (Recommended)	<p>Select this option to perform external user authentication, using EA. This option sends the user credentials presented by the client to EA for authentication. Sample user authentication validations that EA can perform include:</p> <ul style="list-style-type: none"> ◆ Through LDAP to bind to user in LDAP ◆ Through Tivoli Access Manager (TAM) ◆ Through a customer java exit <p>Choose this option to enforce the following security policy requirements:</p> <ul style="list-style-type: none"> ◆ To maintain users in an application that is external to SSP ◆ You have an existing infrastructure to validate users against ◆ To use the user mapping provided by EA, refer to the Sterling External Authentication Server documentation library ◆ To implement multi-factor authentication and bind the factors together in the LDAP infrastructure
Authenticate Users Locally	<p>Select this option to authenticate users using information in the SSP local user store. This option requires you to maintain the users in the SSP configuration. Select this option for the following security requirements:</p> <ul style="list-style-type: none"> ◆ To store and maintain users in the SSP user store. ◆ No external infrastructure exists for user authentication to interface with.
No User Authentication	<p>Select this option if you do not want to validate trading partner credentials in the DMZ. If you select this method, we recommend that you enforce SSL client authentication to provide at least one factor of authentication in the DMZ. If you select no user authentication, you may pass the user credentials through to the destination node in the internal network and validate the user credentials at the internal network. Choose this option to enforce the following security policy requirements:</p> <ul style="list-style-type: none"> ◆ Enforce single factor authentication in the DMZ and authenticate the trading partner using SSL client authentication. Pass the user credentials to Sterling Integrator or Connect:Direct trusted zone application so it will authenticate the user and differentiate between users accessing the system. ◆ Use an SSL session break or IP break in the DMZ but do not authenticate the trading partner. Do not enforce SSL client authentication or authenticate the user. Pass the user credentials to the external network in order Sterling Integrator or Connect:Direct to differentiate between users accessing the system. ◆ You implement a bulletin board type system, where user credentials are not important. This option is not a typical implementation. Carefully evaluate your environment before using this configuration.

IP Address Checking (Netmap Check)

IP address checking validates the IP address of the trading partner and makes sure that the IP address is an allowed address. You perform IP address checking with SSP or through EA.

Inbound Node List for the FTP, HTTP, and SFTP protocols—Use SSP to validate the IP address from which a remote trading partner connects. When a trading partner connects to SSP, SSP looks up the IP address in the inbound node list of the netmap. If the IP address is not found, the session ends.

You can specify wildcard characters in the inbound node list, to provide the flexibility to be as granular in your check as you require. For example, you can specify an entry of * in the inbound node list. This value allows connections from all IP addresses. If you specify an IP address for each trading partner in the inbound node list, only connections from the client IP addresses identified are allowed. The more specific the IP address is in the inbound node list, the stricter the IP address check is.

Netmap Check for Connect:Direct—For Connect:Direct connections, the netmap contains one node list that is used for both inbound and outbound nodes. Connect:Direct does not use the IP address to find the netmap entry to use. It uses the node name provided by the initiating node (PNODE). However, a parameter in the Connect:Direct adapter allows you to check the IP address of the initiating node.

External Authentication (recommended)—Validate the IP address using EA to perform certificate or user validation. If SSP is configured to use EA for user or certificate authentication, it sends the IP address to EA. EA validates the IP address and determines if the IP address is valid for a user or for a certificate subject name, common name, or other specified values in the certificate.

Summary of Authentication

SSP provides flexibility in how you authenticate users and connections.

The table below summarizes the options available in SSP and the factor of authentication for each. Recommended options are in bold.

	SSL Client Authentication Enforced			SSL Client Authentication Not Enforced		
	No User Authentication	Users Authenticated Locally	Users Authenticated via EA	No User Authentication	Users Authenticated Locally	Users Authenticated via EA
Pass Through User Credentials	Single factor authentication in DMZ (SSL client auth only)	Multi-factor authentication in DMZ	Multi-factor authentication in DMZ (recommended)	No authentication in DMZ	Single factor authentication in DMZ (user auth only)	Single factor authentication in DMZ (user auth only)
Outbound User Credentials Mapped from EA	N/A	N/A	Multi-factor authentication in DMZ	N/A	N/A	Single factor authentication in DMZ (user auth only)
	No User Authentication	Users Authenticated Locally	Users Authenticated via EA	No User Authentication	Users Authenticated Locally	Users Authenticated via EA

	SSL Client Authentication Enforced			SSL Client Authentication Not Enforced		
Outbound User Credentials from Outbound Node in Netmap	Single factor authentication in DMZ (SSL client auth only.) All users look the same at Sterling Integrator or C:D in trusted zone.	Multi-factor authentication in DMZ. All users look the same at Sterling Integrator or C:D in trusted zone.	Multi-factor authentication in DMZ. All users look the same at Sterling Integrator or C:D in trusted zone.	No authentication in DMZ. All users look the same at Sterling Integrator or C:D in trusted zone.	No authentication in DMZ. All users look the same at Sterling Integrator or C:D in trusted zone.	Single Factor authentication in DMZ (user authentication only.) All users look the same at Sterling Integrator or C:D in trusted zone.

Authenticating SSP to the Trusted Zone Application

After SSP authenticates the remote trading partner, it creates another session to the application in the trusted zone. For this connection, SSP is the client and is authenticated by the trusted zone application. SSP provides SSL client authentication and user authentication.

SSL client authentication (recommended)—if you want to secure the session between SSP and the application in the trusted zone, you can require that SSP present a certificate during SSL client authentication. This certificate is authenticated by the trusted zone application during the SSL handshake. Use this option if you want to enforce the following security features:

- ◆ Secure the connection from SSP to the trusted zone application (recommended).
- ◆ You require multiple factors of authentication by the trusted zone application and will authenticate SSP, using SSL client and user authentication.
- ◆ You require a single factor of authentication by the trusted zone application, and you will authenticate SSP using SSL client authentication only.

User authentication—SSP is required to provide user credentials when logging on to the application in the trusted zone. The following are user authentication options:

- ◆ Pass-through (recommended)—this option sends the user credentials presented by the trading partner to the application in the trusted zone for authentication. This mechanism allows the user identity to be maintained at the trusted zone application.
- ◆ EA Mapped User Credentials—the user credentials are mapped using EA. When SSP uses EA for user authentication, it receives the user credentials from the trading partner and sends them to EA for validation. If configured, EA returns the mapped user credentials, and SSP uses them to log on to the application in the trusted zone.
- ◆ Netmap—the user credentials are defined in the outbound node of the netmap that is used by SSP to establish a session with the application, in the trusted zone. SSP logs in to the trusted zone application as the same user for all sessions. This method is not recommended.

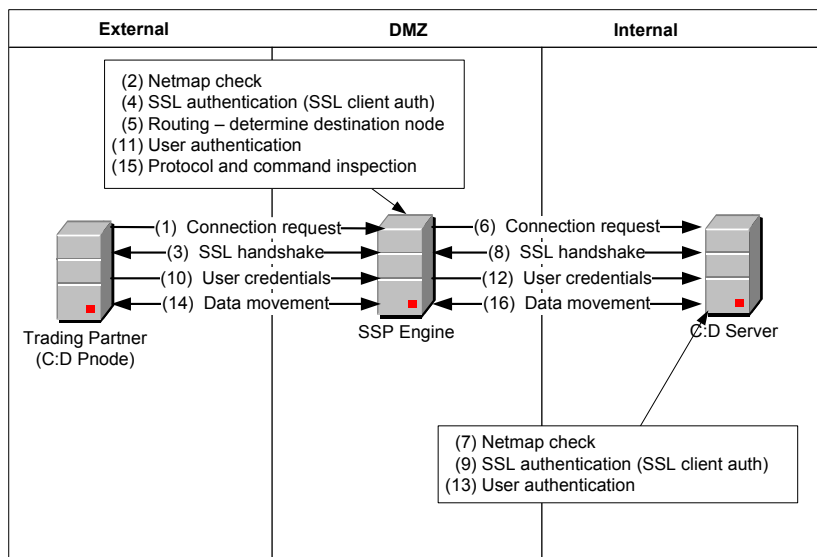
View a Finished SSP Configuration

This section provides diagrams of the SSP flow for the protocols.

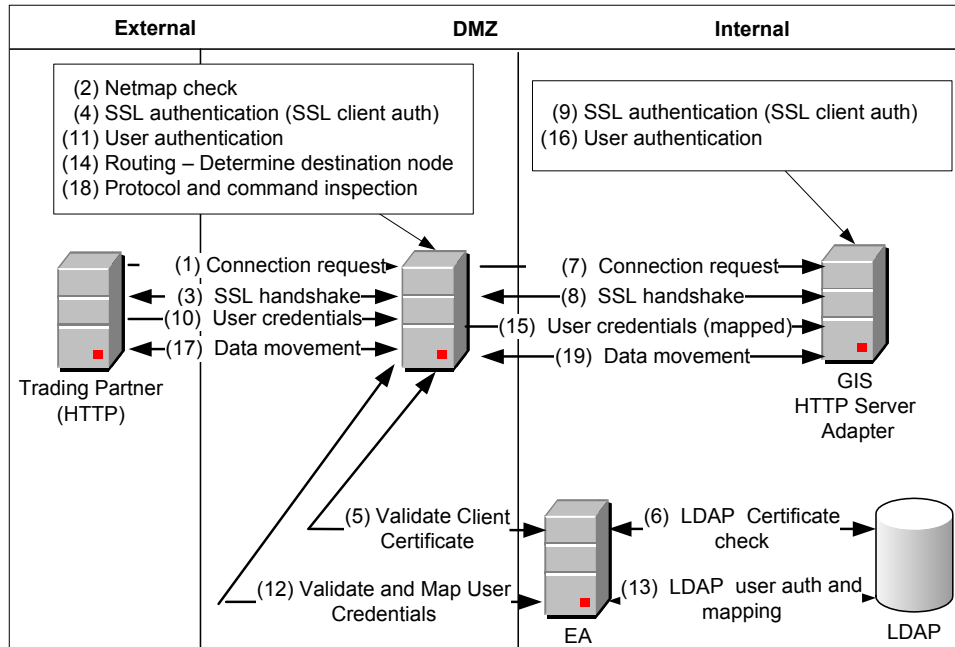
- Connect:Direct Reverse Proxy Diagrams
- Connect:Direct Forward Proxy Diagrams
- FTP Reverse Proxy
- HTTP Reverse Proxy
- SFTP Reverse Proxy

Connect:Direct Reverse Proxy Diagrams

The following illustration describes the Connect:Direct reverse proxy authentication steps:

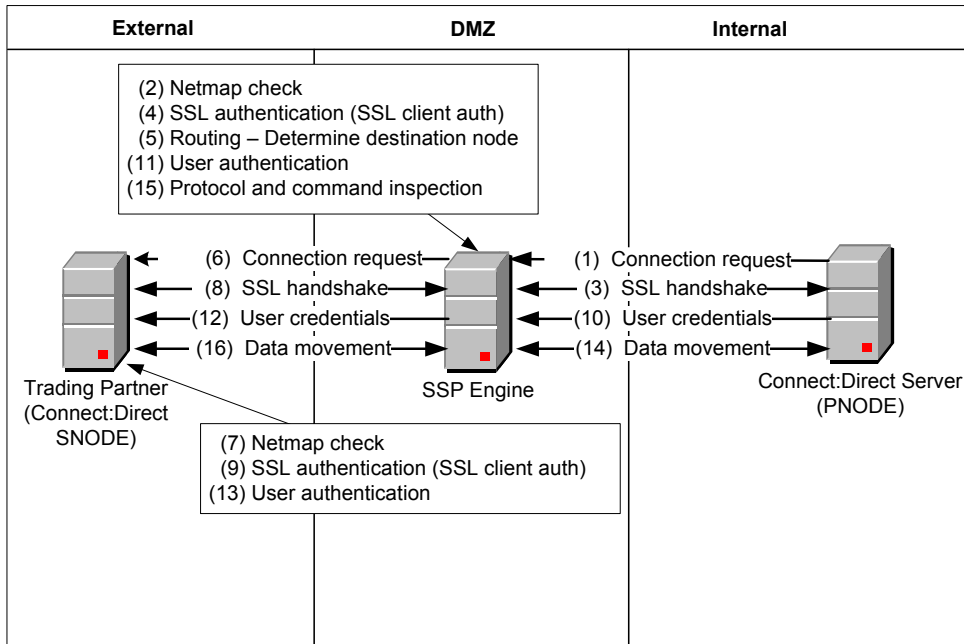


The following illustration describes a Connect:Direct reverse proxy authentication using EA:

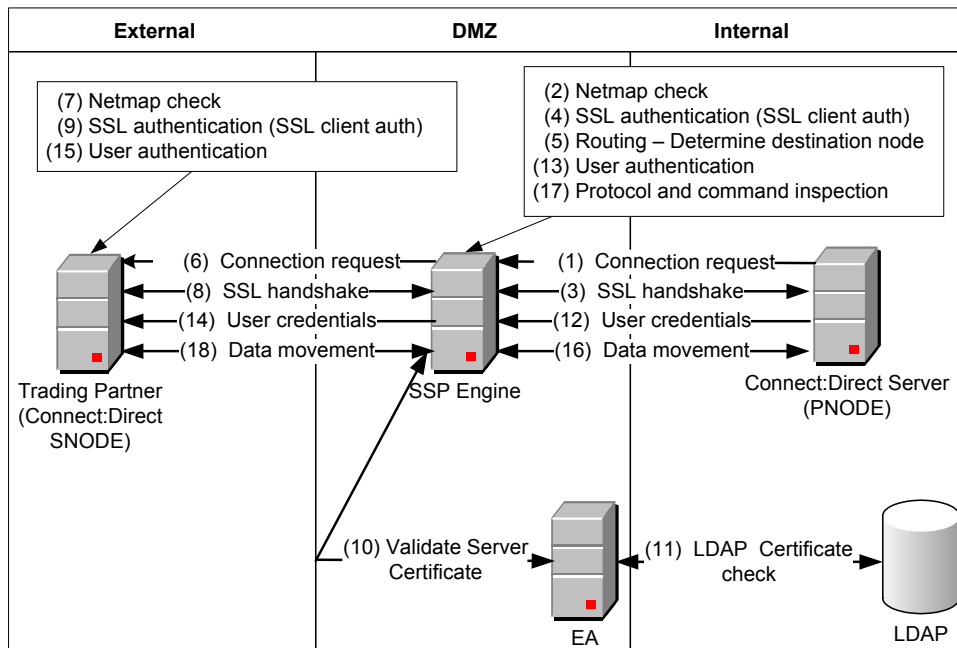


Connect:Direct Forward Proxy Diagrams

The following illustration describes the steps in a Connect:Direct forward proxy authentication:

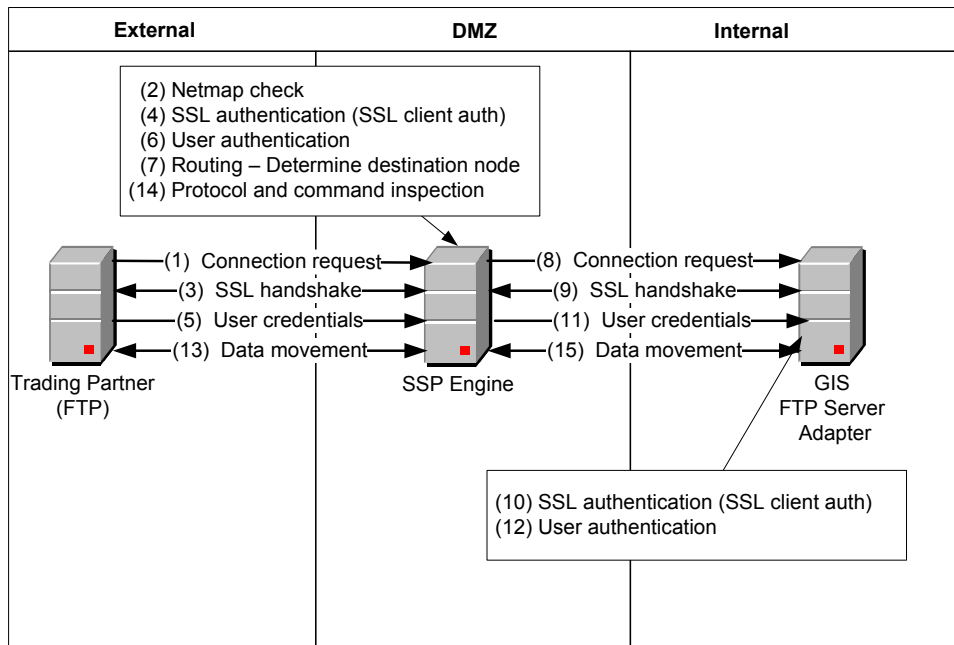


The following illustration describes the steps in a Connect:Direct forward proxy authentication using EA:

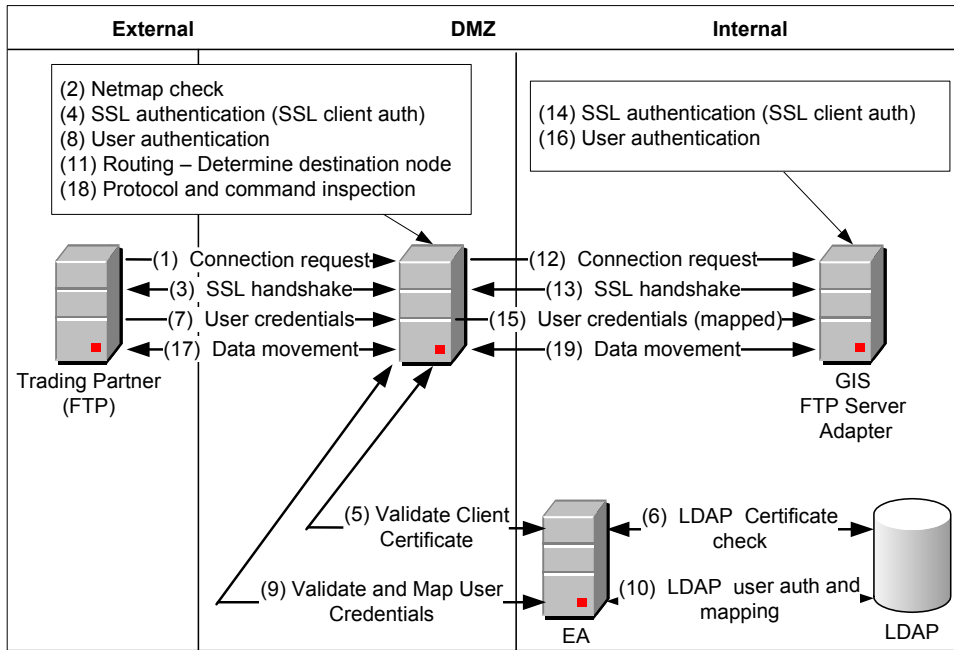


FTP Reverse Proxy

The following illustration describes the steps in an FTP reverse proxy authentication:

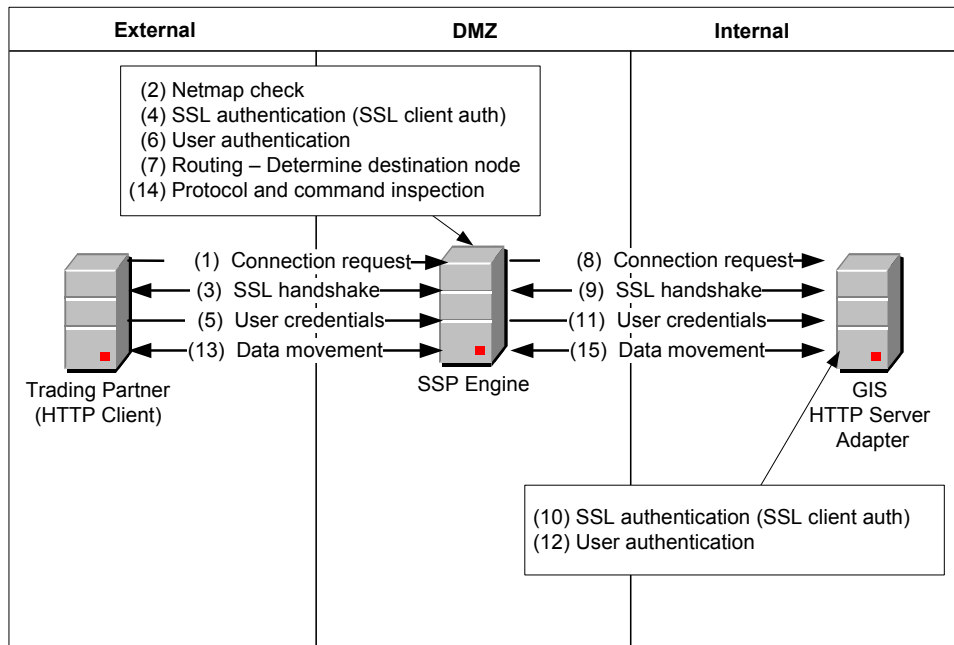


The following illustration describes the steps in an FTP reverse proxy authentication using EA:

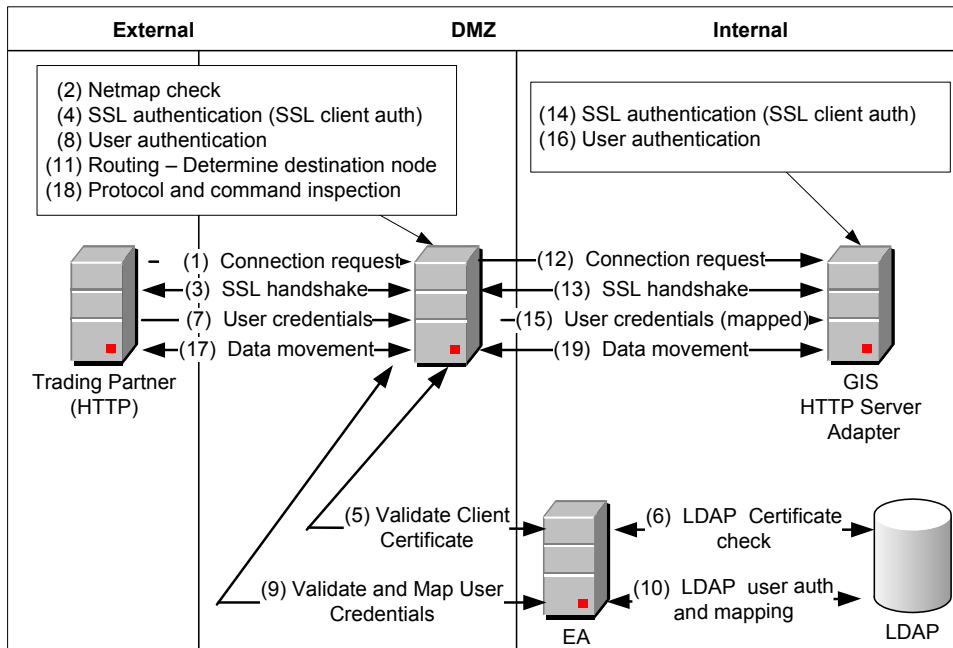


HTTP Reverse Proxy

The following illustration details the steps in an HTTP reverse proxy authentication:

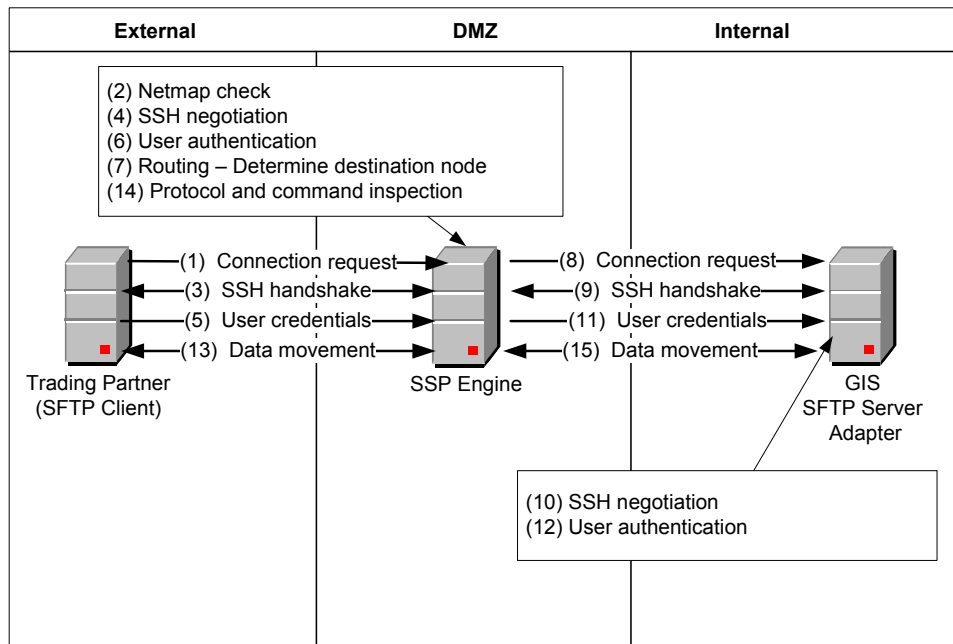


The following illustration details the steps of an HTTP reverse proxy authentication using EA:

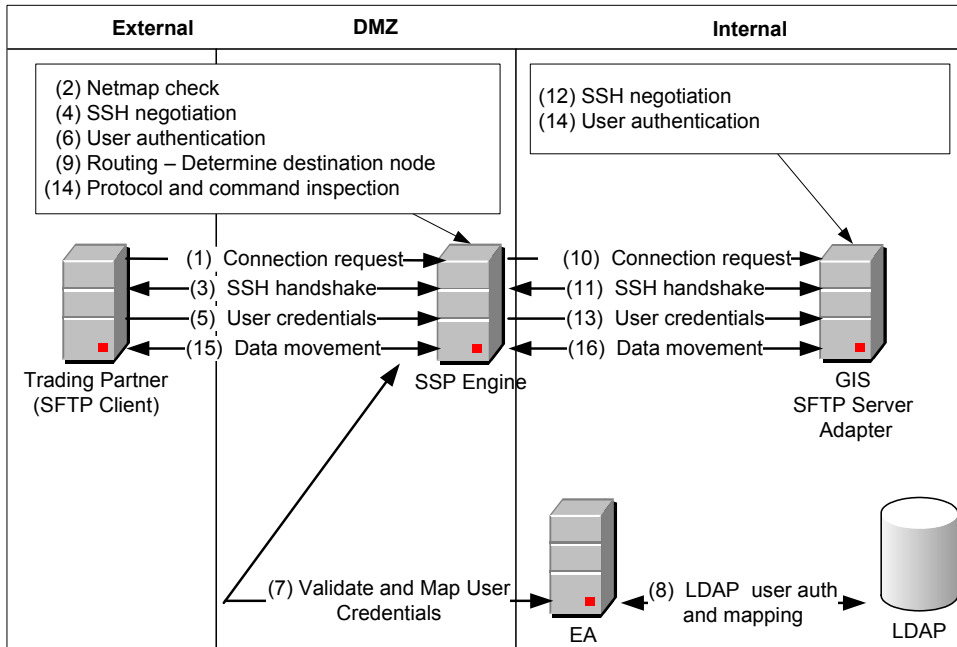


SFTP Reverse Proxy

The following illustration details the steps of an SFTP reverse proxy authentication:



The following illustration describes the steps in an SFTP reverse proxy authentication with EA:



Plan Your SSP Configuration

Before you are ready to configure SSP, plan how you will implement your proxy environment and determine what level of security is required to access the server in the trusted zone.

- Determine the Communications Protocol to Configure
- Identify Secure Session Requirements for a Connect:Direct, HTTP, or FTP Environment
- Identify Secure Session Requirements for an SSH (SFTP) Environment
- Determine Validation Requirements for Inbound Trading Partners (Inbound Nodes) to SSP
- Determine Connection Requirements for the Connection to the Sterling Integrator Server or Connect:Direct Node (Outbound Node)
- Set Up a Password Policy
- Set Up User Accounts to Configure the SSP Environment
- Set Up Users for Inbound Connections
- Set Up Sterling Integrator/Connect:Direct Servers (Outbound Node Servers) in the Trusted Zone
- Determine Security Requirements for Communications Sessions Between CM and the Engine
- Configure a Sterling External Authentication Server
- Configure a remote perimeter Server

Determine the Communications Protocol to Configure

SSP supports four protocols. Identify the protocol required for your environment, as defined below:

- Connect:Direct**—if you are using SSP to communicate between two Connect:Direct nodes or between a Connect:Direct node and an Sterling Integrator Connect:Direct server adapter, configure a Connect:Direct proxy adapter.
- FTP**—configure an FTP reverse proxy adapter if you are using SSP to communicate between an FTP client and Sterling Integrator.
- HTTP**—configure an HTTP reverse proxy adapter if you are using SSP to communicate between an HTTP client and Sterling Integrator.
- SFTP**—configure an SFTP reverse proxy adapter if you are using SSP to communicate between an SSH client and Sterling Integrator.

Follow the instructions in the scenario chapters to configure SSP for a protocol. If you plan to use more than one protocol, completely test and configure one protocol before adding a configuration for another protocol.

Identify Secure Session Requirements for a Connect:Direct, HTTP, or FTP Environment

If you are configuring a Connect:Direct, HTTP, or FTP environment, determine what secure communications sessions you will configure.

- Determine whether your environment requires a secure communications session between SSP and the inbound client and what security protocol is required for the session (SSL or TLS).

Determine whether your environment requires a secure communications session between SSP and the outbound server and what security protocol is required (SSL or TLS).

If you plan to use External Authentication for certificate or user authentication, determine whether your environment requires a secure communications session between SSP and EA and what security protocol is required for the session (SSL or TLS).

If a secure connection is required, do the following:

- ◆ Generate a self-signed certificate or obtain a CA certificate.
- ◆ For a server and EA certificate, obtain the certificate or the root certificate and import the root certificate into the trust store.
- ◆ Import the private key and certificate into the system certificate store.
- ◆ For client authentication, obtain the public certificate or root certificate of the inbound trading partner node and import it into the trusted certificate store.

Refer to *Manage Certificates for SSL/TLS Transactions with Trading Partners* on page 471 for instructions.

Identify Secure Session Requirements for an SSH (SFTP) Environment

If you are configuring an SFTP environment, determine what secure session options to configure. Authentication for an SFTP connection is performed with the exchange of session keys for the server and the client. To implement authentication for SFTP connections, you must create SSH key stores and import SSH keys into them. These key stores and keys can then be selected when you configure SFTP adapters.

Public key server authentication is mandatory. Therefore, you must configure both a local host key and a known host key.

To implement public key client authentication, you configure an authorized user key and a local user key.

Refer to *Manage SSH Keys for SFTP Transactions* on page 289 for instructions.

Determine Validation Requirements for Inbound Trading Partners (Inbound Nodes) to SSP

Determine security policy requirements for the inbound connection. Security options include:

Require no authentication.

Configure inbound node matching to allow only specific hosts to connect to SSP.

Validate the user ID by comparing it to information stored in the SSP local user store. Validate the certificate by comparing it to information in the SSP local certificate store.

Validate the trading partner (client) user ID and/or certificate using EA. Some of the validation methods that can be implemented using EA include:

- ◆ Query an LDAP or HTTP server to validate dates and signature on an inbound certificate
- ◆ Authenticate user Common Name (CN) specified in the certificate of the inbound node or group name with which the CN is associated

- ◆ Validate attributes of the certificate against information stored on LDAP server
- ◆ Validate certificate against a certificate revocation list (CRL) stored on LDAP or HTTP server
- ◆ Authenticate the user ID and password submitted as logon credentials for the target server by comparing them against information stored on an LDAP or TAMS server and authorize access

Determine Connection Requirements for the Connection to the Sterling Integrator Server or Connect:Direct Node (Outbound Node)

Identify requirements for connection to the secure outbound node. Possible requirements include:

Not requiring that the user ID and password be authenticated in SSP. The user ID and password provided by the trading partner is passed to the Sterling Integrator or Connect:Direct server for authentication.

Connecting to the Sterling Integrator or Connect:Direct server (outbound node) using a user ID and password stored in the SSP netmap configuration.

Connecting to the Sterling Integrator or Connect:Direct server (outbound node) using information accessed from EA. The EA server determines whether an alternate user ID and password mapped to the trading partner (client) user ID should be used to connect to the outbound Sterling Integrator or Connect:Direct server.

Set Up a Password Policy

You have the ability to identify security requirements for a group of users and then configure a password policy to define the security requirements. After you define a password policy, you can apply it to users who configure the SSP environment or to users who connect to the SSP engine and send files to a secure server.

Refer to *Manage User Accounts and Passwords* on page 255 for instructions on defining a password policy and associating it with a user.

Set Up User Accounts to Configure the SSP Environment

You must create user accounts for users who will access the SSP Configuration Manager tool to configure the SSP environment. You can create operator users who have read-only access or define administrator users who have full access to Configuration Manager.

Refer to *Manage User Accounts and Passwords* on page 255 for instructions on defining Configuration Manager users.

Set Up Users for Inbound Connections

Depending upon your configuration, you may need to create a user account for a trading partner who plans to connect to the SSP engine to transfer files to Sterling Integrator or Connect:Direct server to authenticate the user ID and password in the SSP local user store.

Refer to *Manage User Accounts and Passwords* on page 255 for instructions to define inbound users.

Set Up Sterling Integrator/Connect:Direct Servers (Outbound Node Servers) in the Trusted Zone

You set up servers in the trusted zone by performing the following tasks:

- Define users in Sterling Integrator for HTTP, FTP, and SFTP environments
- Define users at the Connect:Direct server for a Connect:Direct environment
- Make necessary configuration updates to the Sterling Integrator adapters
- Make necessary changes to the Connect:Direct netmaps
- Test the connection between SSP and the Sterling Integrator or Connect:Direct server

Determine Security Requirements for Communications Sessions Between CM and the Engine

When you install CM and an engine, a secure communications channel is required to communicate. By default, the SSL communication is configured using a single key for both the engine and the system where the web server and CM are installed.

To secure the communication between these components, replace the factory certificates. Refer to *Manage Certificates Between SSP Components* on page 275 for instructions.

Configure a Sterling External Authentication Server

An advanced method of user and certificate authentication is provided through an optional Sterling Commerce product called Sterling External Authentication Server. If you plan to use this tool to authenticate users or certificates, you must configure an EA server.

Refer to *Configure SSP for Sterling External Authentication Server (EA)* on page 273 for instructions on configuring an EA server.

Configure a remote perimeter Server

A local perimeter server (internal) is installed with SSP and will be used to manage communications. You can install a remote perimeter server if you want an additional perimeter server.

Refer to *Configure Perimeter Servers to Manage SSP Communications* on page 263 for sample implementations of the remote perimeter server and for instructions on configuring a remote perimeter server. Refer to *Install a Remote Perimeter Server* on page 73 for instructions on installing a remote perimeter server.

Install or Upgrade SSP on UNIX or Linux

Before you install SSP, review the system requirements. Confirm that your system meets all requirements. Follow the procedures to install or upgrade SSP.

Verify your installation by starting CM and the engine, and ensuring that they can communicate.

SSP Installation Checklist for UNIX or Linux

Use the following checklist to ensure that you complete all the tasks necessary to install SSP:

Installation Task	Procedure to Complete
Verify that your system meets the requirements specified for this release.	<i>System Requirements</i> on page 61.
Upgrade the engine, if you installed version 3.0 or later, or install SSP for the first time.	<i>Install or Upgrade the Engine on UNIX or Linux</i> on page 57.
If you install the engine on a computer with more than one Network Interface Card (NIC), specify the IP bind address of the NIC associated with that engine.	Change the IP Address for an Engine on page 248. Change the IP Address for an Engine. Click Manage SSP Engines on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Upgrade CM, if you installed version 3.0 or later, or install CM for the first time.	<i>Install or Upgrade CM on UNIX or Linux</i> on page 58.
Obtain a temporary license key and copy it to the appropriate directory.	<i>Obtain a License Key File for UNIX or Linux</i> on page 59
Start the engine.	Start the Engine on UNIX or Linux on page 325. Start the Engine on UNIX or Linux. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .

Installation Task	Procedure to Complete
Run CM.	Run CM on UNIX or Linux on page 327. Run CM on UNIX or Linux. Click Manage Engines on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Create an engine definition in CM.	Create an Engine Definition on page 59
Verify the engine and CM connection.	Manage SSP Engines on page 245. Manage SSP Engines. Click Manage SSP Engines on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Check in certificates for the connection between the engine and CM.	Manage Certificates Between SSP Components on page 275. Manage Certificates Between SSP Components. Click Manage Certificates Between SSP Components on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Determine if your environment requires a remote perimeter server.	Configure Perimeter Servers to Manage SSP Communications on page 263. Configure Perimeter Servers to Manage SSP Communications. Click Configure a Remote Perimeter Server on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
If required, install a remote perimeter server.	Install a Remote Perimeter Server on page 73

SSP Startup Worksheet for UNIX or Linux

Use the following worksheet to record the host name or IP address of CM and the engine, listening ports, and the URL for the CM sign-in screen. You refer to this information to use the application and set up your environment.

Note: When assigning ports, check that ports are not used by other software.

CM	Value at Installation
Host name or IP address	
CM listen port	
Web server listen port	

URL to Connect to CM

Engine	Value at Installation
Host name or IP address	
Listen port	

Install or Upgrade the Engine on UNIX or Linux

Use this procedure to install or upgrade the engine.

If you previously installed version 3.0 or later of the engine, you can upgrade to this version by installing over the existing files. If you upgrade the engine, the passphrases and port definitions are maintained as well as configuration and log files. All adapter definitions created in the previous version can be used in the new installation.

To install or upgrade an engine on UNIX or Linux:

1. Navigate to the directory where you downloaded the SSP installation file.

Refer to the following table to identify the file to install the engine on your operating system:

Hardware	File
IBM System p5 and IBM Power System	SSP.V3301.AIX.bin
HP Integrity system with Intel Itanium processor	SSP.V3301.HP-IA.bin
HP 9000 (PA-RISC)	SSP.V3301.HP.bin
x64/x86 Linux (32-bit)	SSP.V3301.Linux.bin
x64/x86 Linux (64-bit)	SSP.V3301.Linux_X64.bin
Sun SPARC system	SSP.V3301.SolarisSPARC.bin

Note: Log on to the UNIX system with the privileges required to install software.

2. Type the following command to retrieve the SSP engine, CM, and perimeter server installation files from the archive:

```
tar xvf SSP installation file
```

3. Type the name of the engine installation file for your platform and press **Enter**.
4. Read the terms of the license agreement. At the end of the agreement, type **Y** at the prompt.

5. For a new installation, perform the following steps:
 - a. When prompted for the installation location, press **Enter** to accept the default directory or specify a different directory.
 - b. Accept the default value **63366** for the engine listen port or specify a different port. Record the engine listen port on the Startup Worksheet, and press **Enter**.
 - c. Type a passphrase and press **Enter**. You need this passphrase in the future.
 - d. Retype the passphrase and press **Enter**.
6. For an upgrade, perform the following steps:
 - a. Specify the directory where the previous version is installed and press **Enter**. A prompt displays the directory and a message indicating that the Sterling Secure Proxy installation already exists.
 - b. Type **C** to continue.
7. Review the pre-installation summary, and press **Enter**.
8. Press **Enter**. The command prompt is displayed.

Install or Upgrade CM on UNIX or Linux

Use this procedure to install or upgrade CM.

If you previously installed version 3.0 or later of CM, you can upgrade to this version by installing over the existing files. After you upgrade, the passphrases and port definitions from the previous version are maintained as well as configuration and log files. All previously defined adapter definitions can be used in the new installation.

To install or upgrade CM on UNIX or Linux:

1. Navigate to the directory where you extracted the CM installation file from the archive in the previous procedure.

Refer to following table to identify the file to use to install CM on your operating system:

Hardware	File
IBM System p5 and IBM Power System	SSPcm.V3301.AIX.bin
HP Integrity system with Intel Itanium processor	SSPcm.V3301.HP-IA.bin
HP 9000 (PA-RISC)	SSPcm.V3301.HP.bin
x86 Linux (32-bit)	SSPcm.V3301.Linux.bin
x64 Linux (64-bit)	SSPcm.v3301.Linux_X64.bin
Sun SPARC system	SSPcm.V3301.SolarisSPARC.bin

Note: Log on to the UNIX system with the privileges required to install software.

2. Type the name of the CM installation file for your platform and press **Enter**.

3. Read the terms of the license agreement. At the end of the agreement, type **Y** at the prompt.
4. For a new installation, perform the following steps:
 - a. When prompted for the installation location, press **Enter** to accept the default directory or specify a different directory.
 - b. Accept the default value **62366** for the CM listen port, or specify a different port. Record the CM listen port on the Startup Worksheet, and press **Enter**.
 - c. Type a passphrase and press **Enter**. You need this passphrase in the future.
 - d. Retype the passphrase and press **Enter**.
 - e. Accept the default value of **8443** for the web server listen port or specify a different port. Record the web server listen port on the Startup Worksheet, and press **Enter**.
5. For an upgrade, perform the following steps:
 - a. Specify the directory where the previous version is installed and press **Enter**. A prompt displays the directory and a message indicating that the Sterling Secure Proxy installation already exists.
 - b. Type **C** to continue.
6. Review the pre-installation summary, and press **Enter**.
7. Press **Enter**. The command prompt is displayed.
8. If you previously configured a single sign on HTTP adapter, open property tag and you will find the url used for SSP3.2 was not removed. I was told it should be in SSO object.

Obtain a License Key File for UNIX or Linux

One license is required for each engine. You receive a temporary license key file after you order the product. The temporary key allows you to use SSP for a limited time. You receive a permanent license key after you provide information about the computer where the engine is installed.

If you upgrade SSP, you can use your existing license. You are not required to complete these procedures. Refer to the Sterling Secure Proxy License Key Guide for instructions to obtain a temporary key and a permanent key.

Create an Engine Definition

The engine resides in the DMZ and runs the proxy adapters that handle client communication between clients and servers in your trusted zone. The engine receives configuration information from CM. You create an engine definition using CM.

Before you configure the engine, gather the following information. After you create the engine definition, validate the configuration by ensuring that CM can view it.

CM Field	Feature	Value
Engine Name	Name of the engine	
Engine Host	IP address of the engine	
Engine Listen Port	Port number of the engine	

To define an engine:

1. Click **Configuration** from the menu bar.
2. Click **Actions > New Engine**.
3. Specify the following values:
 - ◆ Engine Name
 - ◆ Engine Host
 - ◆ Engine Listen Port
4. Click **Save**.

Verify that the engine is running. Click Manage SSP Engines on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm.

System Requirements

System requirements vary with business needs and your system environment. Factors include number of transactions processed, amount of data transferred, and running SSP with a perimeter servers. Review the requirements before you begin the installation tasks.

SSP UNIX and Linux System Requirements

This section identifies the system requirements for UNIX and Linux platforms. A JRE is installed with SSP. Configuration information is maintained on Configuration Manager (CM) and the engine. The space to store configuration files depends on the files you transmit and how long you maintain them, as well as the level of logging. The minimum space in the following table identifies the space required if you turn on debugging.

SSP UNIX and Linux Host System Requirements

SSP requires the following RAM and disk space requirements on a UNIX or Linux host system:

Component	File Descriptor Size	RAM Minimum	Disk Space Minimum
CM	N/A	512 MB	2 GB
Engine	N/A	1 GB	2 GB
Perimeter Server	2048 or greater (preferred setting: unlimited)	1 GB	2 GB

SSP UNIX or Linux Operating Systems Supported

SSP supports the following UNIX and Linux operating systems:

Hardware	Operating System
HP Integrity system with Intel Itanium processor	HP-UX, version 11.23 SSP supports 64-bit JRE with this operating system.
HP 9000 (PA-RISC)	HP-UX, version 11.23 SSP supports 64-bit JRE with this operating system.
IBM System p5 and IBM Power Systems	AIX 5L, version 5.3. SSP supports 64-bit JRE with this operating system.
x64/x86 64-bit	Red Hat Enterprise Linux Advanced Server, version 5 SuSE SLES, version 10 SSP supports 64-bit JRE with these operating systems.
x64/x86 32-bit	Red Hat Enterprise Linux Advanced Server, version 5 SuSE SLES, version 10
Sun SPARC system	Solaris, version 10 SSP supports 64-bit JRE with this operating system.
	VMware ESX and VMware vSphere with any UNIX or Linux operating system supported by SSP. Consider VMware configuration, administration, and tuning issues. Your VMware administrator must address these. Sterling Commerce does not provide advice regarding VMware-specific issues.
x86 (Intel VT-x and AMD-V) 32-bit and 64-bit	Kernel-based Virtual Machine (KVM) with Red Hat Enterprise Linux Advanced Server, version 5.4. Consider KVM configuration, administration, and tuning issues. Your Red Hat administrator must address these. Sterling Commerce does not provide advice regarding KVM-specific issues

Perimeter Server Requirements in UNIX or Linux

You can install and run a remote perimeter server (PS), on a different computer from CM or the engine. The PS supports the UNIX or Linux platforms supported by SSP.

Hardware Accelerator Board

SSP supports the cryptographic Sun Crypto Accelerator 10 hardware accelerator board.

Hardware Security Module (HSM) Requirements

SSP supports the following Hardware Security Module (HSM) appliance to store certificates:

- Safenet ProtectServer Gold
- Safenet ProtectServer External
- Thales nShield PCI

Thales netHSM

SSP Windows System Requirements

This section identifies system requirements for Windows platforms. A JRE is installed with SSP. Configuration information is maintained on both CM and the engine. How much is required to store configuration files depends on the size of the files and how long you maintain files, as well as the level of logging. The following table identifies the space required if you turn on debugging.

SSP Windows Host System Requirements

SSP requires the following minimum RAM and disk space requirements on a Windows system:

Component	RAM Minimum	Disk Space Minimum
CM	512 MB	2 GB
Engine	1 GB	2 GB
Perimeter Server	1 GB	2 GB

SSP-Supported Windows Operating Systems

SSP supports the following Windows operating systems:

Windows Server 2003 Enterprise Edition Service Pack 1 (32-bit)

Windows Server 2008 R2 (64-bit). SSP supports 64-bit JRE with this operating system.

VMware ESX and VMware vSphere with any Windows operating system supported by SSP. Consider VMware configuration, administration, and tuning issues. Your VMware administrator must address these. Sterling Commerce does not provide advice regarding VMware-specific issues.

Perimeter Server Requirements on Windows

You can run a remote perimeter server (PS) on a different computer from CM or the engine. The PS supports the following Windows platforms:

Windows 2003 Server Enterprise Edition R2 (32-bit)

Windows 2003 Server Standard Edition R2 (32-bit)

Windows Server 2008 R2 (64-bit)

Client Connections Supported

SSP is compatible with FTP, HTTP, or SSH-SFTP clients that comply with the relevant RFCs. The following clients have been tested and approved for interoperability with SSP:

Client	Protocol
Connect:Direct for z/OS (formerly OS/390) version 4.5 or later	Connect:Direct (SSL, TLS)

Client	Protocol
Connect:Direct for UNIX version 3.6.01 or later	Connect:Direct (SSL, TLS)
Connect:Direct for Windows version 4.2 or later	Connect:Direct (SSL, TLS)
Connect:Direct Select version 1.1 or later	Connect:Direct (SSL, TLS)
Connect:Direct for i5/OS version 3.6.00 or later	Connect:Direct (SSL, TLS)
Connect:Direct FTP+ version 1.1.08 or later	Connect:Direct (SSL, TLS)
Sterling Integrator version 4.1 or later	FTP (SSL, TLS) HTTP (SSL, TLS) SSH-SFTP Connect: Direct (SSL, TLS)
Sterling Secure Client	FTP (SSL, TLS) SSH-SFTP HTTP - WebDAV (SSL)
Connect:Express for z/OS (formerly OS/390) version 4.2.2 or later	PeSIT
Connect:Express for UNIX version 1.4.4 or later	PeSIT
Connect:Express for Windows version 3.0.5 or later	PeSIT
Internet Explorer 7 and 8	HTTP - myFileGateway
Firefox 3.5	HTTP - myFileGateway
Safari 3.2.3 and 4.0 on Windows and Mac OS X (10.5.7 and 10.6.0)	HTTP - myFileGateway
cURL 7.12.1 or later with openssl 0.9.7a or later	FTP (SSL, TLS) HTTP
OpenSSH 4.3p2 or later	SSH
WS_FTP Professional 2007 or later	FTP (SSL, TLS) SSH-SFTP

Web Browsers Supported by CM

Sterling Secure Proxy supports the following web browsers when using CM:

- Firefox 3.0 or later running on Windows
- Microsoft Internet Explorer 7

Server Connections Supported

Sterling Secure Proxy supports the following server connections:

- Connect:Direct for z/OS (formerly OS/390) version 4.5.00 or later
- Connect:Direct for UNIX version 3.6.01 or later
- Connect:Direct for Windows version 4.2 or later
- Connect:Direct for i5/OS version 3.6.00 or later
- Connect:Direct Select version 1.1 or later

Gentran Integration Suite (GIS) version 4.3.21 or later
 Sterling Integrator version 5.0.03 or later
 Sterling File Gateway (SFG) version 1.1 with GIS version 4.3.22 or later (4.3.x)
 Sterling File Gateway (SFG) version 2.0 with Sterling Integrator version 5.0.03 or later

Sterling Security Products Supported

SSP supports the following Sterling security products:

Sterling Certificate Wizard 1.2.03 or later
 Sterling External Authentication Server 2.3.00 or later

Cipher Suites Supported

SSP supports the following cipher suites for the Connect:Direct, FTP, and HTTP protocols:

TLS_RSA_WITH_RC4_128_SHA
 TLS_RSA_WITH_RC4_128_MD5
 TLS_RSA_WITH_DES_CBC_SHA
 TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
 TLS_RSA_EXPORT_WITH_RC4_40_MD5
 TLS_RSA_WITH_NULL_MD5
 TLS_RSA_WITH_AES_256_CBC_SHA
 TLS_RSA_WITH_AES_128_CBC_SHA
 TLS_RSA_WITH_3DES_EDE_CBC_SHA

Review Resources for UNIX or Linux

Before installation, review any network and security-specific configuration details relevant for the hardware used to install CM and the engine. Consider details that are specific to your environment.

Refer to the following list of resources as you plan the use of network and security-related resources to install and configure SSP:

Configuration Resource	SSP Usage
TCP ports	Use available port numbers, in appropriate port ranges. The following SSP components require listening ports: <ul style="list-style-type: none"> ◆ CM ◆ Jetty web server ◆ Engine
Internet Explorer or Firefox	Access the CM logon screen from Internet Explorer or Firefox.

Configuration Resource	SSP Usage
CM	Install CM in the trusted company zone. You can set up multiple engines with the same CM, but only one CM can be set up to control an engine. CM port handles listen requests from the Jetty web server. The default port number is 62366.
Jetty web server	The Jetty web server is installed when you install CM, and handles listen requests from the web browser. The web server port number is an element specified in the address bar when connecting to the logon screen. The default port number for the Jetty web server is 8443.
SSP engine	<p>The engine operates during production, and routes traffic.</p> <p>Install an engine in the DMZ. The default port number is 63366.</p> <p>If you install the engine on a computer with more than one Network Interface Card (NIC), specify the IP bind address of the card associated with that engine.</p> <p>Each engine requires a different license key and engine definition.</p> <p>When you define an engine in CM, you identify either the host name or the IP address in the definition. Create only one definition for each engine you install.</p>
Perimeter server	A local perimeter server is installed when you install the engine. It manages communications between the engine and other nodes. You can install a remote perimeter server separately on another computer.
Sterling External Authentication Server	To provide another level of security by authenticating users or certificates, or mapping users, install Sterling External Authentication Server (EA). For more information, refer to the Sterling External Authentication Server documentation library.
Default certificates	To secure communication, SSP is configured with default certificates that are exchanged between CM and the engine. Replace these certificates with your own after installation. Refer to Manage Certificates Between SSP Components at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm . Refer to <i>Manage Certificates Between SSP Components</i> on page 275.

Install or Upgrade SSP on Windows

Before you install SSP, review the system requirements. Confirm that your system meets all requirements. Follow the procedures to install or upgrade SSP. Verify your installation by starting CM and the engine, and ensuring that they can communicate.

SSP Installation Checklist for Windows

Installing SSP requires you to complete several tasks. Use the following checklist to ensure that you complete all the tasks necessary for an installation:

Installation Task	Procedure to Complete
Verify your system meets the hardware and software requirements specified for this release.	<i>System Requirements</i> on page 61
Upgrade the engine, if you installed version 3.0 or later, or install SSP for the first time.	<i>Install or Upgrade the Engine on Windows</i> on page 69
If you install the engine on a computer with more than one NIC, specify the IP bind address of the NIC associated with that engine.	Change the IP Address for an Engine. Click Manage SSP Engines on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Upgrade CM, if you installed version 3.0 or later, or install CM for the first time.	<i>Install or Upgrade CM on Windows</i> on page 69.
Obtain a temporary license key and copy it to the appropriate directory.	<i>Sterling Commerce License Key Guide</i>
Request and install a permanent license key.	<i>Sterling Commerce License Key Guide</i>
Start the engine and CM.	Start the Engine on Windows. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .

Installation Task	Procedure to Complete
Log onto CM.	Start the CM on Windows. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Create an engine definition in CM.	<i>Create an Engine Definition</i> on page 70.
Verify the engine and CM connection.	View the Engine and CM on Windows. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Check in certificates for the connection between the engine and CM	Manage Certificates Between SSP Components. Click Manage Certificates Between SSP Components on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Determine if your environment requires a remote perimeter server.	Configure Perimeter Servers to Manage SSP Communications. Click Configure a Remote Perimeter Server on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
If required, install a remote perimeter server.	<i>Install a Remote Perimeter Server</i> on page 73.

SSP Startup Worksheet for Windows

Use the worksheet to record the host name or IP address of CM and the engine, listen ports, and the URL for the CM log in screen. You refer to this information when you use the application and set up your environment. If you change this information, use this worksheet to record your changes.

Note: When assigning ports, check that ports are not used by other software.

CM	Defined at Installation	New
Host name or IP address of CM		
CM listen port		
Web server listen port		
URL to Connect to CM		

Engine	Defined at Installation	New
Host name or IP address of the engine		
Engine listen port		

Install or Upgrade the Engine on Windows

Use this procedure to install or upgrade the engine.

If you installed version 3.0 or later, you can upgrade to this version by installing over the existing files. If you install this version in the same directory, the passphrases and port definitions are maintained as well as configuration and log files. All adapter definitions created in the previous version can be used in the new installation.

At installation, define a passphrase for CM and the engine, to ensure that files are secure. A passphrase is six or more characters and contains any characters. The passphrase for CM is independent of the engine passphrase. To start CM or the engine, type the passphrase. You type the passphrase at shutdown.

To install or upgrade an engine on Windows:

1. Navigate to the directory where you downloaded the SSP installation file for Windows.
2. Double-click the SSP.V3301.Windows.zip file to extract the SSP engine, CM, and perimeter server installation files for Windows.
3. Take one of the following actions:
 - ◆ To install the engine on Windows Server 2003 (32-bit), double-click **SSP.V3301.Win.exe**.
 - ◆ To install the engine on Windows Server 2008 (64-bit), double-click **SSP.V3301.Win_X64.exe**.
4. After the introduction, click **Next**.
5. Scroll down in the license agreement and read the agreement. Click the radio button to accept the terms and click **Next**.
6. Accept the default directory, or click **Choose** to navigate to a different directory. For an upgrade, change to the directory where the previous version is installed. Click **Next**.
7. To continue a new installation:
 - a. Accept the default value **63366** for the engine listen port or specify a different port. Record the engine listen port on the Startup Worksheet and click **Next**.
 - b. Type a passphrase. Retype the passphrase and click **Next**.
8. To continue an upgrade, specify the directory where the previous version is installed and press **Next**. A prompt indicates the Sterling Secure Proxy installation already exists.
9. Review the pre-installation summary. Click **Install**.
10. At the Installation Complete screen, click **Done**.

Install or Upgrade CM on Windows

Use this procedure to install or upgrade CM.

If you previously installed version 3.0 or later of CM, you can upgrade to this version by installing over the existing files. If you install this version in the same directory, the passphrases and ports are maintained as well as configuration, log files, and adapter definitions.

To install or upgrade CM on Windows:

1. Navigate to the directory where you extracted the CM installation files from the archive in the previous procedure.
2. Take one of the following actions:
 - ◆ To install the CM on Windows Server 2003 (32-bit), double-click **SSPcm.V3301.Win.exe**.
 - ◆ To install the CM on Windows Server 2008 (64-bit), double-click **SSPcm.V3301.Win_X64.exe**.
3. After the introduction, click **Next**.
4. At the end of the license agreement, click the radio button to accept the terms and click **Next**.
5. Accept the default directory, or click **Choose** to navigate to a different directory. For an upgrade, change to the directory where the previous version is installed. Click **Next**.
6. Perform the following steps to continue a new installation:
 - a. Accept the default value **62366** for the CM listen port or specify a different port. Record the CM listen port on the Startup Worksheet. Click **Next**.
 - b. Type a passphrase. Retype the passphrase and click **Next**.
 - c. Accept the default value of **8443** for the web server listen port or specify a different port. Record the web server listen port on the Startup Worksheet. Click **Next**.
7. To continue an upgrade, specify the directory where the previous version is installed and press **Next**. A prompt displays the directory and a message indicating that the Sterling Secure Proxy installation already exists.
8. Review the pre-installation summary before continuing. Click **Install**.
9. At the Installation Complete screen, click **Done**.

Obtain and Install a License Key File on Windows

One license is required for each engine. You receive a temporary license key file in an e-mail after you order the product. The temporary key allows you to use SSP for a limited time. You receive a permanent license key after you provide information about the computer where the engine is installed.

If you upgrade SSP, you can continue to use your existing license. You are not required to complete these procedures. Refer to the Sterling Secure Proxy License Key Guide.

Create an Engine Definition

The engine resides in the DMZ and runs the proxy adapters that manage client communication requests to servers in your trusted zone. To perform this function, the engine receives configuration information from CM. Use CM to create an engine definition that contains configuration information for the engine.

Before you configure the engine, gather the following information that you will need to configure the engine. After you configure the engine, validate the configuration by ensuring that CM can view the engine.

CM Field	Feature	Value
Engine Name	Name of the engine	
Engine Host	IP address of the engine	
Engine Listen Port	Port number of the engine	

To define an engine:

1. If necessary, select Configuration from the menu bar.
2. Click Actions > New Engine.
3. Specify the following values:
 - ◆ Engine Name
 - ◆ Engine Host
 - ◆ Engine Listen Port
4. Click Save.
5. Verify that the engine is running. Refer to *Manage SSP Engines* on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm for instructions.

Install a Remote Perimeter Server

SSP uses perimeter servers to increase security between internal and external communications. A local perimeter server (internal) is installed with SSP. The local mode server is useful in environments that do not require a DMZ solution.

To configure your environment so that your firewall only allows connections established from inside a more secure environment, install a remote perimeter server in a DMZ. You configure the remote perimeter servers within SSP. After you install and configure a remote perimeter server, you map how the perimeter server is used: inbound, outbound, or External Authentication. For more information, refer to *Configure a Remote Perimeter Server* on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm.

Perimeter Server Installation Prerequisites

Prior to installing and configuring a perimeter server on a remote system, you must complete the following tasks and gather the required information:

- Install CM and the engine.

- Go to the ESD download directory that contains the PS installer files.

- Obtain the IP address for both the remote perimeter server computer and the engine computer.

- If you plan to install the perimeter server in a less secure network zone than the SSP engine, open the port for connections from the engine to the remote perimeter server computer on which you plan to install your perimeter server.

- If you plan to install the perimeter server in a more secure network zone than the SSP engine, open the port for connections from the remote perimeter server computer on which you plan to install your perimeter server to the engine.

Perimeter Server Installation Guidelines

When you install a perimeter server, follow these guidelines:

Each perimeter server is limited to two TCP/IP addresses: internal interface and external interface. Internal interface is the TCP/IP address that the perimeter server uses to communicate with the engine. External interface is the TCP/IP address that the perimeter server uses to communicate with trading partners.

To use additional TCP/IP addresses, install additional perimeter servers.

To install an additional perimeter server on a computer with an existing instance, install the new perimeter server in unique installation directory.

To upgrade an existing perimeter server, install a new instance of perimeter server in the installation directory of the existing perimeter server.

The combination of internal TCP/IP address and port must be unique for all perimeter servers installed on one computer.

- ◆ If a perimeter server is installed using the wildcard address, then all ports must be unique.
- ◆ If a perimeter server is installed using the wildcard address, then its port is not available for use by service adapters that use the server or any other perimeter server on that computer.
- ◆ The internal and external interface may use the same TCP/IP address. However, the port used by the perimeter server is not available to the service adapters that use the server.

Install remote perimeter Server in a More Secure Network on UNIX or Linux

To install a perimeter server in a more secure network than your Sterling Secure Proxy server:

1. Navigate to the directory containing the downloaded perimeter server installer file for your platform.

Platform	Installation File Name
IBM System p5 and IBM Power system	PS.V3301.AIX.bin
HP Integrity system with Intel Itanium processor	PS.V3301.HP-IA.bin
HP 9000 (PA-RISC)	PS.V3301.HP.bin
x64/x86 Linux (32-bit)	PS.V3301.Linux.bin
x64/x86 Linux (64-bit)	PS.V3301.Linux_X64.bin
Sun SPARC system	PS.V3301.SolarisSPARC.bin

2. Copy the perimeter server installation file for your platform to your home directory or base directory. If you are using FTP to copy the file, make sure your session is set to binary mode.
3. To begin the installation, type the installation file name and press Enter.
The installation program displays the Introduction screen.
4. Press Enter to continue the installation.
If you type quit, the installation program will terminate.

5. Read the License Agreement information. Press Enter to page through the license agreement. When you are prompted to accept the License Agreement, press Y or y. The Choose Installation Folder screen is displayed.
6. Press Enter to accept the default installation folder, or type the absolute path name of the directory where the perimeter server will be installed.
7. Confirm that the installation directory is correct by typing Y or y. The Network Zone screen is displayed.
8. Type 2 to install the perimeter server in a more secure network. The installation program displays a list of network interfaces available on the perimeter server host.
9. Select the network interface for the perimeter server to use to communicate with the SSP engine, or press Enter if a specific interface address is not required.
10. Type the port number of the local port the perimeter server will use to communicate with the SSP engine. Specify a port number greater than or equal to 1024. If a specific port is not required, press Enter.
The installation program displays a list of network interfaces available on the perimeter server host.
11. Select the network interface for the perimeter server to use to communicate with the backend server, or press Enter if a specific interface address is not required.
12. Type the hostname or IP address of the SSP engine host that will be connected to this perimeter server.
13. Type the port number the SSP engine will listen on for requests from the perimeter server.
14. Verify the Post-Installation Summary information, and press Enter.
When the perimeter server is installed, the installation program displays an Installation Complete message.
15. Press Enter to exit the installation.
16. Change to the installation directory.
17. Type `startupPS.sh` to start the perimeter server.

Install a remote perimeter Server in a Less Secure Network on UNIX or Linux

To install a perimeter server in a less secure network than your Sterling Secure Proxy server:

1. Navigate to the directory containing the downloaded perimeter server installer file for your platform.

Platform	Installation File Name
IBM System p5 and IBM Power System	PS.V3301.AIX.bin
HP Integrity system with Intel Itanium processor	PS.V3301.HP-IA.bin
HP 9000 (PA-RISC)	PS.V3301.HP.bin

Platform	Installation File Name
x64/x86 Linux (32-bit)	PS.V3301.Linux.bin
x64/x86 Linux (64-bit)	PS.V3301.Linux_X64.bin
Sun SPARC system	PS.V3301.SolarisSPARC.bin

2. Copy the perimeter server installation file for your platform to your home directory or base directory. If you are using FTP to copy the file, make sure your session is set to binary mode.
3. To begin the installation, type the installation file name and press Enter.
The installation program displays the Introduction screen.
4. Press Enter to continue the installation.
If you type quit, the installation program will terminate.
5. Read the License Agreement information. Press Enter to page through the license agreement. When you are prompted to accept the License Agreement, press Y or y. The Choose Installation Folder screen is displayed.
6. Press Enter to accept the default installation folder, or type the absolute path name of the directory where the perimeter server will be installed.
7. Confirm that the installation directory is correct by typing Y or y. The Network Zone screen is displayed.
8. Type 1 to install the perimeter server in a less secure network. The installation program displays a list of network interfaces available on the perimeter server host.
9. Select the network interface for the perimeter server to use to communicate with the SSP engine, or press Enter if a specific interface address is not required.
10. Type the port number of the local port the perimeter server will listen on for requests from the SSP engine. Specify a port number greater than or equal to 1024. Press Enter. The installation program displays a list of network interfaces available on the perimeter server host.
11. Select the network interface for the perimeter server to use to communicate with trading partners, or press Enter if a specific interface address is not required.
12. Verify the Post-Installation Summary information, and press Enter.
When the perimeter server is installed, the installation program displays an Installation Complete message.
13. Press Enter to exit the installation.
14. Change to the installation directory.
15. Type startupPS.sh to start the perimeter server.

Install remote perimeter Server in a More Secure Network in Windows

To install a perimeter server in a more secure network in a Windows environment:

1. Navigate to the directory containing the downloaded perimeter server installer file for your platform.

Platform	Installation File Name
Windows Server 2003 Enterprise Edition Service Pack 1 (32-bit)	PS.V3301.Win.exe
Windows Server 2008 R2 (64-bit)	PS.V3301.Win_X64.exe

2. Copy the perimeter server installation file for your platform to the Windows server. If you are using FTP to copy the file, be sure your session is set to binary mode.
3. To begin the installation, run the perimeter server installation .exe file.
The installation program displays the Introduction screen.
4. Click Next to continue the installation.
5. Read the License Agreement information, accept the terms of the License Agreement, and click Next. The Choose Installation Folder screen is displayed.
6. Click Next to accept the default installation folder, click Choose to navigate to an installation folder, or type the absolute path name of the directory where the perimeter server will be installed and click Next.
The Network Zone screen is displayed.
7. Select the Perimeter Server in a more-secure zone button, and click Next. The installation program displays a list of network interfaces available on the perimeter server host.
8. Select the network interface for the perimeter server to use to communicate with the SSP engine, or click Next if a specific interface address is not required.
9. Type the port number of the local port the perimeter server will use to communicate with the SSP engine. Specify a port number greater than or equal to 1024. If a specific port is not required, click Next.
The installation program displays a list of network interfaces available on the perimeter server host.
10. Select the network interface for the perimeter server to use to communicate with the backend server, or click Next if a specific interface address is not required.
11. Type the hostname or IP address of the SSP engine host that will be connected to this perimeter server.
12. Type the port number the SSP engine will listen on for requests from the perimeter server, and click Next.
13. Verify the Pre-Installation Summary, and click Next.

When the perimeter server is installed, the installation program displays an Installation Complete message.

14. Click Done to exit the installation.
15. Change to the installation directory.
16. Do one of the following:
 - ◆ Run startPSService.cmd to start the perimeter server.
 - ◆ Configure the perimeter server service to start automatically as a Windows Service at system startup.

Install remote perimeter Server in a Less Secure Network in Windows

To install a perimeter server in a less secure network in a Windows environment:

1. Navigate to the directory containing the downloaded perimeter server installer file for your platform.

Platform	Installation File Name
Windows Server 2003 Enterprise Edition Service Pack 1 (32-bit)	PS.V3301.Win.exe
Windows Server 2008 R2 (64-bit)	PS.V3301.Win_X64.exe

2. Copy the perimeter server installation file for your platform to the Windows server. If you are using FTP to copy the file, be sure your session is set to binary mode.
3. To begin the installation, run the perimeter server installation .exe file.
The installation program displays the Introduction screen.
4. Click Next to continue the installation.
5. Read the License Agreement information, accept the terms of the License Agreement, and click Next. The Choose Installation Folder screen is displayed.
6. Click Next to accept the default installation folder, click Choose to navigate to an installation folder, or type the absolute path name of the directory where the perimeter server will be installed and click Next.
The Network Zone screen is displayed.
7. Select the Perimeter Server in a less-secure zone button, and click Next. The installation program displays a list of network interfaces available on the perimeter server host.
8. Select the network interface for the perimeter server to use to communicate with the SSP engine, or click Next if a specific interface address is not required.
9. Type the port number of the local port the perimeter server will listen on for requests from the SSP engine. Specify a port number greater than or equal to 1024. Click Next.

The installation program displays a list of network interfaces available on the perimeter server host.

10. Select the network interface for the perimeter server to use to communicate with trading partners, or click Next if a specific interface address is not required.
11. Verify the Pre-Installation Summary information, and click Next.

When the perimeter server is installed, the installation program displays an Installation Complete message.

12. Click Done to exit the installation.
13. Change to the installation directory.
14. Do one of the following:
 - ◆ Run startPSService.cmd to start the perimeter server.
 - ◆ Configure the perimeter server service to start automatically as a Windows Service at system startup.

Upgrade Perimeter Server in Windows, UNIX, or Linux

To upgrade an existing instance of perimeter server:

1. Run the perimeter server installation program.
2. Read and accept the License Agreement.
3. On the Installation Folder screen, select the installation directory of the existing perimeter server.

The installation program detects the existing perimeter server installation and displays an update message.

4. Select the option to update the existing installation.

The installation program will use the configuration information of the existing installation to configure the updated installation.

5. Verify the Pre-Installation Summary information and complete the installation.
6. Start the perimeter server.

Restrict the Policy for a remote perimeter Server

To limit perimeter server activity:

1. Install a remote perimeter server. Select the option to indicate that the perimeter server is in a more-secure network zone.

2. Edit the `restricted.policy` file located in the installation directory. The following is a sample `restricted.policy` file.

```
// Standard extensions get all permissions by default
grant codeBase "file:${java.ext.dirs}/*" {
  permission java.security.AllPermission;
};

grant {
  // Grant all permissions needed for basic operation.
  permission java.util.PropertyPermission "*", "read";
  permission java.security.SecurityPermission "putProviderProperty.*";
  permission java.io.FilePermission "-", "read,write";
  permission java.io.FilePermission ".", "read";
  // Needed to allow lookup of network interfaces.
  permission java.net.SocketPermission "*", "resolve";
};

grant {
  // Adjust for your local network requirements.
  // Needed to connect out for the persistent connection
  permission java.net.SocketPermission "localhost: ", "connect";
  // For each target FTP Server
  //
  // permission java.net.SocketPermission "ftphost: ", "connect";
  // Control connection.
  // permission java.net.SocketPermission"
  ftphost:lowPort-highPort", "connect"; // Passive data connections.
  // For each target HTTP Server//
  //
  permission java.net.SocketPermission "httphost: ", "connect";
  // For each target C:D snode
  //
  // permission java.net.SocketPermission "snode: ", "connect";
};
```

Edit the `grant` section, highlighted above, to define your local network requirements. Add a permission line for each back-end server Sterling Secure Proxy server can access.

Commented examples are provided for each type of back-end server that Sterling Secure Proxy supports.

Note: Do not edit the `grant` sections called `grant codeBase` or `Grant all permissions needed for basic operations`.

3. To turn on restrictions in a UNIX installation, edit the `remote_perimeter.properties` file located in the perimeter server installation directory. Set the value of `restricted` to `true` as shown below:

```
restricted=true
```

Restrictions will take effect the next time you start this perimeter server.

4. To turn on and activate restrictions in Windows:
 - a. Edit the installPS.cmd file located in the perimeter server installation directory. Remove the comment markers from the following line:

```
rem set POLICY="-Djava.security.manager -Djava.security.policy==restricted.policy"
```

- b. Run stopPSService.cmd to stop the current perimeter server.
- c. Run uninstallPSService.cmd to uninstall the existing perimeter server Windows service.
- d. Run installPS.cmd to install the modified version of the Windows service.
- e. Run startPSService.cmd to run the restricted perimeter server.

Note: If the perimeter server attempts to access restricted network resources, the connection is rejected and logged in the perimeter server log.

Upgrade SSP from Version 2.0.x to Version 3.x

Use the procedures in this section to upgrade SSP from version 2.0, 2.0.01 or 2.0.02 to version 3.x. To upgrade from version 3.0, follow the installation instructions.

SSP version 3.x uses a different architecture from version 2.0.x. It allows you to configure your environment using the Configuration Manager (CM). It then moves the configuration information to an Engine, to use during production. SSP version 2.0.x does not use an Engine or CM. Configuration and production occur on an SSP node, and data is stored in a database.

Before upgrading your environment, identify the configuration of your SSP version 2.0.x. Then, complete the procedures identified for each configuration. Configurations include:

Single SSP environment—If you installed SSP on one node, refer to *Upgrade a Single SSP Node* on page 84.

Clustered SSP environment—If you installed SSP on two or more nodes and all nodes use the same configuration information to provide high availability and secondary engines accept incoming requests if the primary engine is not available, refer to *Upgrade SSP Clustered Nodes* on page 88.

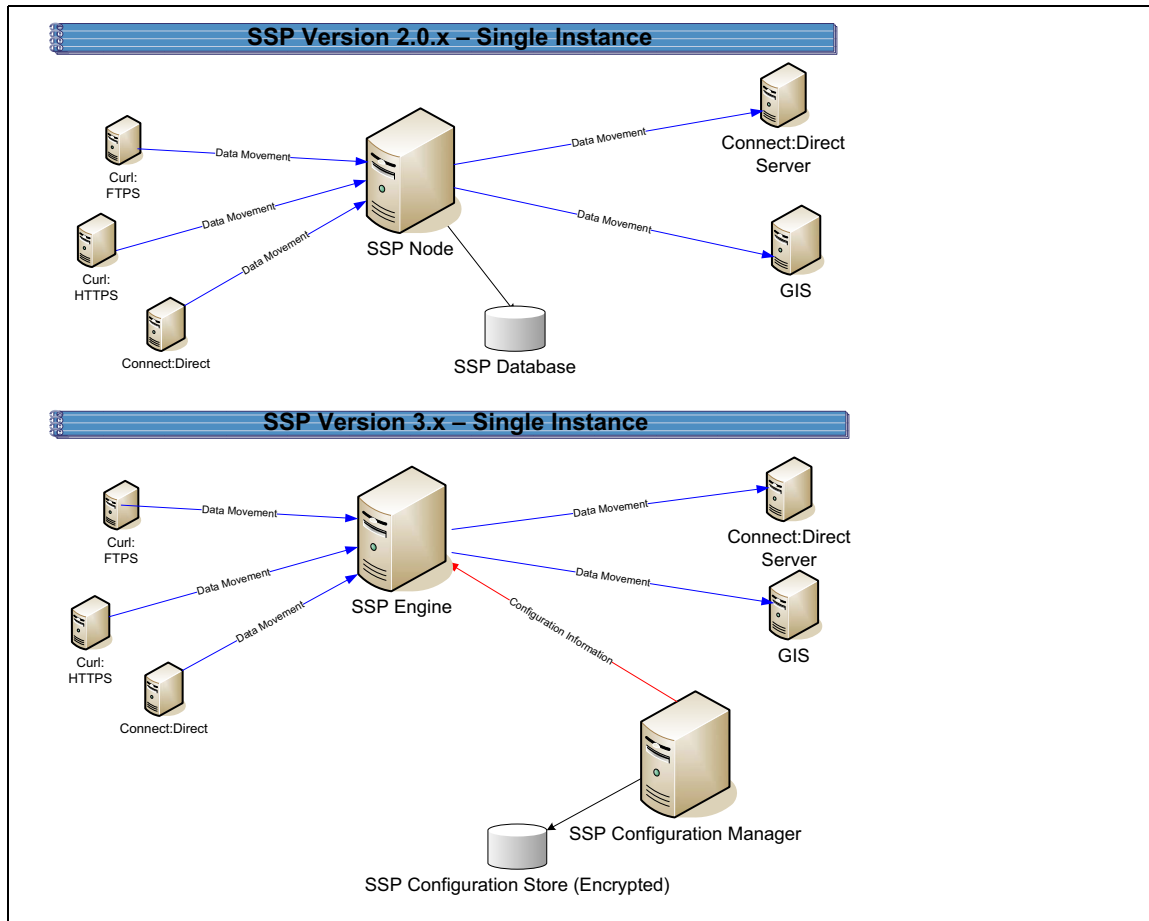
Load balancing SSP environment—If you installed SSP version 2.0.x on two or more nodes and created a load balancing environment to provide redundancy and share the workload among multiple servers, refer to *Upgrade an SSP Loading Balancing Environment* on page 93 for instructions on how to upgrade this environment.

Multiple SSP nodes environment—If you installed two or more SSP nodes and each node manages separate incoming requests, the configuration is unique for each node. Refer to *Upgrade a Multiple SSP Nodes Configuration* on page 97.

Move certificates used on an HSM device in SSP version 2.0.02. Release 2.0.02 supported the use of an HSM device. To use the HSM certificates created in version 2.0.02, complete the procedure, *Move Key Certificates Created in SSP 2.0.02 on the HSM* on page 115.

Upgrade a Single SSP Node

If you installed SSP version 2.0.x on one node, use the information in this section to upgrade your environment. The following diagram compares an SSP version 2.0.x single instance environment to SSP version 3.x.



To upgrade a single node configuration created in version 2.0.x, first export information from SSP 2.0.x. Then, install an SSP version 3.x CM and engine. If you use remote perimeter servers (PS), install a new PS for each instance. Be sure to identify the settings used in version 2.0.x so that you can use this information when you install the new PS. To keep the existing PS configuration, install the new PS over the existing software. Then, run the upgrade script to convert the 2.0.x files to version 3.x. When you run the script, you define the engine to create and associate with the converted files. Refer to *Upgrade Tasks* on page 86.

Each exported object is renamed to identify the engine it is associated with. For example, if you created a Connect:Direct adapter in version 2.0.x called CDAdapter and you define the SSP node as engine1, the adapter is renamed to CDAdapter-engine1 when it is converted to SSP 3.x.

Single Node File Conversion Illustration

The following table illustrates how version 2.0.x objects are converted to version 3.x when you convert a single SSP instance. Each object name is converted to version 3.x.0. modified by adding the engine name to the end of it.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
	Engine called engine1	No engine was defined in version 2.0.x. Each SSP node performed configuration and production tasks. The engine in version 3.x performs only production tasks.
ConnectAdapter1	ConnectAdapter1-engine1 CDNETMAP-ConnectAdapter1-engine1 CDPOLICY_1-engine1 STEPINJ_1-engine1	All information associated with a Connect:Direct adapter was defined in the adapter in version 2.0.x. The conversion divides the information into four components in version 3.x: connection information in an adapter, node information in a netmap, security requirements in a policy, and step injection definitions in a step injection object. If an identical policy or step injection is defined in more than one adapter, only one item is created. The policy or step injection is then shared by the adapters.
HTTPAdapter1	HTTPAdapter1-engine1	
FTPAdapter1	FTPAdapter1-engine1	
HTTPNetmap1	HTTPNetmap1-engine1	
FTPNetmap1	FTPNetmap1-engine1	
HTTTPolicy1	HTTTPolicy1-engine1	
FTPPolicy1	FTPPolicy1-engine1	
Users	defUserStore	If you do not define a user store at conversion, a default is used. You can create one by using the -userstore argument at conversion.
System Certificates	dfltKeyStore	If you do not define a key store at conversion, a default is used. You can create one by using the -keystore argument at conversion
CA Certificates	dfltTrustStore	If you do not define a trust store at conversion, a default is used. You can create one by using the -truststore argument at conversion.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
Perimeter Server1	PerimeterServer1-engine1	
	EA_hostname_port-engine1	No EA object existed. It was defined as part of an adapter. If an EA server is defined in more than one adapter, using the same host and port, only the first instance is created and shared among the adapters.
	PASSWORDPOLICY-engine1	

Pre-Upgrade Checklist

Before you begin an upgrade, obtain the following information:

Be sure the temporary license key for version 3.x is available on the computer where you will install the engine.

If you use a remote perimeter server (PS), obtain the PS host name. If you install the PS in a less secure zone than the engine, obtain the host name and port number where the PS will be installed.

Upgrade Tasks

Complete the following tasks to upgrade a single instance of SSP:

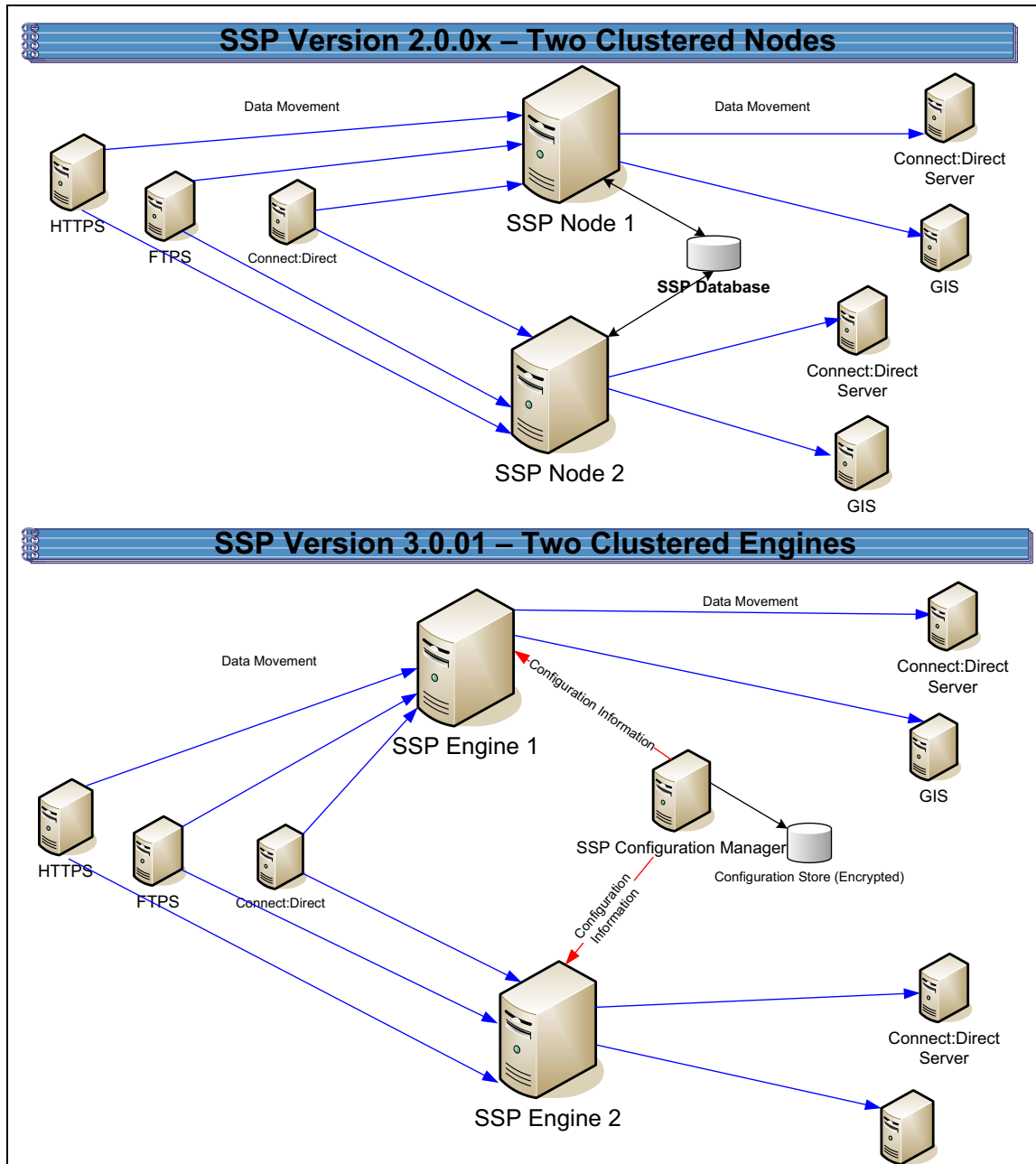
Installation Task	Procedure to Complete or Information Needed
Start SSP version 2.0.x.	<i>Start and Log On to SSP Version 2.0.x</i> on page 101.
Export the SSP 2.0.x resources.	<i>Export SSP Version 2.0.x Information</i> on page 101.
Write down the export file name and password.	
Install the SSP 3.x Engine. Note: Install the engine but do not start it.	For UNIX or Linux, refer to <i>Install or Upgrade the Engine on UNIX or Linux</i> on page 57. For Windows, refer to <i>Install or Upgrade the Engine on Windows</i> on page 69.
Install SSP 3.x CM.	For UNIX or Linux, refer to <i>Install or Upgrade CM on UNIX or Linux</i> on page 58. For Windows, refer to <i>Install or Upgrade CM on Windows</i> on page 69.
Obtain and install a license key.	For UNIX or Linux, refer to <i>Obtain a License Key File for UNIX or Linux</i> on page 59. For Windows, refer to <i>Obtain and Install a License Key File on Windows</i> on page 70.

Installation Task	Procedure to Complete or Information Needed
<p>If you use an external perimeter server (PS), do the following:</p> <ol style="list-style-type: none"> 1 Stop the version 2.0.x PS. 2 Install a version 3.x PS. 3 If SSP 2.0.x is installed on the same computer with the version 3.x engine, stop SSP 2.0.x. 	<p><i>Stop Perimeter Server Version 2.0</i> on page 102.</p> <p><i>Install a Remote Perimeter Server</i> on page 73</p> <p><i>Stop SSP Version 2.0.x</i> on page 102.</p>
Back up SSP version 3.x configuration files.	<i>Back Up Version 3.x Configuration Files</i> on page 103.
Run the upgrade script.	<i>Convert Files from SSP Version 2.0.x to Version 3.x</i> on page 103.
View the upgrade log to ensure that the conversion succeeded.	<i>Read the Upgrade Log File</i> on page 107.
Start and log on to CM.	<p>For UNIX or Linux, refer to <i>Run CM on UNIX or Linux</i> on page 327.</p> <p>For Windows, refer to <i>Run CM on Windows</i> on page 329. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm.</p>
Open the engine definition and verify the configuration.	<i>Validate an Engine Definition</i> on page 109.
Open the adapter definitions and verify each adapter configuration.	<i>Validate an Adapter</i> on page 110.
If you use a PS, validate the PS definition.	<i>Validate a PS Definition for a PS in a More Secure Zone</i> on page 110 or <i>Validate a PS Definition for a PS in a Less Secure Zone</i> on page 110.
If you changed any HTTP adapter property values, check the properties and make any necessary changes.	<i>Maintain Changes to HTTP Properties</i> on page 111.
If you made any changes to a Connect:Direct adapter properties in version 2.0.x, make the property changes in version 3.x.	<i>Implement Property Changes Made to a Connect:Direct Adapter</i> on page 114.
If you made any changes to FTP adapter properties in version 2.0.x, make the changes in version 3.x.	<i>Maintain Changes to FTP Properties</i> on page 113.
If you changed the log on attempts allowed in version 2.0.x, make the changes in version 3.x.	<i>Change How Many Times a User Can Attempt to Log In Before a Lock Occurs</i> on page 115.
Make sure that new FTP and HTTP adapter properties are correctly set.	<i>New FTP Adapter Properties in Version 3.x</i> on page 114 or <i>New Properties in Version 3.x HTTP Adapter</i> on page 113.

Installation Task	Procedure to Complete or Information Needed
Start the engine.	Refer to <i>Start and Stop Configuration Manager and the Engine</i> on page 325. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
Verify that the engine can communicate with CM.	<i>Validate the Connection Between Engines and CM</i> on page 111.

Upgrade SSP Clustered Nodes

If you installed SSP version 2.0.x on two or more nodes and created a cluster environment to provide failover support, the configuration information at each node is the same and the nodes share a database. The following diagram compares an SSP version 2.0.x cluster environment to 3.x:



To upgrade a cluster configuration created in version 2.0.x, first export information from one SSP 2.0.x node. Then, install an SSP version 3.x CM. Install an engine for each cluster node in your environment. If you use remote perimeter servers (PS), install a new PS for each instance. To keep the existing configuration, install the new PS over the existing software. To install PS in a new location, be sure to identify the settings used in version 2.0.x so that you can use this information when you install the new PS. Then, run the upgrade script to convert the 2.0.x files to version 3.x. When you run the script, you define the primary engine to create and associate with the converted files. After you determine that the configuration is working on the primary engine, use CM to create additional engines needed in the cluster environment. For each additional engine, make a copy of the adapters and associate the copy with the engine you added.

Each object exported from version 2.0.x is renamed to identify the engine it is associated with. For example, if you created a Connect:Direct adapter in version 2.0.1 called CDAdapter and you define the SSP node1 as engine1, the adapter is renamed to CDAdapter-engine1 when it is converted.

Cluster Nodes File Conversion Illustration

The following table identifies how version 2.0.x objects are converted to version 3.x for a cluster environment. Each object name is converted to version 3.x. modified by adding the engine name to the end of it.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
	Engine called engine1	No engine was defined in version 2.0.x. Each SSP node performed configuration and production tasks. The engine in version 3.x performs only production tasks.
ConnectAdapter1	ConnectAdapter1-engine1 CDNETMAP-ConnectAdapter1-engine1 CDPOLICY_1-engine1 CDSTEPINJ_1-engine1	All information associated with a Connect:Direct adapter was defined in the adapter in version 2.0.x. The conversion divides the information into four components in version 3.x: connection information in an adapter, node information in a netmap, security requirements in a policy, and step injection definitions in a step injection object. If an identical policy or step injection is defined in more than one adapter, only one item is created. The policy or step injection is then shared by the adapters.
HTTPAdapter1	HTTPAdapter1-engine1	
FTPAdapter1	FTPAdapter1-engine1	
HTTPNetmap1	HTTPNetmap1-engine1	
FTPNetmap1	FTPNetmap1-engine1	
HTTTPolicy1	HTTTPolicy1-engine1	
FTTPolicy1	FTTPolicy1-engine1	
Users	defUserStore	If you do not define a user store at conversion, a default is used. You can create one by using the -userstore argument at conversion.
System Certificates	dfitKeyStore	If you do not define a key store at conversion, a default is used. You can create one by using the -keystore argument at conversion

Version 2.0.x Object	Converts to Version 3.x Object	Notes
CA Certificates	dfltTrustStore	If you do not define a trust store at conversion, a default is used. You can create one by using the -truststore argument at conversion.
Perimeter Server1	Perimeter Server1-engine1	
	EA_hostname_port-engine1	No EA object existed. It was defined as part of an adapter. If an EA server is defined in more than one adapter, using the same host and port, only the first instance is created.
	PASSWORDPOLICY-engine1	
	Engine called engine2	This engine is not created during the conversion. Use CM to define engine2.
ConnectAdapter1	ConnectAdapter1-engine2	This adapter is not created during the conversion. Use CM to copy ConnectAdapter1-engine1 and rename it ConnectAdapter1-engine2. The netmap, policy, and step injection object are reused.
	CDNETMAP-ConnectAdapter1-engine1	
	CDPOLICY_1-engine1	
	CDSTEPINJ_1-engine1	
HTTPAdapter1	HTTPAdapter1-engine2	This adapter is not created during the conversion. Use CM to copy HTTPAdapter1-engine1 and rename it to HTTPAdapter1-engine2.
FTPAdapter1	FTPAdapter1-engine2	This adapter is not created during the conversion. Use CM to copy FTPAdapter1-engine1 and rename it to FTPAdapter1-engine2.
HTTPNetmap1	HTTPNetmap1-engine1	The netmap created during conversion is reused.
FTPNetmap1	FTPNetmap1-engine1	The netmap created during conversion is reused.
HTTTPolicy1	HTTTPolicy1-engine1	The policy created during conversion is reused.
FTTPolicy1	FTTPolicy1-engine1	The policy created during conversion is reused.
Users	defUserStore	The same user store is used by engine 1 and engine 2.
System Certificates	dfltKeyStore	The same keystore is used by engine 1 and engine 2.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
CA Certificates	dfltTrustStore	The same trust store is used by engine 1 and engine 2.
PerimeterServer2	PerimeterServer2	Perimeter servers cannot be shared by engines. Install a new perimeter server and create a new perimeter server definition for the new engine.

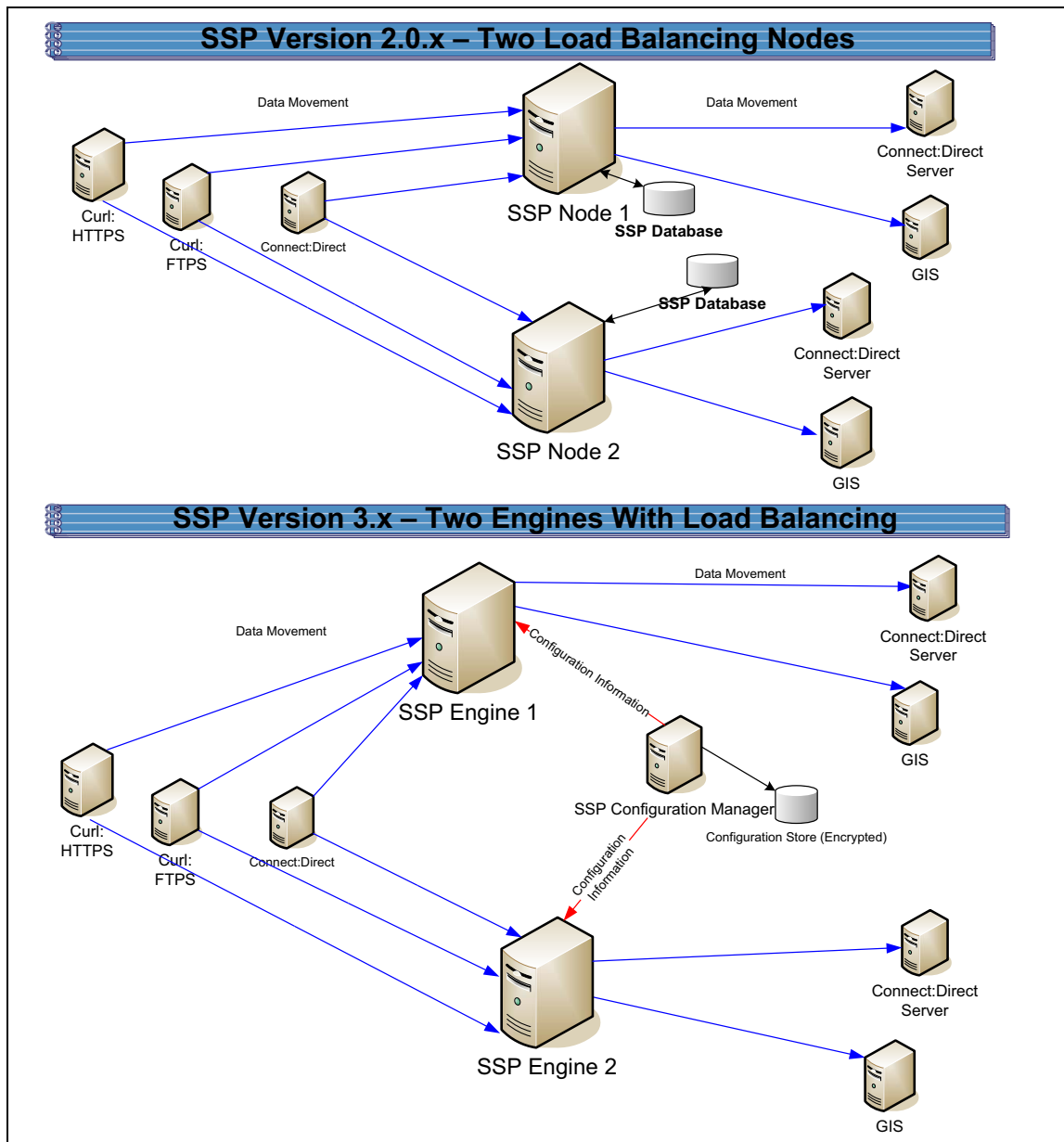
Cluster Nodes Upgrade Checklist

Complete the procedures in the *Upgrade Tasks* on page 86 to begin the upgrade. Complete the following tasks to complete the cluster node upgrade:

Installation Task	Procedure to Complete or Information Needed
Install an SSP 3.x engine at each additional cluster node. Note: Do not start the engine.	For UNIX or Linux, refer to <i>Install or Upgrade the Engine on UNIX or Linux</i> on page 57. For Windows, refer to <i>Install or Upgrade the Engine on Windows</i> on page 69.
Obtain and install a license key file for each engine.	For UNIX or Linux, refer to <i>Obtain a License Key File for UNIX or Linux</i> on page 59. For Windows, refer to <i>Obtain and Install a License Key File on Windows</i> on page 70.
Create an engine definition for each additional engine in the cluster.	<i>Create an Engine Definition</i> on page 59.
Using CM, make a copy of each adapter associated with the primary engine. Associate the adapter copy with the cluster engine you create. Repeat this for each additional node in the cluster.	<i>Copy an Adapter</i> on page 109.
Start all SSP cluster engines.	For UNIX or Linux, refer to <i>Create an Engine Definition</i> on page 59. For Windows, refer to <i>Create an Engine Definition</i> on page 70
Verify that each cluster engine can communicate with CM.	<i>Validate the Connection Between Engines and CM</i> on page 111.

Upgrade an SSP Loading Balancing Environment

If you installed SSP version 2.0.x on two or more nodes and created a load balancing environment to provide redundancy and share the workload among multiple servers, the configuration information at each node is the same but it is stored in different databases. The following diagram compares an SSP version 2.0.x load balancing environment to version 3.x.



To upgrade a load balancing configuration, export information from each SSP 2.0.x node. Be sure to specify a unique engine name and export file for each node. Then, run the upgrade script for each node.

For each export file, exported objects are renamed to identify the engine it is associated with. For example, if you created a Connect:Direct adapter in version 2.0.x called CDAAdapter and you define

the SSP node1 as engine1, the adapter is renamed to CDAdapter-engine1 when it is converted. When you run the upgrade script again and specify the engine name as engine2, a new adapter definition is created and renamed CDAdapter-engine2.

Load Balancing Nodes File Conversion Illustration

The following table identifies how version 2.0.x objects are converted to version 3.x for a load balancing environment. For each engine defined, its objects are created from a unique database. Each object name is converted to version 3.x and modified to add the engine name to the end of it.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
	Engine called engine1	No engine was defined in version 2.0.x. Each SSP node performed configuration and production tasks. The engine in version 3.x performs only production tasks.
ConnectAdapter1	ConnectAdapter1-engine1 CDNETMAP-ConnectAdapter1-engine1 CDPOLICY_1-engine1 CDSTEPINJ_1-engine1	All information associated with a Connect:Direct adapter was defined in the adapter in version 2.0.x. The conversion divides the information into four components in version 3.x: connection information in an adapter, node information in a netmap, security requirements in a policy, and step injection definitions in a step injection object. If an identical policy or step injection is defined in more than one adapter, only one item is created. The policy or step injection is then shared by the adapters.
HTTPAdapter1	HTTPAdapter1-engine1	
FTPAdapter1	FTPAdapter1-engine1	
HTTPNetmap1	HTTPNetmap1-engine1	
FTPNetmap1	FTPNetmap1-engine1	
HTTTPolicy1	HTTTPolicy1-engine1	
FTPPolicy1	FTPPolicy1-engine1	
Users	defUserStore	If you do not define a user store at conversion, a default is used. You can create one by using the -userstore argument at conversion.
System Certificates	dfltKeyStore	If you do not define a key store at conversion, a default is used. You can create one by using the -keystore argument at conversion

Version 2.0.x Object	Converts to Version 3.x Object	Notes
CA Certificates	dfltTrustStore	If you do not define a trust store at conversion, a default is used. You can create one by using the -truststore argument at conversion.
PerimeterServer1	PerimeterServer1-engine1	
	EA_hostname_port-engine1	No EA object existed. It was defined as part of an adapter. If an EA server is defined in more than one adapter, using the same host and port, only the first instance is created.
	PASSWORDPOLICY-engine1	
	Engine called engine2	No engine was defined in version 2.0.x. Each SSP node was separately managed. In version 3.x, all engines can be managed by one CM.
ConnectAdapter1	ConnectAdapter1-engine2	All information associated with a Connect:Direct adapter was defined in the adapter in version 2.0.x. The conversion divides the information into four components in version 3.x: connection information in an adapter, node information in a netmap, security requirements in a policy, and step injection definitions in a step injection object.
	CDNETMAP-ConnectAdapter1-engine2	
	CDPOLICY_1-engine2	
	CDSTEPINJ_1-engine2	
HTTPAdapter1	HTTPAdapter1-engine2	
FTPAdapter1	FTPAdapter1-engine2	
HTTPNetmap1	HTTPNetmap1-engine2	
FTPNetmap1	FTPNetmap1-engine2	
HTTTPolicy1	HTTTPolicy1-engine2	
FTPPolicy1	FTPPolicy1-engine2	
Users	defUserStore	The same user store is used by engine 1 and engine 2
System Certificates	dfltKeyStore	The same keystore is used by engine 1 and engine 2
CA Certificates	dfltTrustStore	The same trust store is used by engine 1 and engine 2.
PerimeterServer2	PerimeterServer2-engine2	

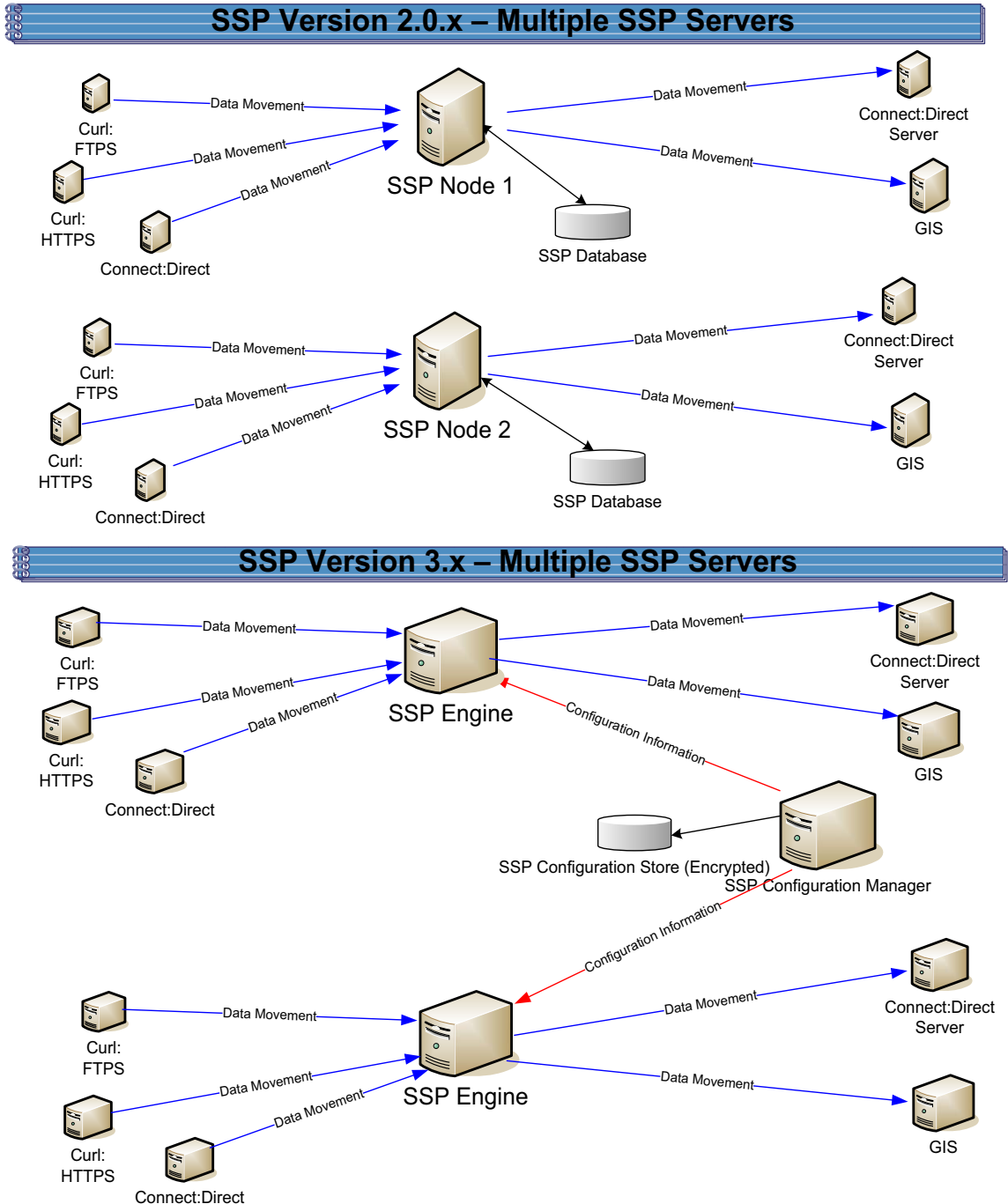
Load Balancing Nodes Upgrade Checklist

Complete the procedures in the *Upgrade Tasks* on page 86 to begin the upgrade. Perform the following procedures to complete the load balancing environment upgrade:

Installation Task	Procedure to Complete or Information Needed
Install an SSP 3.x engine at each additional load balancing location.	For UNIX or Linux, refer to <i>Install or Upgrade the Engine on UNIX or Linux</i> on page 57. For Windows, refer to <i>Install or Upgrade the Engine on Windows</i> on page 69.
Obtain and install a license key file for each load balancing engine.	For UNIX or Linux, refer to <i>Obtain a License Key File for UNIX or Linux</i> on page 59. For Windows, refer to <i>Obtain and Install a License Key File on Windows</i> on page 70.
Export SSP version 2.0.x resources from each additional SSP node.	<i>Export SSP Version 2.0.x Information</i> on page 101.
Write down the export file name and password.	
Run the upgrade script and identify the name of the additional engine (node).	<i>Convert Files from SSP Version 2.0.x to Version 3.x</i> on page 103.
View the upgrade log to ensure that the conversion for the node succeeded.	<i>Read the Upgrade Log File</i> on page 107.
From CM, verify each load balancing engine definition.	<i>Validate an Engine Definition</i> on page 109.
Open the adapter definitions for each load balancing engine. Make sure that each adapter is correctly defined.	<i>Validate an Adapter</i> on page 110.
Start all SSP load balancing engines.	For UNIX or Linux, refer to <i>Create an Engine Definition</i> on page 59. For Windows, refer to <i>Create an Engine Definition</i> on page 70.
Verify that the load balancing engine can communicate with CM.	<i>Validate the Connection Between Engines and CM</i> on page 111.

Upgrade a Multiple SSP Nodes Configuration

If you installed SSP version 2.0.x on multiple nodes and the configuration information for each node is unique, use the information in this section to identify how to upgrade your environment. The following diagram compares an SSP version 2.0.x multiple node environment to version 3.x:



To upgrade the configuration created in version 2.0.x, export information from each node. Then, run the upgrade script at each node to convert the files to version 3.x. When you run the upgrade

script, you define the engine to create and associate with the converted files. Be sure to define a unique engine name for each node.

Each exported object is renamed to identify the engine it is associated with. For example, if you created a Connect:Direct adapter in version 2.0.x called CDAdapter and you define the SSP node as engine1, the adapter is renamed to CDAdapter-engine1 when it is converted to SSP 3.x.

Multiple Node Environment File Conversion Illustration

The following table identifies how version 2.0.x objects are converted to version 3.x for a multiple node environment. For each engine defined, its objects are created from a unique database. Each object name is converted to version 3.x and modified by adding the engine name to the end of it.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
	Engine called engine1	No engine was defined in version 2.0.x. Each SSP node was separately managed.
ConnectAdapter1	ConnectAdapter1-engine1 CDNETMAP-ConnectAdapter1-engine1 CDPOLICY_1-engine1 CDSTEPINJ_1-engine1	Each object name is modified by adding the engine name to the end of it in version 3.x. All information associated with a Connect:Direct adapter was defined in the adapter in version 2.0.x. The conversion divides the information into four components in version 3.x: connection information in an adapter, node information in a netmap, security requirements in a policy, and step injection definitions in a step injection object. If an identical policy or step injection is defined in more than one adapter, only one item is created. The policy or step injection is then shared by the adapters.
HTTPAdapter1	HTTPAdapter1-engine1	
FTPAdapter1	FTPAdapter1-engine1	
HTTPNetmap1	HTTPNetmap1-engine1	
FTPNetmap1	FTPNetmap1-engine1	
HTTTPolicy1	HTTTPolicy1-engine1	
FTTPolicy1	FTTPolicy1-engine1	
Users	defUserStore	If you do not define a user store at conversion, a default is used. You can create one by using the -userstore argument at conversion.

Version 2.0.x Object	Converts to Version 3.x Object	Notes
System Certificates	dfltKeyStore	If you do not define a key store at conversion, a default is used. You can create one by using the -keystore argument at conversion
CA Certificates	dfltTrustStore	If you do not define a trust store at conversion, a default is used. You can create one by using the -truststore argument at conversion.
PerimeterServer1	PerimeterServer1-engine1	
	EA_hostname_port-engine1	No EA object existed. It was defined as part of an adapter. If an EA server is defined in more than one adapter, using the same host and port, only the first instance is created.
	PASSWORDPOLICY-engine1	
	Engine called engine2	No engine was defined in version 2.0.x. Each SSP node was separately managed. In version 3.x, all engines can be managed by one CM.
ConnectAdapter1	ConnectAdapter1-engine2	Each object name is modified by adding the engine name to the end of it in version 3.x. All information associated with a Connect:Direct adapter was defined in the adapter in version 2.0.x. The conversion divides the information into four components in version 3.x: connection information in an adapter, node information in a netmap, security requirements in a policy, and step injection definitions in a step injection object. If an identical policy or step injection is defined in more than one adapter, only one item is created. The policy or step injection is then shared by the adapters.
	CDNETMAP-ConnectAdapter1-engine2	
	CDPOLICY_1-engine2	
	CDSTEPINJ_1-engine2	
HTTPAdapter1	HTTPAdapter1-engine2	
FTPAdapter1	FTPAdapter1-engine2	
HTTPNetmap1	HTTPNetmap1-engine2	
FTPNetmap1	FTPNetmap1-engine2	
HTTTPolicy1	HTTTPolicy1-engine2	
FTPPolicy1	FTPPolicy1-engine2	
Users	defUserStore	The same user store is used by engine 1 and engine 2

Version 2.0.x Object	Converts to Version 3.x Object	Notes
System Certificates	dfltKeyStore	The same keystore is used by engine 1 and engine 2
CA Certificates	dfltTrustStore	The same trust store is used by engine 1 and engine 2.
PerimeterServer2	PerimeterServer2-engine2	

Load Balancing Multiple Node Upgrade Checklist

Complete the procedures in the *Upgrade Tasks* on page 86 to begin the upgrade. Perform the following procedures to complete the multiple node environment upgrade:

Installation Task	Procedure to Complete or Information Needed
Install an SSP 3.x engine at each additional server location.	For UNIX or Linux, <i>Install or Upgrade the Engine on UNIX or Linux</i> on page 57. For Windows, <i>Install or Upgrade the Engine on Windows</i> on page 69.
Obtain and install a license key file for each additional engine.	<i>Sterling Commerce License Key Guide.</i>
Run the upgrade script at each additional engine.	<i>Convert Files from SSP Version 2.0.x to Version 3.x</i> on page 103.
View the upgrade log to ensure that the conversion succeeded.	<i>Read the Upgrade Log File</i> on page 107.
Start and log on to CM.	For UNIX or Linux, refer to <i>Run CM on UNIX or Linux</i> on page 327. For Windows, refer to <i>Run CM on Windows</i> on page 329. Click Start or Stop an SSP Component on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm .
From CM, open the engine definition and verify the configuration.	<i>Validate the Converted Components in SSP Version 3.x</i> on page 109.
Open the adapter definitions. Make sure that each adapter is correctly defined.	<i>Validate an Adapter</i> on page 110.
Start the SSP engine.	<i>Create an Engine Definition</i> on page 59.
Verify that the engines can communicate with CM.	<i>Validate the Connection Between Engines and CM</i> on page 111.

Start and Log On to SSP Version 2.0.x

To start and log on to SSP version 2.0.x:

1. Do one of the following:
 - ◆ To start SSP on UNIX or Linux:
 - a. Change the directory to *install_dir/bin*.
 - b. Type **run.sh**.
 - c. **Enter** the passphrase that you supplied during installation.
 - ◆ To start SSP on Windows, double-click the SSP icon on your Windows desktop.

When startup is complete, a message such as the following is displayed: *Open your Web browser to http://host:port/dashboard*, where *host:port* is the IP address and port number where SSP is installed.

2. Open a browser window and type the URL address for SSP version 2.0.x.
3. Type the user ID and password in the **User ID** and **Password** fields. The default values are `proxy_admin` and `password`.

Export SSP Version 2.0.x Information

To move configuration information defined in SSP version 2.0.x to version 3.x, first export the resource files from version 2.0.x.

To export SSP version 2.0.x resource files:

1. From the Deployment menu, select **Import/Export**.
2. Next to **Export Resources**, click **Go!**
3. With **XML Document** selected, click **Next**.
4. With **No** selected, click **Next**.
5. With **Standard** selected as the export type, click **Next**.
6. Select all of the resources to export and click **Next**. Resource types include:
 - ◆ Accounts
 - ◆ Proxy Policies
 - ◆ Perimeter Servers
 - ◆ Digital Certificates
 - ◆ Proxy Netmaps
 - ◆ Service Configurations
7. Select Users as the account type to export and click **Next**.
8. To export all users, click the double-right arrows to move all users to the To Be Exported column. Click **Next**.

9. To export all permission definitions, click the double-right arrows to move all permission definitions to the To Be Exported column. Click **Next**.
10. Select CA Digital Certificates and System Certificates to export all digital certificates. Click **Next**.
11. To export all CA digital certificates, click the double-right arrows to move all certificates to the To Be Exported column. Click **Next**.
12. To export all system certificates, click the double-right arrows to move all certificates to the To Be Exported column. Click **Next**.
13. To export all proxy policies, click the double-right arrows to move all policies to the To Be Exported column. Click **Next**.
14. To export all netmaps, click the double-right arrows to move all netmaps to the To Be Exported column. Click **Next**.
15. To export all perimeter servers, click the double-right arrows to move all items to the To Be Exported column. Click **Next**.
16. To export all service configurations (adapters), click the double-right arrows to move all items to the To Be Exported column. Click **Next**.
17. Type the passphrase defined during the version 2.0.x installation twice and click **Next**.
18. Click **Finish** to export the resources and create the export file.
19. To view the export report, click **View Export Report**. Make sure that all resources were successfully exported.
20. Click **Download Export data (.xml or .jar)** to save the export file.
21. Click **Return**.

Stop Perimeter Server Version 2.0

To stop a version 2.0 perimeter server:

1. Change the directory to `/install_dir/bin` where `install_dir` is the location where the PS is installed.
2. Type **stopPs.sh** and press **Enter**.

Stop SSP Version 2.0.x

To stop SSP version 2.0.x:

1. If necessary, open SSP version 2.0.x. Refer to *Start and Log On to SSP Version 2.0.x* on page 101.
2. From the Administration menu, select **System Tools>Troubleshooter**.
3. Click **Stop the System** and wait for shutdown to complete.

Back Up Version 3.x Configuration Files

Before you upgrade version 2.0.x files to version 3.x, first back up the version 3.x configuration files. Back up the folder called `/install_dir/conf/` on the computer where CM is installed.

Convert Files from SSP Version 2.0.x to Version 3.x

After you export the resource files from SSP version 2.0.x, run the upgrade script. The script first validates the objects in the file. If an object is not valid, a warning is generated and written to the upgrade log. It then performs a dependency check to ensure that items associated with an object are available in the export file. For example, if you exported an HTTP adapter that uses SSL, the dependency check searches for the certificate used in the HTTP secure communications. If it is not available, a dependency warning is generated and written to the upgrade log. The script then converts the objects to version 3.x syntax and imports the objects into CM.

Run the script using one or more of the following modes:

Validation (-v)—reads the export file and generates a list of warnings that will occur if the file is converted. It does not convert the objects.

Default—validates the export file and determines if it can be converted. It then performs a dependency check. If no validation or dependency warnings are generated, the objects are converted. If warnings occur, the file is not converted and warnings are written to the upgrade log.

Ignore warning (-w)—validates the export file and performs a dependency check. Objects are then converted. Any dependency or validation warnings are written to the upgrade log.

Dependency check (-d)—validates the export file and determines if it can be converted. It then performs a dependency check. If no validation warnings are generated, the objects are converted. It ignores dependency warnings and writes them to the upgrade log.

Overwrite (-o)—converts an export file and if an object already exists in the version 3.x configuration, it overwrites the object with the new information. All other modes ignore an object that already exists.

Validate an Export File

Complete this procedure to validate an export file and write warnings that will occur at conversion to the upgrade log. This procedure does not convert the objects to version 3.x.

To validate an export file:

1. From the `/install_dir/bin` directory, where `install_dir` is the CM installation directory, type the following command and press **Enter**: Refer to *Upgrade Script Options* on page 106 for a description of the parameters:

```
./sspUpgrade export_file engine_name -
```

2. Type the passphrase defined at installation for SSP version 2.0.x and press **Enter**.
3. Type the passphrase defined when you installed CM.

Convert Version 2.0.x Files With New Engine If No Warnings Are Found

Complete this procedure to convert objects from SSP version 2.0.x to version 3.x and create a new engine. You identify the name of the engine to create and the engine host and port as well as the version 2.0.x file to convert on the command line.

The upgrade script reads the export file and determines if objects are valid. It then performs a dependency check to determine if any item referenced by an exported object is missing. If any object is not valid or if a dependency check warning is generated, the files are not converted. If the objects are valid, an engine is created with the values you specify. Then, objects are converted to version 3.x format and associated with the engine.

To convert the version 2.0.x export file version 3.x and create a new engine, if no warnings are generated:

1. From the `/install_dir/bin` directory, where `install_dir` is the CM installation directory, type the following command and press **Enter**. Refer to *Upgrade Script Options* on page 106 for a description of the parameters.

```
export_file_name engine_name - enginehostvalue -  
engineportvalue
```

2. Do one of the following:
 - ◆ If you have not backed up the `/install_dir/conf/` folder, type `n` and press **Enter** to stop the script. After you perform the backup, perform this procedure again.
 - ◆ Type `y` and press **Enter** to continue.
3. Type the passphrase defined at installation for SSP version 2.0.x and press **Enter**.
4. Type the passphrase defined when you installed CM 3.x and press **Enter**.

Convert Version 2.0.x Files With Existing Engine If No Warnings Are Found

Complete this procedure to convert an export file from SSP version 2.0.x to version 3.x and associate converted files with an engine that is already defined in version 3.x. You identify the name of the engine to associate the converted objects with on the command line.

The upgrade script reads the export file and determines if objects are valid. It then performs a dependency check to determine if any item referenced by an exported object is missing. If any object is not valid or if a dependency check warning is generated, the files are not converted. If the objects are valid, they are converted to version 3.x format and associated with the engine you specified.

To convert the version 2.0.x export file to version 3.x, if no warnings are generated, and associate them with an engine that is already defined in version 3.x:

1. From the `/install_dir/bin` directory where `install_dir` is the CM installation directory, type the following command and press **Enter**:

```
export_file_name engine_name
```

2. Do one of the following:
 - ◆ If you have not backed up the `/install_dir/conf/` folder, type `n` and press **Enter** to stop the script. After you perform the backup, perform this procedure again.
 - ◆ Type `y` and press **Enter** to continue.
3. Type the passphrase defined at installation for SSP version 2.0.x and press **Enter**.
4. Type the passphrase defined when you installed CM3.x and press **Enter**.

Convert Version 2.0.x Files and Ignore Warnings

Complete this procedure to convert an export file from SSP version 2.0.x to version 3.x and ignore warnings.

Caution: We strongly recommend that you resolve warnings before converting files to the version 3.x format. Converting files with warnings may prevent adapters from working. If you convert files that contain warnings or dependencies to version 3.x, be sure to resolve the warnings. Then, open and save the engine definition to ensure that the changes are pushed to the engine.

The script first reads the export file and determines if objects are valid. It then performs a dependency check to determine if any item referenced by an exported object is missing. The `-w` option allows the files to be converted to version 3.x format, even if validation warnings occur. The `-d` option allows the files to be converted to version 3.x format, even if dependency warnings occur. All warnings are written to the upgrade log.

To convert the export file even if warnings occur:

1. From a command line prompt, go to the `/install_dir/bin` directory, where `install_dir` is the CM installation directory.
2. Do one of the following:
 - ◆ To convert the export file even if validation or dependency warnings occur, type the following command:

```
export_file_name engine_name -enginehost value engineport value -
```

- ◆ To convert the export file even if dependency warnings occur, type the following command:

```
export_file_name engine_name -enginehost value -engineport value -
```

Note: To associate converted files with an engine that is already defined in version 3.x, you do not have to specify an enginehost and engineport value on the command line.

3. Do one of the following:
 - ◆ If you have not backed up the `/install_dir/conf/` folder, type `n` and press **Enter** to stop the script. After you perform the backup, start over with this procedure.

- ◆ Press **Enter** to continue.
- 4. Type the passphrase defined at installation for SSP version 2.0.x and press **Enter**.
- 5. Type the passphrase defined when you installed CM and press **Enter**.

Upgrade Script Options

Following are the arguments to use when running the upgrade script:

Argument	Description	Required
export_file_name	The name assigned to the file you exported from version 2.0.x.	Y
engine_name	The engine name where the resources should be copied. <ul style="list-style-type: none"> ◆ If this engine has not been created, the upgrade script creates it and assigns it the default values. It then adds all the resources to the engine definition. If you do not define the -enginehost and -engineport parameters, use CM to complete the engine definition. If you provide a value for the parameters called enginehost and engineport, the engine is configured as part of the upgrade procedure and is ready for use. ◆ If the engine name already exists in version 3.x, all components in the export file are added to the engine definition. 	Y
-enginehost <i>hostvalue</i>	The engine host name. The default is defaultEngineHost.	
-engineport <i>portvalue</i>	The engine port used to communicate with CM and inbound nodes. The default value is 63366.	
-userstore <i>userStoreName</i>	The name of the user store where user definitions are added. If no user store is specified, definitions are added to the default user store, called defUserStore.	
-truststore <i>trustStoreName</i>	The name of the trust store where trusted certificates are added. If no trust store is specified, trusted certificates are added to the default trust store, called dfltTrustStore.	
-keystore <i>keyStoreName</i>	The name of the keystore where key certificates are added. If no keystore is specified, key certificates are added to the default key store, called dfltKeyStore.	
-conf	An alternate location to copy the files after they are converted. The directory must already exist and must contain the key file needed to encrypt the files. The default directory is ../conf.	
-help or -h	To view help for the command.	

Argument	Description	Required
Following are the options to identify how the script is implemented:		
	If no option is defined, the upgrade process validates the parameters in the export file and performs a dependency check to determine if items referenced by an exported object are available. If any validation or dependency warnings are identified, the upgrade is stopped. If any object being upgraded already exists in CM, it is not replaced.	
-v	Performs a validation to make sure that the 2.0.x export file can be converted to version 3.x format without warnings. However, the file is not converted. Any warnings are written to a log file. Use this option to identify warnings and fix them before you move the information into version 3.x.	
-d	Converts the export file, even when dependency warnings occur. A dependency check determines if any item referenced by an exported object is missing. Dependency check warnings are written to the log. If a validation warning occurs, the upgrade process is stopped, and no files are updated.	
-w	Converts the export file, even when validation or dependency warnings occur. Be sure to resolve any warnings before you begin sending data through SSP.	
-o	If an item already exists, overwrites the item with the new information.	

Read the Upgrade Log File

After you run the upgrade script, make sure that the upgrade is successful. Read the upgrade log located in the *Engineinstall_dir*\logs folder in the Engine installation directory.

Following is a sample log message:

```
21 Apr 2010 13:09:30,746 5281 [main] WARN
com.sterlingcommerce.hadrian.tools.gis.conversion.GISConverter - General
warning(s) occurred, upgrade process stopped.
```

A message includes the following information:

Field	Description	Sample Message Text
Date and timeStamp	The date when the message is written.	21 Apr 2010 13:09:30
Process ID	An ID assigned to the message.	746 5281
Message type	The type of message written: INFO or WARN. Use the WARN messages to troubleshoot a conversion problem.	WARN

Field	Description	Sample Message Text
Program module	The module that generated the warning.	com.sterlingcommerce. hardrian.tools.gis. conversionGISConverter
Message text	A description of the informational message or warning.	General warning(s) occurred, upgrade process stopped.

Following are some of the warning messages that are written to the upgrade log. Use the messages to troubleshoot any problems that occur:

Warning Message	Description
DEPENDENCY CHECK WARNING: Netmap inbound node <i>nodename</i> is missing key certificate <i>certificatename</i>	The key certificate referenced in the netmap inbound node is missing. If you specify the -d argument on the command line, the items available in the export file will be converted to version 3.x and can be used. However, you must import the certificate into SSP 3.x before you are ready for a production environment.
Warning	A problem occurred when an item was converted to the version 3.x format.
GENERAL WARNING: Engine host and/or port is not provided for newengine, using default values.	You did not define a host and port argument for the engine you created. You must use CM to update the Engine before you are ready for production. Refer to <i>Validate the Converted Components in SSP Version 3.x</i> on page 109.
General warning(s) occurred. Upgrade process stopped.	Warnings cause the upgrade process to stop. If you want the upgrade process to continue even when warnings occur, use the -w argument.
Upgrade process begins saving configuration with warnings	The -w argument was used on the command line.
WARN:General warning (s) ignored	The -w argument was used on the command line. Even though a warning occurred the conversion continues. Be sure to validate your configuration before you move to a production environment.
Upgrade is completed successfully.	The export file was successfully converted to version 3.x format.
Validation of C:\source\temp\ssp2.0.2export\exportfile.xml is completed	The export file has been validated.
General exception(s) occurred.	The export was stopped because a warning occurred.

Copy an Adapter

When you upgrade a cluster environment, you define multiple engines. One engine is the primary engine and performs the main workload. Each additional engine performs the work, if the primary engine is unavailable. Configuration must be the same at all engines in the cluster. Engines can share configuration files for netmaps, policies, user stores, trust stores, and keystores. They cannot share adapter configuration files because each adapter is associated with one engine.

To ensure that information is the same at each engine, create a copy of each adapter defined at the primary node. Then, associate the copy of the adapter with the new engine.

To copy an adapter definition and associate it with a secondary engine:

1. If necessary, select Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter to copy.
3. Select Actions > Copy Selected.

A new item is renamed to *CopyofAdapter*, where *AdapterName* is the name of the original adapter.

4. Rename the adapter. Be sure to remove the name of the primary engine and replace it with the name of the engine you are configuring.
5. From the Engine drop-down list, select the name of the engine you are configuring.
6. Click Save.
7. Repeat this process for every adapter that you want to use with this engine.

Validate the Converted Components in SSP Version 3.x

After you run the upgrade script, the converted items are now available in SSP version 3.x. Before using SSP 3.x, open the engine, adapters, and any remote perimeter servers and validate the definitions.

Validate an Engine Definition

When you run the upgrade script, you identified an engine in the engine name parameter. If the upgrade was successful, an engine definition is now available in SSP 3.x.

If you specified the `-enginehost` and `-engineport` arguments in the upgrade script, the engine is ready to use. Use this procedure to validate the engine definition to make sure that the host and port values are correct.

If you did not specify the `-enginehost` and `-engineport` arguments in the upgrade command, an engine is defined but it does not have a valid host and port value. Use this procedure to define the host and port associated with the engine.

If necessary, gather the following information and use it as you configure the engine:

CM Field	Feature	Value
Engine Name	Name of the engine	

CM Field	Feature	Value
Engine Host	IP address of the engine	
Engine Listen Port	Port number of the engine	

To validate an engine definition:

1. Click Configuration from the menu bar.
2. Expand the Engines tree and click the engine to validate.
3. Check the following values and change them as needed:
 - ◆ Engine Host
 - ◆ Engine Listen Port
4. Click Save.

Validate an Adapter

When you perform an upgrade, version 2.0.x adapters are converted to 3.x. Before you use the adapters in a version 3.x production environment, open each adapter and validate the settings.

To view an adapter definition:

1. If necessary, select Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter to view.
3. View the configuration for the adapter. If necessary, modify the configuration.
Refer to the online help for a description of each field and valid values.
4. Click Save.
5. Click OK.

Validate a PS Definition for a PS in a More Secure Zone

To validate a perimeter server definition when the PS is in a more secure zone:

1. From CM, click Advanced from the menu bar.
2. Click the Perimeter Servers tree to expand it.
3. Click More Secure Zone to view the more secure PS definitions.
4. Click the more secure PS to validate.
5. Make sure that the Proxy Local Listen Port is correctly defined.
6. Click Save.

Validate a PS Definition for a PS in a Less Secure Zone

To validate a perimeter server definition when the PS is in a less secure zone:



1. From CM, click Advanced from the menu bar.

2. Click the Perimeter Servers tree to expand it.
3. Click Less Secure Zone to view the less secure PS definitions.
4. Click the less secure PS to validate.
5. Make sure that the Perimeter Server Host and Perimeter Server Port are correct.
6. Click Save.

Validate the Connection Between Engines and CM

After you ensure that the engine definition is valid, use the following procedure to make sure that the engine can connect to CM.

To validate engine connections:

1. Click Monitoring from the menu bar.
2. Click Engine Status (All). A list of all configured engines is displayed, including the status. Status is displayed as follows:
 - ◆  Engine is running
 - ◆  Engine is not running
3. Make sure that the engine is running.

Maintain Changes to HTTP Properties

You had the ability to modify the following properties for version 2.0.x HTTP adapters in the *install_dir/properties/httpproxy.properties* file:

Common exploits that are blocked for an adapter (blockexploit)

Commands allowed (http.commands.allowed)

Commands prohibited (http.commands.prohibited)

Maximum length of an HTTP header in an incoming HTTP request
(httpMaxHeaderFieldLength)

Maximum number of HTTP headers allowed in the incoming HTTP request
(httpMaxNumHeaderFields)

Modified properties are not maintained when you convert to version 3.x

Note: In 2.0.x, the properties applied to all HTTP adapters. In version 3.x, properties are defined for each adapter.

To maintain HTTP property changes in version 3.x:

1. Write down the changes you made to HTTP properties in version 2.0.x:

Exploit to Block: _____

Additions to methods allowed: _____

Additions to prohibited methods: _____

Maximum length of an HTTP header: _____

Maximum number of HTTP headers allowed: _____

2. Open CM version 3.x.
3. From the Configuration panel, expand the Adapters tree and click the adapter to modify.
4. On the HTTP Adapter Configuration panel, click the Properties tab.
5. To edit an existing value, type the new value in the Value field.
6. To delete an item, click the radio button to the left of an item and click Delete.
7. To add a new item, click New.
8. Modify one of the properties as required:
 - ◆ To add a block common exploits value, type `block.exploit.strings.n` as the Key value, where *n* is a unique number appended to the `block.exploit.strings` key. Be sure that you increment the number and do not duplicate an existing key. Type the value to block in the Value field.
 - ◆ To add an HTTP command allowed, type `http.commands.allowed` in the Key value. Type the commands to allow in the Value field.
 - ◆ To add an HTTP command prohibited, type `http.commands.prohibited` in the Key value. Type the commands to prohibit in the Value field.
 - ◆ To modify the maximum header fields length allowed, type `httpMaxHeaderFieldLength` in the Key value. Type the maximum header length in the Value field.
 - ◆ To modify the maximum number of header fields allowed, type `httpMaxNumHeaderField` in the Key value. Type the maximum header value in the Value field.
9. Click OK.
10. Click Save.
11. Repeat steps 3 through 10 for each adapter you want to update.

New Properties in Version 3.x HTTP Adapter

New properties are defined in version 3.x for the HTTP adapter. These properties have default values that may change the behavior of an adapter. If necessary, change one or more of these properties for your environment. Properties include:

- max.ps.client.threads—Maximum number of threads in the pool used during a connection with a client. Default value is 10.
- max.ps.server.threads—Maximum number of threads in the pool used during a connection with a server. Default value is 10.

Maintain Changes to FTP Properties

You had the ability to modify the following FTP adapter properties for version 2.0.x in the *install_dir/properties/httpproxy.properties* file:

- Commands allowed in the ftp.commands.allowed string
- Commands prohibited in the ftp.commands.prohibited string

Modified values for these properties are not maintained when you convert to version 3.x.

Note: In 2.0.x, the properties applied to all HTTP adapters. In version 3.x, properties are defined for each adapter.

To maintain FTP property changes in version 3.x:

1. Write down the changes you made to FTP properties in version 2.0.x.:

Additions to methods allowed: _____

Additions to prohibited methods: _____

2. Open CM version 3.x.
3. From the Configuration navigation panel, expand the Adapters tree and click the FTP adapter to modify.
4. On the FTP Adapter Configuration panel, click the Properties tab.
5. To edit an existing value, type the new value in the Value field.
6. To delete an item, click the radio button to the left of an item and click Delete.
7. To add a new item, click New.
8. Modify one of the properties as required:
 - ◆ To add an FTP command allowed, type ftp.commands.allowed in the Key value. Type the command to allow in the Value field.
 - ◆ To add an FTP command prohibited, type ftp.commands.prohibited in the Key value. Type the command to prohibit in the Value field.
9. Click OK.
10. Click Save.

11. Repeat steps 3 through 10 for each adapter you want to update.

New FTP Adapter Properties in Version 3.x

New FTP adapter properties are defined in version 3.x. These properties have default values that may change the behavior of an adapter. If necessary, change one or more of the following properties for your environment:

`max.ps.server.threads`—Maximum number of threads in the pool used during a connection with a server. Default value is 10.

`ftp.ssl.pbsz.required`—Identifies whether the SSL command, PBSZ, is required. Valid values include Y|Yes|y|No|N|n. The default is Y.

`ftp.ssl.prot.required`—Identifies whether the SSL command, PROT, is required. Valid values include Y|Yes|y|No|N|n. The default is Y.

`max.ps.client.threads`—Maximum number of threads in the pool used during a connection with a client. Default value is 10.

`ftp.max.command.length`—Maximum length allowed for a client command. The default is 1024. The command length is unlimited if this parameter is set to 0. If this length is exceeded, an error is logged and the connection is closed.

`ftp.max.response.length`—Maximum length allowed for a server ftp response. The default is 4096. The server ftp length is unlimited if the parameter is set to 0. If this length is exceeded, an error is logged and the connection is closed. Set this parameter to 0 when communicating with a z/OS FTP server.

Implement Property Changes Made to a Connect:Direct Adapter

You had the ability to modify properties for a Connect:Direct adapter in Version 2.0.x. If you made changes, they are not maintained when you upgrade to version 3.x. Properties that may be modified include:

`CDSP|BreadCrumbAddress=granted`—By default, this property is set to *granted* to allow information to be added to messages and identify the presence of a proxy in a communications session. You may have changed this value to *denied* to prevent proxy information from being added to a message.

`CDSP|BreadCrumbAddressTransparentContent=wishboneHoast`—Identifies the string that is placed in the Connect:Direct FMH message if `BreadCrumbAddress` is set to *denied*. If `BreadCrumbAddress` is set to *granted*, information about the adapter is placed in the FMH message.

To implement Connect:Direct property changes in version 3.x:

1. Identify the changes you made in version 2.0.x. Write down the changes below:

Connect:Direct property changes: _____

2. Open CM version 3.x.
3. From the Configuration navigation panel, expand the Adapters tree and click the Connect:Direct adapter to modify.

4. On the Connect:Direct Adapter Configuration panel, click the Properties tab.
5. Click New.
6. Type the property string in the Key field and the value in the Value field.
7. Click OK.
8. Click Save.

Change How Many Times a User Can Attempt to Log In Before a Lock Occurs

You can modify the lock out parameter for HTTP and FTP in SSP 2.0.x to change how many consecutive times a user can attempt to log in before being locked out. Any changes made to this parameter are not maintained when you upgrade to version 3.x. In addition, version 3.x changes the behavior of a user lockout. In version 2.0.x, the user remained locked out until you unlocked the account. In version 3.x, you define a lockout duration. When the lockout duration elapses, starting from the last failed login attempt, the user can then access SSP. For each user store that you define, you must identify the lockout duration and the user lockout threshold.

To change how many times a user can attempt to log in before a lock occurs and how long to lock out a user:

1. Write down the value you assigned to log in attempts allowed in SSP version 2.0.x. This value is defined in the `maxConsecutiveAuthAttempts` property in the `ftpproxy.properties` and `httpproxy.properties` files located in the `install_dir/properties` directory.

Value of Log In Attempts Allowed: _____

2. Open CM version 3.x.
3. Click Credentials on the menu bar.
4. Expand the User Store tree and click the user store where user definitions are defined. The default user store is `defUserStore`.
5. Set the user attempts allowed in the User Lockout Threshold field.
6. Identify how long a user is locked out in the User Lockout Duration field.
7. Click Save.

Move Key Certificates Created in SSP 2.0.02 on the HSM

If you used HSM to store certificates in SSP version 2.0.02 and you want to use these certificates in SSP version 3.x, complete this procedure.

Use one of the following procedures to convert HSM key certificates from SSP 2.0.02 to SSP 3.x.

To convert HSM key certificates from an SSP 2.0.02 installation:

1. Type `RemoveSystemCert -l` and redirect the output of the script to a file. This command lists the system certificates stored in version 2.0.02 and writes them to the file. Remove the lines from the top of the file up to the first "PrivateKeyInfo for ID" line.

2. Export the configuration into an XML file. Refer to *Export SSP Version 2.0.x Information* on page 101.
3. Stop SSP.

To convert HSM key certificates from an SSP 3.x installation:

1. Install the SSP 3.x engine on the same computer where SSP 2.0.02 is installed. For UNIX or Linux, refer to *Install or Upgrade the Engine on UNIX or Linux* on page 57. For Windows, refer to *Install or Upgrade the Engine on Windows* on page 69.
2. Type the following command to enable HSM support:

```
setupHSM -enable hsm=hardwaredevicename path = locationofHSMSoftware
```

Refer to *Store System Certificates on a Hardware Security Module (HSM)* on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm.

3. Start the engine. For UNIX or Linux, refer to *Create an Engine Definition* on page 59. For Windows, refer to *Create an Engine Definition* on page 70
4. Install CM. For UNIX or Linux, refer to *Install or Upgrade CM on UNIX or Linux* on page 58. For Windows, refer to *Install or Upgrade CM on Windows* on page 69.
5. Create an engine definition on CM. Make sure that the engine shows on monitoring screen as running. Refer to *Validate an Engine Definition* on page 109.
6. Stop CM.
7. Type the following command at CM to obtain the HSM keys and add them to the CM database. Identify the file that you created in step 2 on page 116 in the file parameter.

```
manageKeyCerts -loadHSM file=filecreatedfromversion2.0.02HSM
```

The *file* is the name of the file created in step 1 on page 115.

Refer to *Store System Certificates on a Hardware Security Module (HSM)* on the Documentation Library at www.sterlingcommerce.com/Documentation/SSP33/HomePage.htm for more information on the command parameters.

8. Type the following command at CM:

```
sspUpgrade d
```

Specify the engine name and the -d option to ignore dependencies. The -d option is required to ensure that the script runs successfully. The *export_file_name* is the name of the file created in step 2 on page 116.

Connect:Direct Proxy Configuration

The Connect:Direct configuration scenarios describe how to configure Connect:Direct protocol connections to and from the SSP engine using the Configuration Manager.

Note: Configuration information must be available at the engine before communication sessions with Connect:Direct can be established.

Organization of the Connect:Direct Configuration Scenarios

The first scenario instructs you how to do a basic setup. Each successive scenario adds an additional security feature to the basic configuration. After you go through each scenario, test the connection to ensure that it is correctly configured. You determine your security needs and configure the security features applicable to your environment.

The scenarios include the following:

- Create a basic Connect:Direct configuration
- Add SSL/TLS support
- Configure PNODE-based routing
- Add local user authentication
- Copy data or run a program based on the success or failure of a Connect:Direct Process step
- Block Connect:Direct tasks from a PNODE

The remaining configuration scenarios require Sterling External Authentication Server (EA), an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to EA to enforce the following advanced security features:

- Authenticate an inbound certificate or user using EA
- Configure user mapping
- Configure certificate-based routing
- Perform user mapping to the SNODE using EA

Additional procedures are provided to instruct you how to configure the following features:

- Define alternate nodes for failover support
- Enable action based on protocol errors


Complete Scenario Worksheets

Before you perform each Connect:Direct configuration, gather the information on the worksheet provided. You use this information as you configure each feature. Complete worksheets as follows:

- Enter a value for each listed SSP feature. Fields listed in the worksheet are required.
- Accept default values for fields not listed in the worksheet.
- The worksheet identifies the Configuration Manager field where you will specify each value.

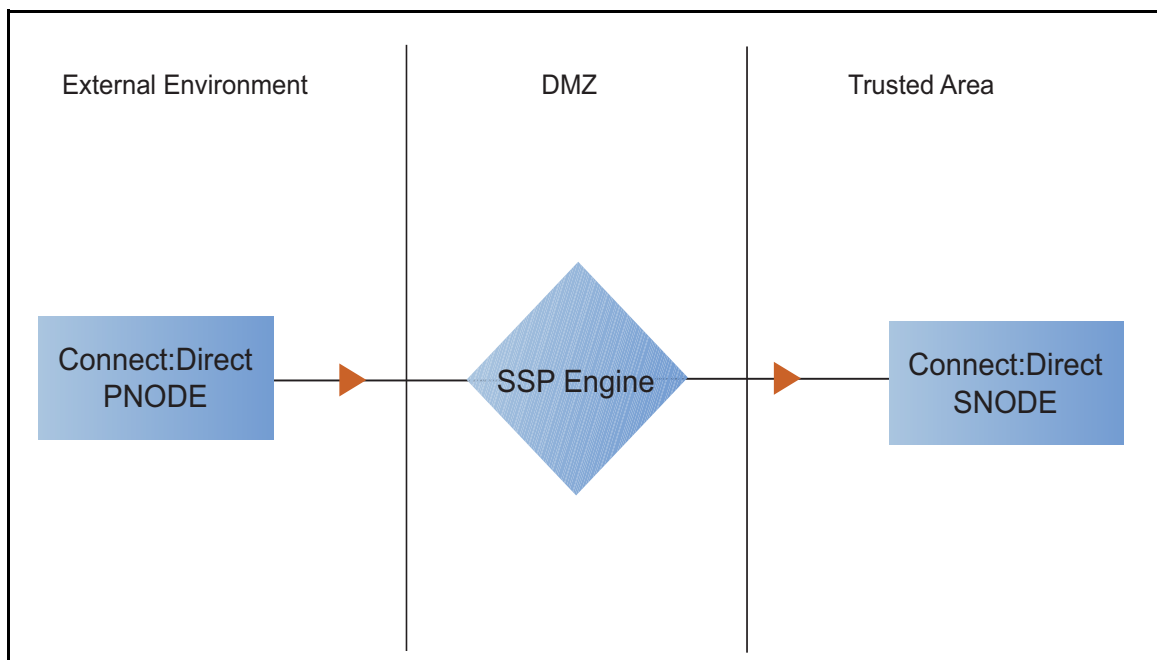
Complete and Test Configuration Scenarios

Work through the sequence of Connect:Direct configuration scenarios in the order in which they are presented to add security features. Be sure to test each feature before you add the next one to the configuration. Before you move SSP into production, ensure that you have configured and tested all security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: . To view more information about the error, hover over the icon.

Create a Basic Connect:Direct Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection between Connect:Direct servers. Using default values, the PNODE presents a User ID to connect to the SNODE without EA. As a result, no authentication occurs in SSP and the user ID presented by the PNODE is used to connect to the SNODE. The basic configuration uses standard routing to route connections to the node you define in the adapter. You are instructed on how to configure PNODE routing, mixed routing, and certificate-based routing in later scenarios.



Before you configure a Connect:Direct connection, make sure that an engine has been configured. Refer to *Install or Upgrade SSP on UNIX or Linux* or *Install or Upgrade SSP on Windows* for instructions.

After you configure SSP, validate the configuration by initiating a Connect:Direct connection from the PNODE. For more information on testing the configuration, see *Test the Connect:Direct Connections* on page 139.

Complete the following tasks to define a basic Connect:Direct configuration:

- Create a policy
- Define Connect:Direct nodes in a netmap
- Define a Connect:Direct adapter

Basic Connect:Direct Configuration Worksheet

Before you configure SSP for Connect:Direct connections, gather the information on the basic Connect:Direct configuration Worksheet. You use this information as you configure a basic Connect:Direct connection for SSP. After you configure Connect:Direct connections, validate the configuration by initiating a Connect:Direct connection from the PNODE.

Policy

Create a basic policy. In a later Connect:Direct configuration scenario, you edit this policy to add security features to it.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy	_____

Netmap (All Connect:Direct Nodes)

Create a netmap that contains connection information for the nodes connecting to and from SSP. For each node, associate a policy with the node.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name	
Connect:Direct Node Definitions		
Node Name	Name to assign to the Connect:Direct node definition	
Connect:Direct Server Address	Host name or IP address of the Connect:Direct server	
Connect:Direct Port	Listening port number of the Connect:Direct server	
Policy	Name of policy you create (Select from a pull-down list.)	
Node Name	Name to assign to the Connect:Direct node definition	
Connect:Direct Server Address	Host name or IP address of the Connect:Direct server	
Connect:Direct Port	Listening port number of the Connect:Direct server	

Configuration Manager Field	Feature	Value
Policy	Name of policy you create (Select from a pull-down list.)	

Connect:Direct Adapter

Create a Connect:Direct adapter that defines information necessary to establish Connect:Direct connections to and from SSP. When configuring the adapter, select the basic netmap and the Connect:Direct server where connections are routed and defined in the netmap definition.

Configuration Manager Field	Feature	Value
Name	Adapter name	
Listen Port	Listen port to use for inbound connections	
Netmap	Netmap to associate with the adapter	
SNODE Netmap Entry	Name of Connect:Direct node where the connection is routed	
Engine	Engine to run the Connect:Direct adapter on	

Create a Basic Connect:Direct Policy

The policy defines how you impose controls to authenticate a Connect:Direct PNODE trying to communicate with a Connect:Direct SNODE over the public Internet. The basic policy does not enforce any controls over the defined node. You add security controls when you define more advanced security settings.

To define a basic policy:

1. Select Configuration from the menu bar.
2. Click Actions > New Policy > C:D Policy.
3. Type a Policy Name.
4. Click Save.

Create a Connect:Direct Netmap

You define connection information for every Connect:Direct node that communicates using SSP. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define Connect:Direct nodes:

1. Select Configuration from the menu bar.

2. Click Actions > New Netmap > C:D Netmap.
 3. Type a Netmap Name.
 4. To define a Connect:Direct node definition, click New.
 5. Specify the following values:
 - ◆ Node Name
 - ◆ Connect:Direct Server Address or hostname
 - ◆ Connect:Direct Server Port (listening port)
 - ◆ Policy
-
- Note:** If you have not defined a policy, click the green plus sign to define one.
-
6. Click OK.
 7. Repeat steps 3 through 5 for each node you want to define. Define at least one PNODE and at least one SNODE in order to establish a connection between two Connect:Direct nodes.
 8. Click Save.

Define the Connect:Direct Adapter Used for the Connection

A Connect:Direct adapter definition specifies system-level communications information necessary for Connect:Direct connections through SSP.

Before you begin this procedure, create a netmap and an engine to associate with the adapter.

To define a Connect:Direct adapter:

1. Select Configuration from the menu bar.
2. Click Actions > New Adapter > C:D Proxy.
3. Specify values for the following:
 - ◆ Name
 - ◆ Listen Port
 - ◆ Netmap
 - ◆ SNODE Netmap Entry
 - ◆ Engine
4. Click Save.

What You Defined with the Basic Connect:Direct Configuration Scenario

Creating connections between Connect:Direct nodes when routing them through SSP requires that you organize information about the Connect:Direct nodes in a policy, a netmap, and an adapter definition. You created these items when you defined the basic Connect:Direct configuration. The next step is to test the configuration to ensure that the connections work. Before you test the configuration, be sure that:

The Connect:Direct SNODE server has a definition in its netmap for the Connect:Direct PNODE. For Connect:Direct for Windows, set the netmap.check parameter to N.

The PNODE server has a definition in its netmap for the SNODE, using the IP address and port of the SSP server.

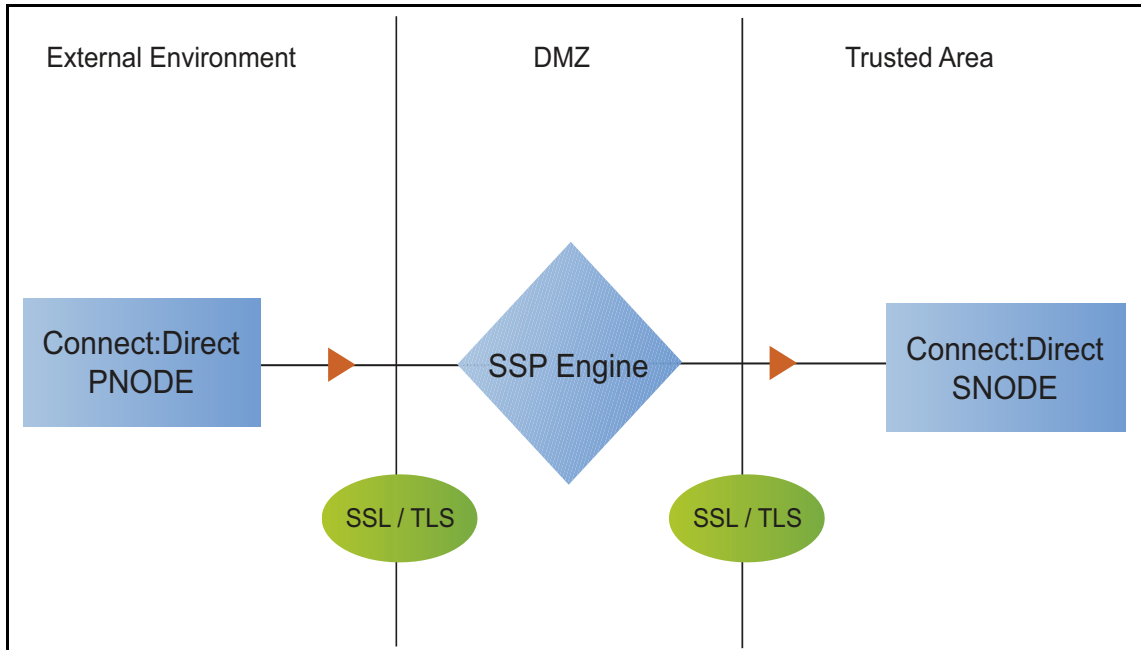
The user ID and password provided by the PNODE are defined at the Connect:Direct SNODE.

Refer to *Test the Connect:Direct Connections* on page 139 for information about testing the Connect:Direct proxy configuration outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Add SSL/TLS Support

This scenario builds on the basic Connect:Direct configuration by enabling security for the nodes you defined in the netmap.



Adding SSL/TLS support to the netmap for the nodes involves selecting the following options for the connections:

- SSL or TLS Protocol
- Cipher suites
- Certificate stores and certificates

Add SSL/TLS support to the PNODE and the SNODE definitions. Set up Secure+ parameter files at both the SNODE and the PNODE servers. Obtain certificates for both sessions and check them into the certificate store. Then, test the connection.

Note: This procedure assumes you have checked in your certificates. Refer to *Manage Certificates for SSL/TLS Transactions with Trading Partners* for more information.

SSL/TLS Support Worksheet

Before you add SSL/TLS support to the connection information you created in the basic Connect:Direct configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Select the security setting and cipher suites to be used to secure the connection. To require that the certificate common name be validated in a certificate presented, enable this option and identify the common name value to check. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Node Name	Name of the node to add security to	Select a node definition that you have already defined
Use Secure+	Enable this option to enable security checking	Enabled
Verify Common Name	Enable this option to enable common name checking. This is optional.	
Certificate Common Name	Value of common name in certificate presented, if Common Name Checking is enabled.	
Security Setting	Security protocol to use. Options include SSL, TLS, or The PNODE host controls SSL Protocol.	
Trust Store	Name of the store for the CA certificate or trusted root certificate.	
CA Certificates/Trusted Root	Name of CA certificate/trusted root	
Key Store	Name of the store for the key or system certificate is stored.	
Key/System Certificate	Name of the SSP system certificate presented to the Connect:Direct server.	
Available Cipher Suites	Select the ciphers to enable by moving them from the Available Cipher Suites to the Selected Cipher Suites field.	

Secure the Connect:Direct Connection Using the SSL or TLS Protocol

The first step to strengthen security is to secure the communications channel. This procedure describes how to enable the SSL or TLS protocol for the Connect:Direct connections to and from SSP in a netmap you created in the basic configuration. To require that SSP perform common name checking, enable this option and identify the common name in the configuration.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP Cert Store. Refer to *Manage Certificates for SSL/TLS Transactions with Trading Partners* for instructions.

To enable the SSL or TLS protocol:

1. Select Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Select a node to modify, and click Edit.
4. Click the Security tab, and then click Use secure+ to enable security.
5. To enable common name checking:
 - a. Click Verify Common Name.
 - b. Type the certificate common name in the Certificate Common Name field.
6. Select values for the following:
 - ◆ Security Setting
 - ◆ Trust Store
 - ◆ CA Certificates/Trusted Root

Note: Be sure to highlight the certificate to select. If only one certificate is displayed in the field, it is not selected until you highlight it.

 - ◆ Key Store
 - ◆ Key/System Certificate
 - ◆ Selected Cipher Suites
7. Click OK.
8. Click Save.

Establish a session initiated by a Connect:Direct PNODE to test the configuration.

Variation on the Add SSL/TLS Support Configuration

After you confirm that the communications session you established using the Add SSL/TLS Support scenario was successful, you may want to further modify your configuration. After testing the SSL/TLS configuration, you can configure the environment to allow the inbound and outbound sessions to use different levels of encryption.

Allow Different Levels of Encryption for the Inbound and Outbound Node

In a Connect:Direct environment where SSP is not being used, one session is established between an SNODE and a PNODE. In the SSP environment, a session break is created; therefore, two

sessions are established: one between the PNODE and SSP and another between SSP and the SNODE. To use the same protocol on both sessions, use the default settings.

Complete this procedure to define one protocol for the inbound node and a different protocol for the outbound node. This function is useful when you want to secure the inbound connection but allow a nonsecure session between SSP and the outbound node.

To enable different levels of encryption for the inbound and the outbound connection:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree, and select the adapter you want to modify.
3. Click the Advanced tab.
4. Enable the Inbound and outbound sessions can have different levels of encryption option.
5. Click Save.

Configure PNODE-Based Routing

The basic configuration uses standard routing to determine where a connection is routed. If you configure standard routing, all sessions through an adapter are routed to the same connection. To allow a PNODE to determine what SNODE it connects to, configure PNODE-based routing. For PNODE-based routing, you must configure a node definition in the netmap for the PNODE and for all the SNODEs you will route to.

Note: PNODE-based routing is supported for Connect:Direct for z/OS version 4.6 or higher, Connect:Direct for UNIX version 3.8 or higher, and Connect:Direct for Windows version 4.4 or higher.

PNODE-based Routing Worksheet

This scenario builds on the basic Connect:Direct configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic Connect:Direct configuration scenario, gather the information on the PNODE-based Routing Worksheet. You use this information as you configure PNODE-based routing.

In the netmap you select, make sure that you have a node definition for the PNODE and for every node where the connection is routed.

Configuration Manager Field	Feature	Value
Name	Adapter name	
Netmap	Netmap to associate with the adapter	
Listen Port	Adapter port number	
Routing Type	Routing type to use for this connection	PNODE-specified

Configure PNODE-based Routing

To configure a Connect:Direct adapter to use PNODE-based routing:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree and select the adapter you want to modify.
3. Select PNODE-specified in the Routing field.
4. Click Save.

Configure Mixed Routing

Mixed routing allows a PNODE to determine what SNODE it connects to. If the PNODE does not identify what SNODE to connect to, mixed routing then routes to the SNODE identified in the SSP configuration. Before PNODE-based routing can be implemented, you must configure a node definition in the netmap for the PNODE and the SNODE.

Note: PNODE-based routing is supported for Connect:Direct for z/OS version 4.6 or higher, Connect:Direct for UNIX version 3.8 or higher, and Connect:Direct for Windows version 4.4 or higher.

Mixed Routing Worksheet

This scenario builds on the basic Connect:Direct configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic Connect:Direct configuration scenario, gather the information on the Mixed Routing Worksheet. You use this information as you configure PNODE-based routing.

Make sure you have a node definition for the PNODE and for the node where the connection is routed in the netmap you select.

Configuration Manager Field	Feature	Value
Name	Adapter name	
Netmap	Netmap to associate with the adapter	
SNODE Netmap Entry	Name of Connect:Direct node where the connection is routed	
Routing Type	Routing type to use for this connection	PNODE-specified and then Standard (mixed)

Configure PNODE Specified and Then Standard (Mixed) Routing

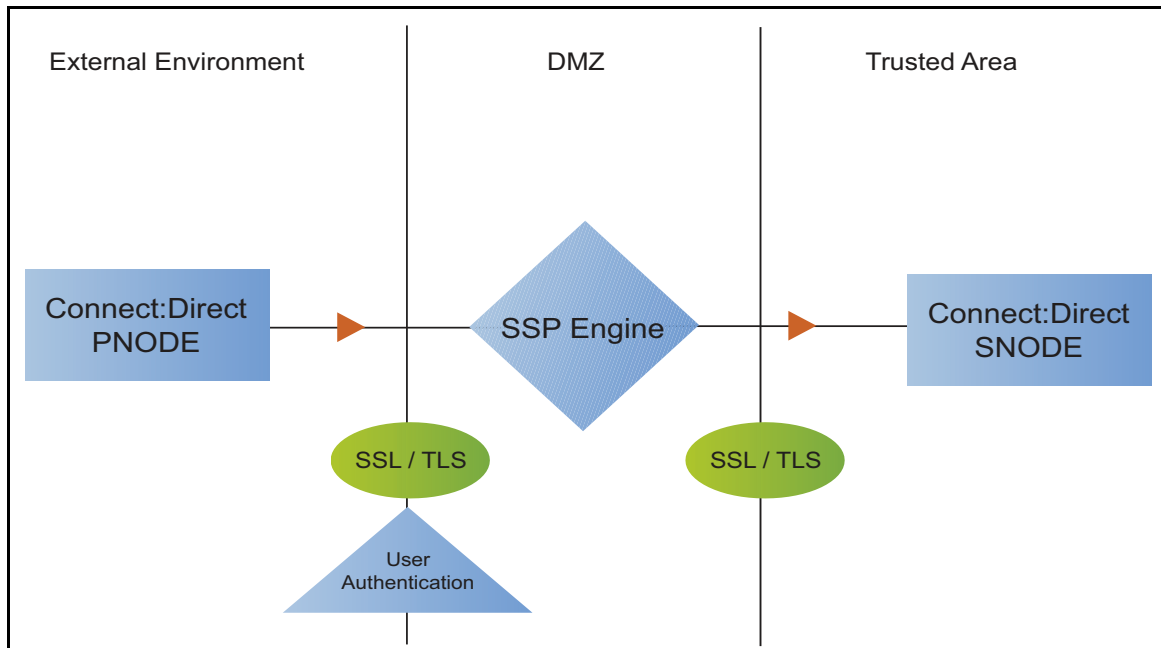
To configure a Connect:Direct adapter to use PNODE specified and then standard (mixed) routing:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree and select the adapter you want to modify.
3. Select PNODE-Specified, then Standard (mixed) in the Routing Type field.
4. Select the SNODE to route connections to in the SNODE Netmap Entry field.

- Click Save.

Add Local User Authentication to a Connect:Direct Connection

This scenario builds on the basic Connect:Direct configuration by adding local user authentication to the PNODE connection using information defined in the local user store. The user ID and password presented by the PNODE are authenticated against information stored in the local user store. The values must match before a connection is established. You must add this information to the local user store before you can test this scenario.



Adding user authentication to the PNODE connection defined in the basic Connect:Direct configuration involves enabling user authentication and specifying information about the PNODE.

After you configure local user authentication, validate the configuration by establishing a session initiated by a Connect:Direct PNODE.

Connect:Direct PNODE Connection (Local User Authentication) Worksheet

Before you add local user authentication to the PNODE connection you created in the basic Connect:Direct configuration scenario, gather the information on the Connect:Direct PNODE Connection (Local User Authentication) Worksheet. Use this information as you configure user authentication for the PNODE connection.

In this scenario, you edit the policy you created in the Connect:Direct basic configuration scenario and enable user authentication. You also add a user ID and password for the Connect:Direct PNODE to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node	
User Authentication	Method to use to authenticate the inbound node	Through local user store
User Store	Name of the user store you create	
User Name	Name of the user you define in the User Store	
Password Confirm Password	The password value to use to validate the inbound connection	

Add User Authentication to the Connect:Direct Inbound Connection

You can strengthen the security of Connect:Direct PNODE connections by enabling local user authentication. This procedure describes how to configure local user authentication.

Note: Check the netmap to ensure that the policy you select is associated with the PNODE you want to authenticate.

To add local user authentication for a PNODE connection:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and select a policy.
3. Click the Advanced tab.
4. Enable the User Authentication Through Local User Store option.
5. Click Save.

Add Credentials to the Local User Store

If you enable user authentication through the local user store, you also add user information to the local user store that is validated by SSP during a Connect:Direct client connection.

Before you begin this procedure:

Enable user authentication for the inbound connection.

Ensure that the engine is configured to use the user store that contains the user credentials.

To add user information to the local user store:

1. Select Credentials from the menu bar.
2. Click User Stores to expand the list of user stores.
3. Select the default user store called defUserStore.
4. From the User Store Configuration panel, click New.

5. Specify values for the following:
 - ◆ User Name
 - ◆ Password
 - ◆ Confirm Password
6. Click OK.
7. Click Save.

Copy Data or Run a Program Based on the Success or Failure of a Process Step (Step Injection)

This scenario builds on the basic Connect:Direct configuration by adding step injection functions to the PNODE connection. Step injection allows you to insert Connect:Direct Process statements into the communications session with the SNODE independent of the PNODE Process statements. These injected statements can provide real-time notification of file delivery, invoke applications, run operating system jobs and commands, and submit other Connect:Direct Processes, all without the need to provide an exit program on the SNODE or without changing the PNODE Process. Even though the PNODE has no indication that these steps have been executed on the SNODE, step injection is defined on the PNODE record in SSP. The results of these steps are logged in the statistics file of the SNODE.

To use step injection, define one or more of the following step injection functions:

Copy session or certificate information to a file at the SNODE at the end of a successful step.

Execute a Connect:Direct Runtask, Runjob, or Submit step on the SNODE at the end of a successful step or run operating system commands, jobs, or programs.

Copy session or certificate information to a file at the SNODE at the end of a failed step.

Execute a Connect:Direct Runtask, Runjob, or Submit step on the SNODE at the end of a failed step or run operating system commands, jobs, or programs.

Use variables to define file names and parameters in a Runtask, Runjob, or Process step.

Step Injection actions referenced in a netmap entry are only performed when the node is communicating as the PNODE in the transaction.

Configure Step Injections Worksheet

Before you create step injection definitions and associate them with a node definition you created in the basic Connect:Direct configuration scenario, gather the information on this worksheet. You use this information as you configure a node with step injection support.

Configuration Manager Field	Feature	Value
Step Injection Name	The name to assign to the step injection you define.	

Configuration Manager Field	Feature	Value
Copy on success	Turn on this option to copy session-specific data to the SNODE at the end of a successful step.	Enabled? Yes or No
Copy identifying information	The information to copy to the SNODE. This field is required if you turn on Enable Copy on success.	Select one of the following options: Copy All Information Copy Certificate Information Copy Session Information
Session information output file	Destination file on the SNODE where information is copied. Variables can be used to define the file name. Refer to <i>Use Variables in a Step Injection Definition</i> on page 132.	
Tcp timeout for copy	Number of seconds to wait for a request or response before ending the session.	
Execute on success	Turn on this option to execute a Runtask, Runjob, or Submit with a Process at the end of a successful step or execute an operating system command or program.	Enabled? Yes or No
Step selection	The step type to execute when a successful step occurs.	Select one of the following options: Runtask, Runjob, Submit
Step parameter	Type the parameters to use for the step. Variables can be used to define the parameters. Refer to <i>Use Variables in a Step Injection Definition</i> on page 132.	
Tcp timeout for step	Number of seconds to wait before timing out.	
Copy on failure	Turn on this option to copy session-specific data to the SNODE at the end of a failed step.	Enabled? Yes or No
Copy identifying information	Type of information to copy to the SNODE if a Process step is unsuccessful.	Select one of the following options: Copy All Information Copy Certificate Information Copy Session Information
Session information output file	Destination file where the copy information is written. Variables can be used to define the file name. Refer to <i>Use Variables in a Step Injection Definition</i> on page 132.	
Tcp timeout for step	Number of seconds to wait before the copy instruction is timed out.	

Configuration Manager Field	Feature	Value
Execute on failure	Turn on this option to execute a Runtask, Runjob, or Submit Process on the SNODE as defined in a submitted Process at the end of a unsuccessful step or execute an operating system command or program.	Enabled? Yes or No
Step selection	Select the step type to execute when a successful step occurs.	Select one of the following options: Runtask Runjob Submit
Step parameter	Define the parameters to use for the step. Variables can be used to define the parameters. Refer to <i>Use Variables in a Step Injection Definition</i> on page 132.	
Tcp timeout for copy	Identify how many seconds to wait before the copy instruction is timed out.	

Configure a Step Injection

Before you can associate a step injection with a node, you must first define the actions to take in a step injection function.

To configure a step injection:

1. Select Advanced from the menu bar.
2. Click Actions > New C:D Step Injection.
3. Type a step injection name.
4. Click the Advanced tab.
5. To copy information into a file at the SNODE:
 - a. Take one of the following actions:
 - Enable Copy on success to copy information to a file after a successful Process copy statement has occurred.
 - Enable Copy on failure to copy information to a file after a Process copy statement has failed.
 - b. Select the type of information to copy to the file in the Copy identifying information field. Options include Copy All Information, Copy Certificate Information, or Copy Session Information.
 - c. Type the name of the file where the information is copied in the Session information output file field.
 - d. Enter how many seconds to wait until the session is timed out in the Tcp timeout for copy field.

6. To execute a Runtask, Runjob, or submit another Connect:Direct Process or execute an operating system command or program at the SNODE:
 - a. Take one of the following actions:
 - Enable Execute on success to perform an action after a successful Process copy statement.
 - Enable Execute on failure to perform an action created after a Process copy statement fails.
 - b. Select the type of step to perform in the Step selection field. Options include Runtask, Runjob, or Submit a Connect:Direct Process or execute an operating system command or program.
 - c. Define the step parameters to use. Refer to the Connect:Direct documentation for more information.
 - d. Enter how many seconds to wait before the session is timed out in the Tcp timeout for step field.
7. Click Save.

Use Variables in a Step Injection Definition

When you configure a step injection function, you can use variables in the Session information output field and Step parameter field. These variables allow you to name output files or execute step parameters based on information obtained during the session. Use the following variables within a step injection action definition:


Variable	Description
<code>\${%DESTFILE%}</code>	Destination file name defined in the Process. Example: Runtask - cmd (copy <code>\${%DESTFILE%}</code> C:\outout\sspfile\dest.txt)
<code>\${%DESTUID%}</code>	Destination user ID defined in the Process.
<code>\${%DESTWPATH%}</code>	Destination file name defined in the Process, including the path.
<code>\${%FROMNODE%}</code>	The node that is sending the file. Returned values are P for PNODE or S for SNODE. Example: CopyonSuccess = C:\Output\copysuccessallinfo_ <code>\${%FROMNODE%}</code> _ <code>\${%SNODE%}</code> _.txt
<code>\${%ORGININUID%}</code>	User ID of the person who initiated the Process.
<code>\${%PNAME%}</code>	Process name.
<code>\${%PNODE%}</code>	Name of the PNODE that initiated the Process.
<code>\${%PNUM%}</code>	Process number.
<code>\${%SNODE%}</code>	Name of the destination SNODE name where the session is running.
<code>\${%SOURCEFILE%}</code>	Source file name defined in the Process step.
<code>\${%SOURCEWPATH%}</code>	Source file name defined in the Process step, including the path.

Variable	Description
<code>\${%STEPCOMPLETE%}</code>	What time and date the step completed, in the format <code>yyyymmdd_hhmmsshh</code> , where <code>yyyy</code> is year, <code>mm</code> = month, <code>dd</code> = day, <code>hh</code> = hour, <code>mm</code> = minute, <code>ss</code> = seconds, and <code>hh</code> = hundredth of seconds.
<code>\${%STEPMSG%}</code>	A message ID.
<code>\${%STEPNAME%}</code>	Name of the step.
<code>\${%STEPSTART%}</code>	The time and date the step started, in the format <code>yyyymmdd_hhmmsshh</code> , where <code>yyyy</code> is year, <code>mm</code> = month, <code>dd</code> = day, <code>hh</code> = hour, <code>mm</code> = minute, <code>ss</code> = seconds, and <code>hh</code> = hundredth of seconds.
<code>\${%TS%}</code>	The time the session began, in milliseconds.
<code>\${%TSNOW%}</code>	The current time, in milliseconds, in the format <code>1132599441883</code> .

Associate a Step Injection With a Connect:Direct Node

After you configure step injection functions, you can then associate a step injection with a Connect:Direct node. Process steps are activated by a PNODE; therefore, step injection functions must be defined in a PNODE record.

To associate a step injection with a Connect:Direct node:

1. Select Configuration from the menu bar.
2. Expand the Netmaps tree, and select the netmap that contains the PNODE definition you want to modify.
3. Select the node to modify and click Edit.
4. Select the step injection function to associate with the node from the Step Injection drop-down list. If you have not defined the step injection function, click  and define a step injection. Refer to *Copy Data or Run a Program Based on the Success or Failure of a Process Step (Step Injection)* on page 129 for instructions.
5. Click OK.
6. Click Save.

Block Connect:Direct Tasks Allowed on a Node

This scenario builds on the basic Connect:Direct configuration by adding the capability to prevent Connect:Direct statements from being executed.

To prevent a Connect:Direct statement from being executed:

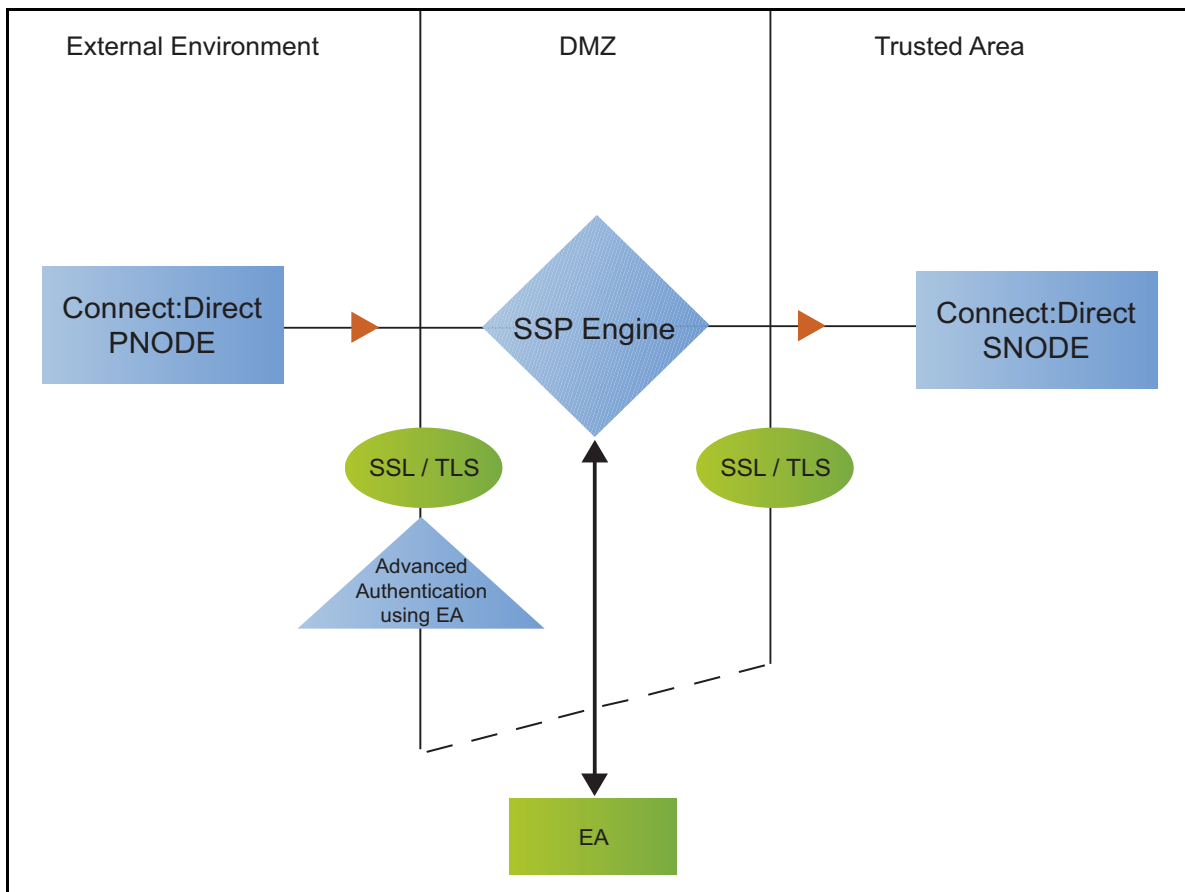
1. Select Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Step Permissions tab.
4. Click on one or more of the following tasks to disable the task:
 - ◆ Runjob step allowed

- ◆ Runtask step allowed
- ◆ Copy step allowed
- ◆ Submit step allowed

5. Click Save.

Strengthen User Authentication Using EA

This scenario builds on the basic Connect:Direct configuration by adding user authentication to the PNODE connection using information defined in EA. To provide a more advanced method of securing a Connect:Direct connection, use EA such as, authenticating certificate information or user credentials presented by the inbound node or performing user ID and password mapping.



Authenticate an Inbound Certificate or User Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines the options that are enabled. Refer to Sterling External Authentication Server help for a complete list of the functions that can be performed in EA.

Authenticate a Certificate or User Using EA - Worksheet

Use the following worksheet to identify the information needed to authenticate a Connect:Direct connection using information in EA. Update the policy you created in the basic Connect:Direct configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the certificate presented by the PNODE?	(Yes or No)
Certificate Authentication - External Authentication Profile	If yes, provide the EA certificate validation definition.	
User Authentication - Through External Authentication	Will you validate user information?	(Yes or No)
User Authentication - External Authentication Profile	If yes, provide the EA user validation definition.	

Authenticate a Connect:Direct Certificate or User Using EA

To authenticate certificate information or user information about the Connect:Direct node against information stored in an LDAP database, you must configure EA. After you configure EA to enable certificate or user authentication, complete this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP to use EA to authenticate a node connection, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

- The public keys for SSP have been sent to the EA server and imported into the EA keystore.

- The EA server connection has been configured in SSP.

To configure authentication of a Connect:Direct node using EA:

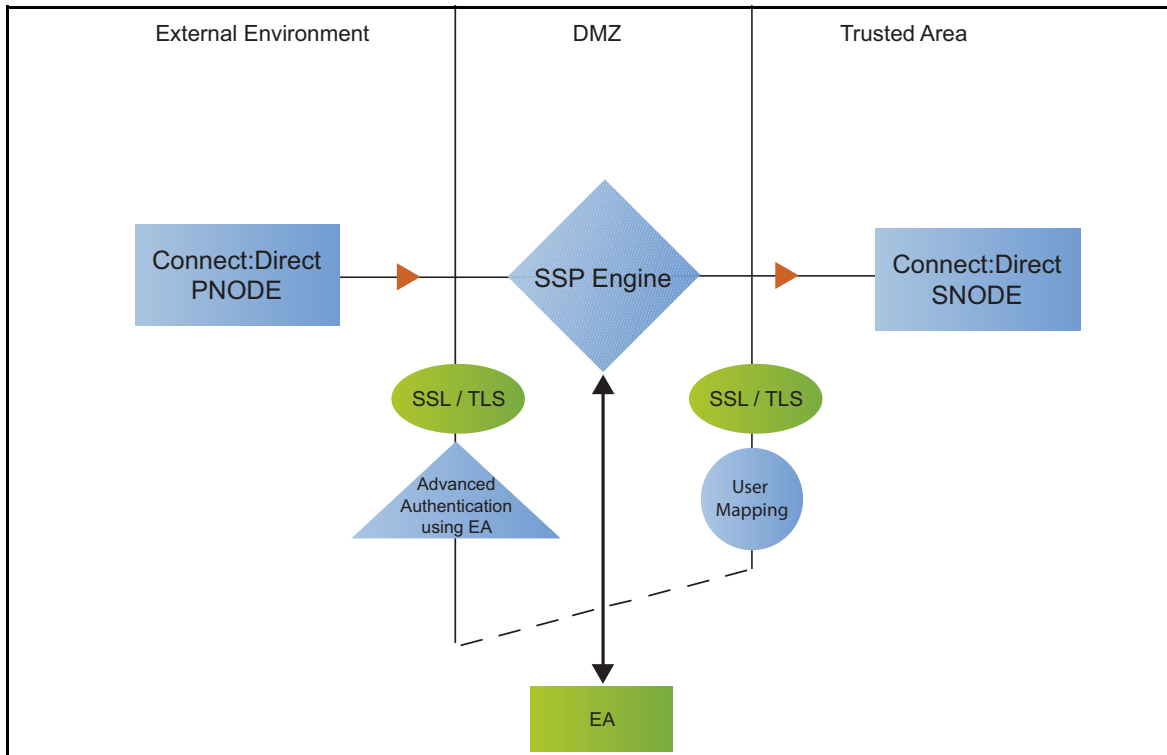
1. Select Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Configure one or more of the following options:
 - ◆ To validate the certificate presented by the node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and enter the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.
 - ◆ To enable user authentication through EA, enable User Authentication - Through External Authentication and type the name of the definition you defined in EA in the User Authentication - External Authentication Profile field.
5. If you do not want to authenticate the user using information in the local user store, deselect the Through Local User Store option.

6. Click Save.

You can now associate this policy with a Connect:Direct node where you want to perform user authentication using information stored in an LDAP database.

Strengthen the Connection to the SNODE With User Mapping

This scenario builds on the basic Connect:Direct configuration by adding user mapping using information defined in Sterling External Authentication Server (EA). To provide a more advanced method of securing a Connect:Direct connection, use EA to map a PNODE user ID and password or PNODE submitter ID to login credentials stored in EA. The mapped login credentials are then used to connect to the SNODE.



Perform User Mapping Using EA - Worksheet

Use this worksheet to identify the user mapping method to enable for the SNODE connection with information in EA:

Configuration Manager Field	Feature	Value
Replace SNODEID with Userid mapped in External Authentication	The PNODE requires a user ID for access to the SNODE and the user ID provided is replaced with a value defined in EA.	Enabled? Yes or No

Configuration Manager Field	Feature	Value
Replace submitter id with Userid mapped in External Authentication	The PNODE requires a submitter ID to access the SNODE. The submitter ID supplied by the PNODE is replaced by a valued defined in EA.	Enabled? Yes or No
Destination Service Name	The name of the service. If no value is provided, the SNODE is used as the service name.	

Perform User Mapping Using Information Stored in EA

If you store user credentials in an LDAP database, use this procedure to map a user ID and password, or a submitter ID provided by the SNODE, to information stored in EA. Two methods are available: you can replace the SNODE ID with information stored in EA or you can replace the submitter ID.

Destination Service Name needs to be selected on the Advanced tab of the Netmap Node screen of the PNODE. If Destination Service Name is not provided, the SNODE name is used.

Before you configure this option:

- Configure a definition in EA.
- Obtain the name of the EA definition.
- Configure a connection between EA and the engine.

To configure user mapping:

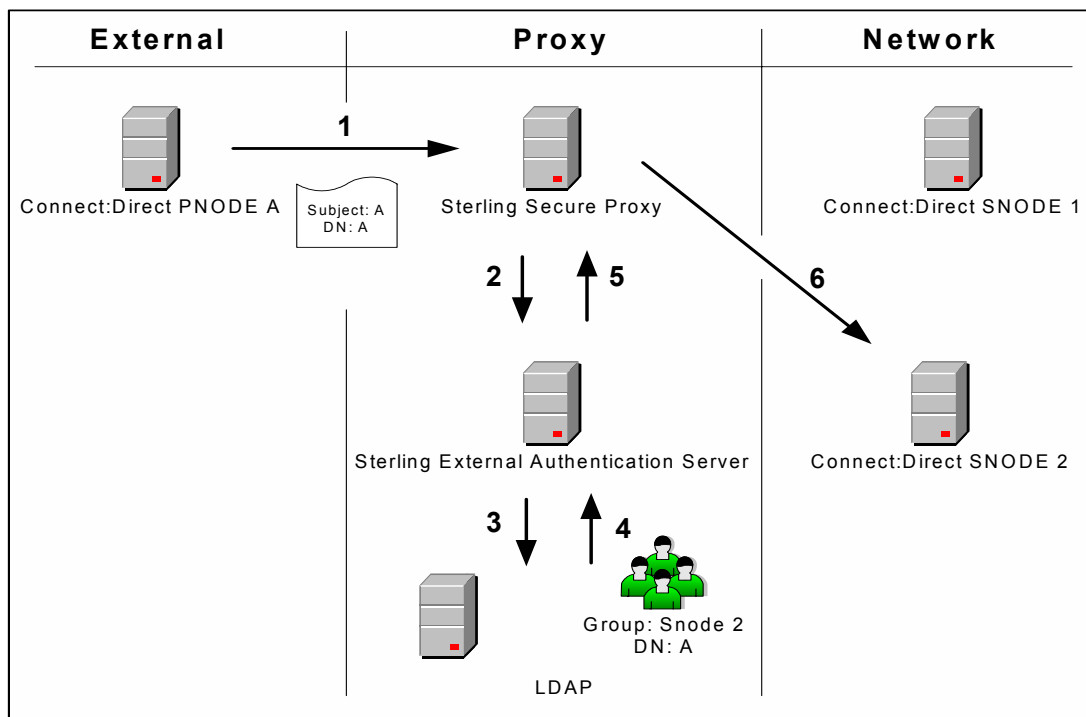
1. Select Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. To enable user authentication through EA, enable the User Authentication Through External Authentication option and type the name of the definition you defined in EA in the External Authentication Profile field.
5. Do one of the following:
 - ◆ To map the user ID presented by the PNODE to information in EA, select Replace SNODEID with UserId mapped in External Authentication.
 - ◆ To map the submitter ID presented by the PNODE to information in EA, select Replace SubmitterID with UserId mapped in External Authentication.
6. Click Save.
7. In the Configuration panel, expand the Netmap option and click the netmap to modify.
8. Select the PNODE to modify and click Edit.
9. Click the Advanced tab.
10. Type the name of the service in the Destination Service Name field. If no value is provided, the SNODE name is used as the service name.

11. Click OK.
12. Click Save.

Configure Certificate-Based Routing

This scenario builds on the basic Connect:Direct configuration by configuring certificate-based routing. Certificate-based routing uses a routing name returned by EA. It is associated with the subject distinguished name found in the PNODE certificate. SSP uses this routing name to determine the SNODE where the incoming SSP connection is routed. To perform certificate-based routing, modify an adapter you defined in the basic Connect:Direct configuration.

The following diagram illustrates the certificate-based routing function:



Summary of Certificate-Based Routing

Following are the steps performed during certificate-based routing:

1. The PNODE passes a certificate chain during an SSL/TLS session. This certificate includes several attributes, such as subject and distinguished name (DN).
2. SSP passes the certificate chain to Sterling External Authentication Server (EA).
3. Using the configuration parameters in a certificate validation request, EA attempts to match PNODE certificate attributes to the LDAP server and requests the associated routing value.
4. LDAP returns the routing value to EA.
5. EA passes the routing value to the SSP engine.
6. SSP routes the PNODE request to the SNODE using the routing value.

Configure Certificate-Based Routing in SSP

Before you test certificate-based routing, you must create a certificate validation request in EA that includes an attribute query definition called Routing Names. This attribute query definition is created to retrieve a routing name value using certificate attributes as search criteria. You must also configure a connection between SSP and EA.

Refer to *Configure SSP for Sterling External Authentication Server (EA)* for instructions.

To configure certificate-based routing:

1. Select Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter you want to modify.
3. Select Certificate-based in the Routing Type field.
4. Click Save.
5. Click the Netmap navigation panel, expand the Netmap tree, and select the Connect:Direct adapter that contains the SNODE where the connection are routed.
6. Select the node to modify and click Edit.
7. Type the routing value to be returned from the LDAP server in the Routing Name field. The routing name must exactly match the routing value returned from the LDAP server. This routing name identifies the SNODE for routing the PNODE request.
8. Click OK.
9. Click Save.
10. Configure SSP to enable certificate authentication using EA. Refer to *Authenticate an Inbound Certificate or User Using EA* on page 134.

Test the Connect:Direct Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between a Connect:Direct PNODE and the engine, initiate a session from the engine to the Connect:Direct SNODE in the trusted zone, and review the SSP log for the results.

This procedure enables you to verify that the engine can:

Establish a Connect:Direct session between a PNODE and SSP

Initiate a session to a Connect:Direct SNODE on behalf of the Connect:Direct PNODE connection

To verify the communications sessions:

1. View the `secureproxy.log`.

2. Confirm that the sessions were established, as shown in the following example.

```
21 Dec 2010 16:47:16,874 INFO [PASConduit1pnode] sys.NODE.CD_Netmap_Secure.ea -
protocol=cd sessid=111111111111 CSP004I 0 Pnode session established.
;45892 XMLErrPolicy=NONE
FMHUpdate=granted NMCheck=NA RTPolicy=Yes RJPolicy=Yes SBPolicy=Yes CPPolicy=Yes
S+Policy=SA_OPTIONAL ExecPolicy=EX_STRONG SessLimit=20 Routing=STD
CSList=TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_3DE
S_EDE_CBC_SHA, CSSelected=RSA_WITH_AES_128_CBC_SHA PNCert=Serial number: 230
Issuer:O=SCI, L=city, ST=Texas, C=US Subject:C=US, ST=Texas, O=SCI, OU=SV,
CN=donnaiaix, EMAIL=user@company.com Not Valid Before:Mon Dec 04 11:41:55 CST 2006
Not Valid After:Thu Dec 01 11:41:55 CST 2016 Signature Algorithm:MD5withRSA

21 Dec 2010 16:47:17,490 INFO [PASConduit1pnode_3016_sessid=119827723531001]
sys.NODE.CD_Netmap_Secure.ea_cd3800 - protocol=cd sessid=11111 CSP005I 0 Snode
session established with Snode=ea_unix_cd3800
```

Pnode/Snode proxy session established.

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Additional Connect:Direct Configuration Options

Additional Connect:Direct configuration options support the following features:

- Define alternate nodes for failover support
- Configure IP Address Checking (Netmap Check)
- Record an error message or shutdown a connection based on protocol errors

Define Alternate Nodes for Failover Support

If you are using standard routing to connect to a Connect:Direct server in the secure zone, you identify a primary server to connect to in the adapter. The primary nodes are defined in the netmap. For each PNODE definition in the netmap, you can identify up to three alternate outbound nodes to connect to if the primary Connect:Direct server is not available.

Two methods of configuring alternate server routing are available.

Select a previously defined outbound node from the drop-down list on the Netmap - Advanced tab. To configure this method, you first configure an outbound node definition in the netmap for each alternate node you want to use. Each connection uses the security and External Authentication settings defined for that outbound node in the netmap.

Select IP address/port from the drop-down Node list on the Advanced tab and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap, and each alternate connection shares the security and EA settings defined in the primary node definition.

If you configure alternate server definitions in the PNODE definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2 and then to the third alternate, Node 3. If all are unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. Select Configuration from the menu bar.
2. Expand the Netmaps tree and click the netmap to modify.
3. Select the node to modify and click Edit.
4. Click the Advanced tab.
5. Do one of the following:
 - ◆ To identify an alternate node that is defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.
 - ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:
 - a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.
 - b. Provide the IP Address and Port number for the alternate outbound node.
6. Click OK.
7. Click Save.

Configure IP Address Checking (Netmap Check)

IP address checking allows SSP to verify that the IP address of an inbound PNODE connection matches the IP address configured for that PNODE in the netmap. If the incoming IP address does not match the IP address of the PNODE in the netmap, SSP rejects the connection.

To enable IP address checking:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and select the policy to modify.
3. Click Check IP Address.
4. Click OK.
5. Click Save.

Once netmap checking is enabled in your Connect:Direct Proxy configuration, you can add additional IP addresses to your Connect:Direct netmap for IP address checking. For each PNODE definition in the netmap, you can identify up to 50 additional IP addresses to use for IP address checking.

Two types of additional IP addresses are available for IP address checking:

SSP can use up to three alternate outbound nodes in your netmap for IP address checking. The alternate outbound nodes can be used for inbound IP address checking and outbound failover node addresses. To configure alternate outbound nodes, refer to *Define Alternate Nodes for Failover Support* on page 140.

If you need more than three additional addresses for IP Address Checking, configure IP check addresses in the Connect:Direct netmap. SSP uses these IP check addresses for inbound IP address checking only. They cannot be used for outbound failover support.

To configure IP check addresses:

1. Select Configuration from the menu bar.
2. Expand the Netmaps tree and select the netmap to modify.
3. Select the Connect:Direct node that you want to add IP addresses to and click Edit.
4. Click the IP Checks tab.
5. If there are existing IP addresses in the Additional IP Checks table, click New to add a new blank record to the table.
6. Type the additional IP address in the table. To add another IP address, click New

Note: Before you navigate to another page in the Additional IP Checks table, click OK to save your last IP address entry.

7. When you finish adding IP addresses, click OK at the bottom of the page.
8. Click Save.

Record an Error Message or Shut Down a Connection Based on Protocol Errors

To write a warning message to the log file or shut down a connection when a protocol violation occurs during a file transfer, enable this function in the Policy definition.

To enable an action based on a protocol error:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and select the policy to modify.
3. Select the action to take on a protocol error in the Protocol Error Action field.
4. Click Save.

FTP Reverse Proxy Configuration

The FTP configuration scenarios describe how to configure FTP protocol connections to and from the SSP engine.

Note: Configuration information must be available on the engine before communication sessions with Sterling Integrator (SI) can be established.

Organization of the FTP Configuration Scenarios

The first scenario instructs you how to configure a basic configuration. Each successive scenario adds a security feature to the basic configuration. After adding a security feature, test the connection to ensure that you have correctly configured it. You determine your security needs and configure the security features applicable for your environment.

The following scenarios help you configure and test SSP for FTP protocol connections to the Sterling Integrator server:

- Create a basic FTP configuration
- Add SSL/TLS support
- Perform user authentication using the local user store
- Provide outbound credentials using the netmap

The remaining configuration scenarios require EA, an optional security feature that must be configured independently. After EA is configured, you can update your basic security definitions to enable SSP to connect to the EA to enforce the following advanced security features:

- Authenticate an inbound certificate or user using EA
- Manage connection requirements to the outbound server using EA

Other options help you do the following:

- Define alternate nodes for failover support
- Define a passive data outbound port range for an FTP Reverse Proxy adapter
- Define a passive NAT address for an FTP Reverse Proxy adapter
- Define an active data outbound port range for an FTP Reverse Proxy adapter


Complete FTP Scenario Worksheets

Before you begin configuring SSP for FTP connections, gather the information on the worksheet provided with the scenario. You use this information as you configure each feature. Complete worksheets as follows:

- Provide a value for each SSP feature listed. Fields listed in the worksheet are required.
- Accept default values for fields not listed.
- The worksheet identifies the Configuration Manager field where you will specify each value.

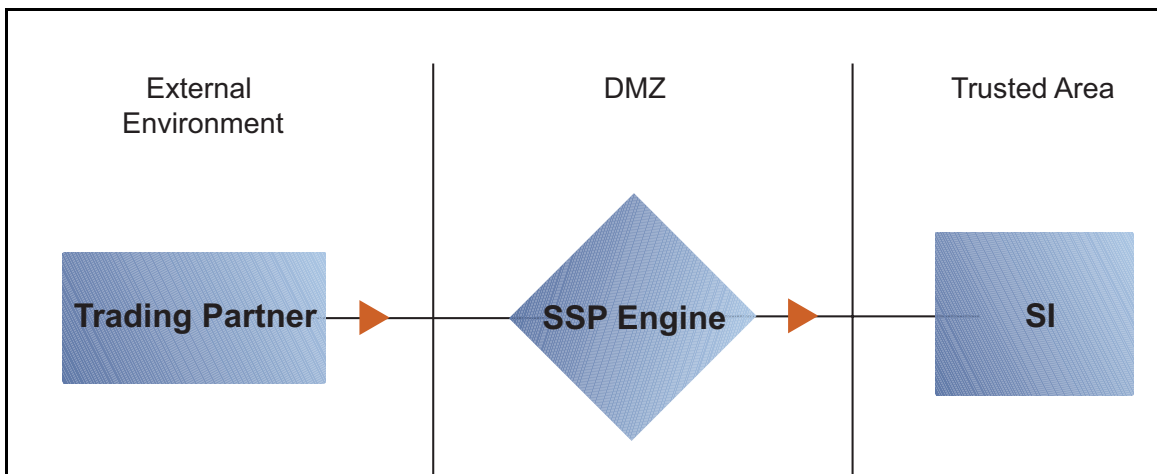
Complete and Test FTP Configuration Scenarios

Work through the sequence of FTP configuration scenarios in the order in which they are presented to add and test more security features. Be sure to test each feature before you add the next to the configuration. Before you move SSP into production, ensure that you have configured and tested the security features needed for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed:  To view more information about the error, hover over the icon.

Create a Basic FTP Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection from a trading partner to the Sterling Integrator server as illustrated below. You accept default values when configuring this scenario. As a result, no authentication occurs in SSP and credentials presented by the inbound node are passed through to the Sterling Integrator server.



After you configure SSP, validate the configuration by initiating an FTP connection from the trading partner. For more information on testing the configuration, see *Test the Inbound and Outbound FTP Connections* on page 162 *Test the Inbound and Outbound FTP Connections*.

Complete the following tasks to define a basic FTP configuration:

- Create a policy
- Define inbound and outbound connections in a netmap
- Define an FTP adapter

Basic FTP Configuration Worksheet

Before you configure SSP for FTP connections, gather the information on the Basic FTP Configuration Worksheet. You use this information as you configure a basic FTP connection.

FTP Policy

Create a basic policy. In a later FTP configuration scenario, you edit this policy to add security features to it.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy.	

FTP Netmap (Inbound and Outbound Connections)

Create a netmap that contains connection information for the nodes connecting to and from SSP: the trading partner (inbound node) and the Sterling Integrator server (outbound node). You will also associate the basic security policy you create with the inbound node.

Configuration Manager Field	Feature	Value
Netmap Name	Name of the netmap.	

Inbound Trading Partner Information

Inbound Node Name	Trading partner name (name to assign to inbound node definition).	
Peer Address Pattern	Host name or IP address pattern.	* (* allows all inbound nodes to connect to the Sterling Integrator server, using this definition. To define a more specific node definition, see <i>Create a Basic FTP Configuration</i> on page 144.
Policy	Name of policy you create.	This value is selected from a pull-down list.

Outbound FTP Server Connection

Node Name	Outbound FTP server node name.
Primary Destination Address	Host name or IP address to connect to the outbound FTP server.
Primary Destination Port	Port number to connect to the outbound FTP server.

FTP Adapter

Create an FTP adapter that defines information necessary to establish FTP connections to and from SSP. When you configure the adapter, select the basic netmap and the outbound FTP server you

define in the netmap definition. If the outbound host uses virtual IP address, set the IP address in the PASV response.

Configuration Manager Field	Feature	Value
Adapter Name	Adapter name.	
Listen Port	Listen port to use for inbound connections.	
Netmap	Netmap to associate with the adapter.	
Standard Routing Node	Name of the outbound node corresponding to the Sterling Integrator server where inbound connections are routed.	
Engine	Engine to run on.	

Create an FTP Policy

The FTP policy defines how you impose controls to authenticate a trading partner trying to access Sterling Integrator server over the public Internet.

To define a policy:

1. Click Configuration from the menu bar.
2. Click Actions > New Policy > FTP Policy.
3. Type a Policy Name.
4. Click Save.

Create an FTP Netmap

You define inbound connection information for your trading partners and outbound connection information for the Sterling Integrator server that SSP connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define inbound and outbound nodes:

1. Click Configuration from the menu bar.
2. Click Actions > New Netmap > FTP Netmap.
3. Type a Netmap Name.
4. To define an inbound node definition, click the Inbound Nodes tab and click **New**.
5. Specify the following values:
 - ◆ Inbound Node Name
 - ◆ Peer Address Pattern
 - ◆ Policy

Note: If you have not defined a policy, click the green plus sign to define one.

6. Click OK.
7. To define an outbound node definition, click the Outbound Nodes tab and click New.
8. Specify the following values:
 - ◆ Outbound Node Name
 - ◆ Primary Destination Address
 - ◆ Primary Destination Port
9. Click OK.
10. Click Save.

Define the FTP Adapter Used for the Connection

An FTP adapter definition specifies system-level communications information necessary for FTP connections to and from SSP. You can create multiple adapter definitions.

Before you begin this procedure, create the following definitions:

A netmap to associate with the adapter

An engine definition to associate with the adapter. Refer to *Install or Upgrade SSP on UNIX or Linux* or *Install or Upgrade SSP on Windows* for instructions.

To define an FTP adapter:

1. Click Configuration from the menu bar.
1. Click Actions > New Adapter > FTP Reverse Proxy.
2. Specify values for the following:
 - ◆ Adapter Name
 - ◆ Listen Port
 - ◆ Netmap
 - ◆ Standard Routing Node
 - ◆ Engine
3. Click Save.

What You Defined with the Basic FTP Configuration Scenario

Creating secure connections to Sterling Integrator servers on behalf of nodes external to your trusted zone requires that you organize information about the trading partners and the Sterling Integrator server in a policy, a netmap, and an adapter definition. You created these items when you defined the Basic FTP Configuration. The next step is testing the configuration prior to configuring additional security features. Before you test the configuration, be sure that:

The Sterling Integrator server has an active FTP server adapter configured to listen for the port specified in the outbound node definition.

The user ID and password provided by the inbound node is defined at the Sterling Integrator server.

Refer to *Test the Inbound and Outbound FTP Connections* on page 162 for information about testing the FTP Reverse Proxy configurations outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Variations on the Basic FTP Configuration

After you confirm that the communications sessions you established using the Basic FTP configuration were successful, you may want to validate sessions using other types of inbound trading partner definitions before you add complexity to the security configuration. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

Inbound FTP Trading Partner Node Definitions

You can modify the inbound trading partner node definitions as follows:

- Define a specific IP address
- Define a wildcard peer pattern
- Define an IP/subnet pattern

Define Connection Requirements Between SSP and Inbound FTP Nodes

You define connection requirements between SSP and inbound nodes by defining inbound node definitions. Refer to your company security requirements to determine how tightly to define the parameters an inbound node must provide to allow a connection.

You can define inbound node definitions to allow only one individual inbound connection, or you can identify IP address patterns and create an inbound definition that allows inbound connections that match the pattern to connect to SSP. Methods of defining inbound nodes are as follows:

Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address are allowed. A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32. SSP also supports individual host names. They must match the value returned by a reverse DNS lookup.

Define an inbound node entry that allows all nodes that match an IP/subnet address pattern. Patterns include:

Matching the first 16 bits of an IP address pattern. For example, 10.20.0.0/16 allows all IP addresses that begin with 10.20.* to connect to SSP.

Matching the first 8 bits of an IP address pattern. For example, 10.0.0.0/8 allows all IP addresses that begin with 10.* to connect to SSP.

Define an inbound node entry that allows all inbound nodes that match a wildcard host name pattern. When a connection is attempted and you have defined a wildcard host name pattern definition, a reverse DNS lookup is performed on the IP address of the inbound connection. The DNS name is compared to the wildcard pattern. Wildcard patterns include:

* matches any number of characters before or after a period. For example, *.a.com allows a connection from b.a.com but not from a.bc.com. A single * allows all inbound nodes to successfully connect to SSP.

? matches one character. For example, a?.com allows a connection from a.b.com but not from a.bc.com.

You can define more than one inbound node definition and use a combination of the node definition methods. However, be sure to order the definitions from most specific to least specific. When an inbound node connection is attempted, SSP compares the IP address of the inbound node to the first inbound node definition. If it matches, a connection is established. If it does not match, SSP checks the next inbound node definition until a match is found. If no match is found, the connection is terminated.

Inbound FTP Connection Definition Worksheet

Use the following worksheet to identify the information needed to configure inbound node definitions for a specific inbound node or for groups of inbound nodes that match a pattern.

Configuration Manager Field	Define Inbound Trading Partner Information	Value
Note: If you define a single node and definitions for multiple nodes using pattern matching, ensure that you order the definitions from most specific to least specific, because SSP processes them in the order in which they are listed.		
Inbound Node Name	Trading partner name.	
Policy Name	Policy to associate with the inbound trading partner.	
For a Single Node		
Peer Address Pattern	IP address	
	Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. SSP supports host name. An example definition is a.b.com.	
	A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32.	
For Multiple IP Addresses Using IP/Subnet Pattern		
Peer Address Pattern	Peer Address IP/Subnet Pattern Options.	
For Multiple Nodes Using Wildcard Peer Address Pattern to Validate Inbound DNS		
Peer Address Pattern	Wildcard Peer Address Pattern.	

Define Inbound Node Connection Definitions for an FTP Connection

This procedure instructs you how to modify the basic FTP configuration to add inbound node definitions for 1) a group of nodes with similar information, or 2) that limit access to one specific inbound node. It assumes that you have already configured an adapter. Gather a list of all inbound trading partners, including names and IP addresses.

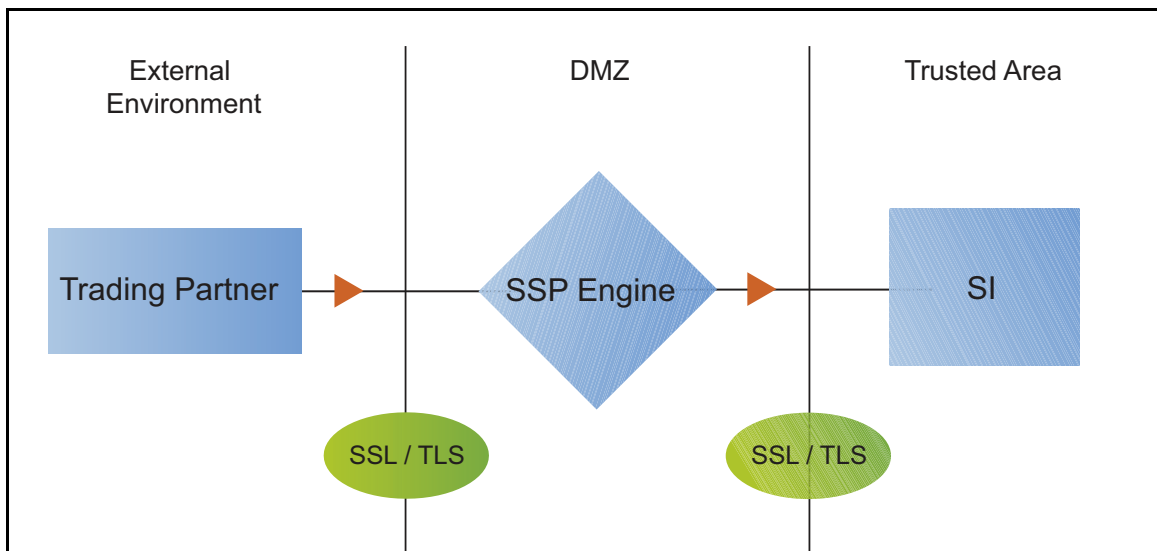
To define inbound connection definitions:

1. Identify patterns that can be used to define groups of inbound nodes.
2. To increase security, you need to define a trading partner connection for any individual IP address.
3. Click Configuration from the menu bar.

4. Expand the Netmaps tree and click the netmap to modify.
5. Click New to add a new inbound node definition.
6. Using the information you defined on the Inbound Connection Definition Worksheet, provide the following information, and click Save:
 - ◆ Inbound Node Name
 - ◆ Peer Address Pattern
 - ◆ Policy
7. Repeat step 6 for every group of connections and every individual IP address connection you want to define.
8. If necessary, reorder the node definitions in the netmap. Order definitions from most specific to least specific because they will be evaluated in order.
 - a. Click the radio button beside the inbound node definition to move.
 - b. Click Move Up or Move Down until the node definition is in the correct order.
9. Click Save.

Add SSL/TLS Support for an FTP Connection

This scenario builds on the Basic FTP Configuration by enabling security for the inbound and outbound nodes you defined in the netmap. Following is a diagram to illustrate the addition of SSL or TLS to the inbound and outbound node connections.



To add SSL/TLS support to the netmap for the inbound and outbound nodes, select the following options for the connections:

- Protocol
- Cipher suites
- Stores and certificates

To effectively configure and test this scenario:

1. Add SSL/TLS support to the inbound node definition first and establish a session initiated by an FTP client to an Sterling Integrator server.
2. Then, add SSL/TLS support to the outbound node definition and establish a session initiated by an FTP client to an Sterling Integrator server.

Note: Before you configure SSL or TLS support, you must check in your certificates. Refer to Manage Certificates for SSL/TLS Transactions with Trading Partners.

SSL/TLS Support Worksheet

Before you add SSL/TLS support to the connection information you created in the Basic FTP Configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Inbound Connection for FTP

Select the security setting and cipher suites to be used to secure the connection. To configure client authentication, enable this option. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Inbound Node Name	Name of inbound node to add security to.	Select an inbound node definition from the list.
Security Setting	Security protocol to use.	(SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Enable Client Authentication	Do you want to require that the inbound connection present its certificate for SSL or TLS client authentication?	(Yes or No)
Trust Store	If client authentication is enabled, identify the trust store used to verify the client certificate.	
CA Certificates/Trusted Root	Name of CA certificate/trusted root (if client authentication is enabled).	
Key Store	The location where the keys and system certificates you want to use are stored.	
Key/System Certificate	Name of SSP system certificate presented to the inbound connection during the handshake.	
Available Cipher Suites Selected Cipher Suites	Select the ciphers to enable by moving them from the Available Ciphers to the Selected Ciphers field.	

Outbound Connection for FTP

Select the security setting and cipher suites to be used to secure the outbound connection. Select the key/system certificate to use to validate the connection.

Configuration Manager Field	Feature	Value
Outbound Node Name	Name of outbound node to add security to.	Select a node definition from the list
Security Setting	Security protocol to use.	(SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Trust Store	If client authentication is enabled, identify the trust store where the certificate is stored.	
CA Certificates/Trusted Root	Identify the certificate to use to secure the outbound connection.	
Key Store	The location where the keys and system certificates you want to use are stored.	
Key/System Certificate	System certificate used to validate the Sterling Integrator server.	
Available Cipher Suites Selected Cipher Suites	Cipher suites to enable.	

Secure the Inbound FTP Connection Using the TLS or SSL Protocol

The first step in strengthening security is to secure the communications channel. This procedure describes how to enable the TLS or SSL protocol for the inbound connection to authenticate SSP to the trading partner initiating the connection. To require that SSP authenticate the inbound node, enable client authentication.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP certificate store.

To enable the TLS or SSL protocol on the inbound FTP node:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Inbound Nodes tab.
4. Select an inbound node to modify, and click Edit.
5. Click the Security tab, and then click Secure Connection to enable security.
6. Select values for the following:
 - ◆ Security Setting
 - ◆ Key Store

- ◆ Key/System Certificate
 - ◆ Available Ciphers
 - ◆ Selected Ciphers
7. To enable client authentication:
 - a. Click Enable Client Authentication.
 - b. Select the Trust Store where the certificate you want to use is located.
 - c. Select the CA Certificates/Trusted Root to use to authenticate the certificate presented by the inbound node.

Note: Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

8. Click OK.
9. Click Save.

Variations on the SSL/TLS Configuration on the Inbound FTP Node

After you confirm that the communications sessions you established using the basic FTP configuration with SSL/TLS enabled on the inbound node were successful, you may want to enable a clear control channel.

Enable a Clear Control Channel for an Inbound FTP Node Connection

If your environment requires that a firewall be able to see the flow of FTP commands and responses, enable the clear control channel option. Enabling clear control channel for the inbound node requires that the inbound FTP client send the clear control channel command and switch the control channel to an unencrypted channel after user authentication is completed.

To enable a clear control channel for an inbound node:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Inbound Nodes tab and select the Inbound Node to modify.
4. Click Edit.
5. Click the Security tab.
6. Enable Clear Control Channel.
7. Click OK.
8. Click Save.

Secure the Outbound FTP Connection Using the TLS or SSL Protocol

If the Sterling Integrator server has enabled the use of SSL or TLS to secure the connection, you must enable the TLS or SSL protocol in the SSP outbound node configuration. This procedure describes how to enable the TLS or SSL protocol to authenticate the Sterling Integrator server to SSP when establishing an outbound connection.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP certificate store.

To enable the TLS or SSL protocol:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Outbound Nodes tab.
4. Select an outbound node to modify, and click Edit.
5. Click the Security tab, and then click Secure Connection to enable security.
6. Select the following security options for the node:
 - ◆ Security Setting
 - ◆ Trust Store
 - ◆ CA Certificates/Trusted Root

Note: Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

- ◆ Key Store
 - ◆ Key/System Certificate
 - ◆ Available Ciphers Suites
 - ◆ Selected Ciphers Suites
7. Click OK.
 8. Click Save.

Variations on the Add SSL/TLS Support on the Outbound Node

After you confirm that the communications session you established using the Add SSL/TLS Support scenario was successful, you may want to further modify your inbound and outbound nodes. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

The following variation applies to this configuration:

Note: You must obtain the necessary certificates and place them in the SSP certificate store before you can configure these options.

Create your own trust store and key store

Enable a clear control channel for an outbound connection

Enable a Clear Control Channel for an Outbound FTP Node Connection

If your environment requires that a firewall be able to see the flow of FTP commands and responses, enable the clear control channel option. If clear control channel is enabled on the outbound node,

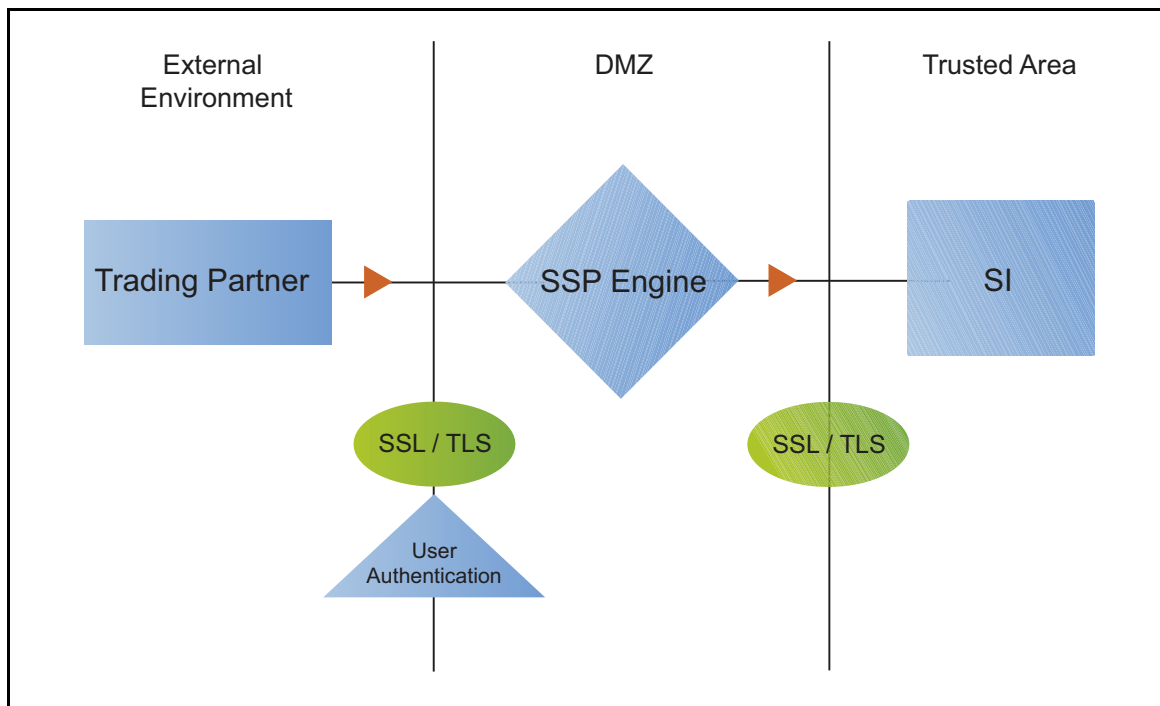
the FTP reverse proxy adapter sends the clear control channel command and switches the command channel to an unencrypted channel, after user authentication is completed.

To enable a clear control channel for an outbound node:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Outbound Nodes tab and select the Outbound Node to modify.
4. Click Edit.
5. Click the Security tab.
6. Enable Clear Control Channel.
7. Click OK.
8. Click Save.

Add Local User Authentication to the Inbound FTP Connection

This scenario builds on the Basic FTP Configuration by adding local user authentication to the inbound connection using information defined in the local user store. The user ID and password presented by the inbound node are authenticated against the information stored in the local user store. The values must match before a connection is established. You must add this information to the local user store before you can test this scenario. Following is an illustration of the secure features supported in this scenario:



Adding user authentication to the inbound connection defined in the Basic FTP Configuration involves enabling user authentication and specifying information about the trading partner.

After you configure user authentication using the local user store information, validate the configuration by establishing a session initiated by an FTP client to an Sterling Integrator server.

FTP Inbound Connection (Local User Authentication) - Worksheet

Before you add user authentication to the inbound connection you created in the Basic FTP Configuration scenario, gather the information on the FTP Inbound Connection (Local User Authentication) - Worksheet. Use this information as you configure user authentication for the inbound connection.

In this scenario, you edit the policy you created in the FTP Basic Configuration scenario and enable user authentication. You also add a user ID and password for the trading partner to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	_____
User Authentication	Method to use to authenticate the inbound node.	Through Local User Store
User Store	Name of the user store you create.	
User Name	Name of the user you define in the User Store.	
Password Confirm Password	The password value to use to validate the inbound connection.	

Add Local User Authentication to the FTP Inbound Connection

You can strengthen the security of inbound connections by enabling user authentication. This procedure describes how to add user information to the local user store to be validated by the engine during an inbound FTP client connection.

Note: Check the netmap to ensure that the policy you select is associated with the inbound nodes you want to authenticate.

To add user authentication for an inbound connection:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy to modify.
3. Click the Advanced tab.
4. Enable the User Authentication Through Local User Store option.
5. Click OK.
6. Click Save.

Add Credentials to the Local User Store

If you enable user authentication through the local user store, you have to add user information to the local user store for validation by SSP during an inbound FTP client connection.

Before you begin this procedure:

Enable user authentication for the inbound connection.

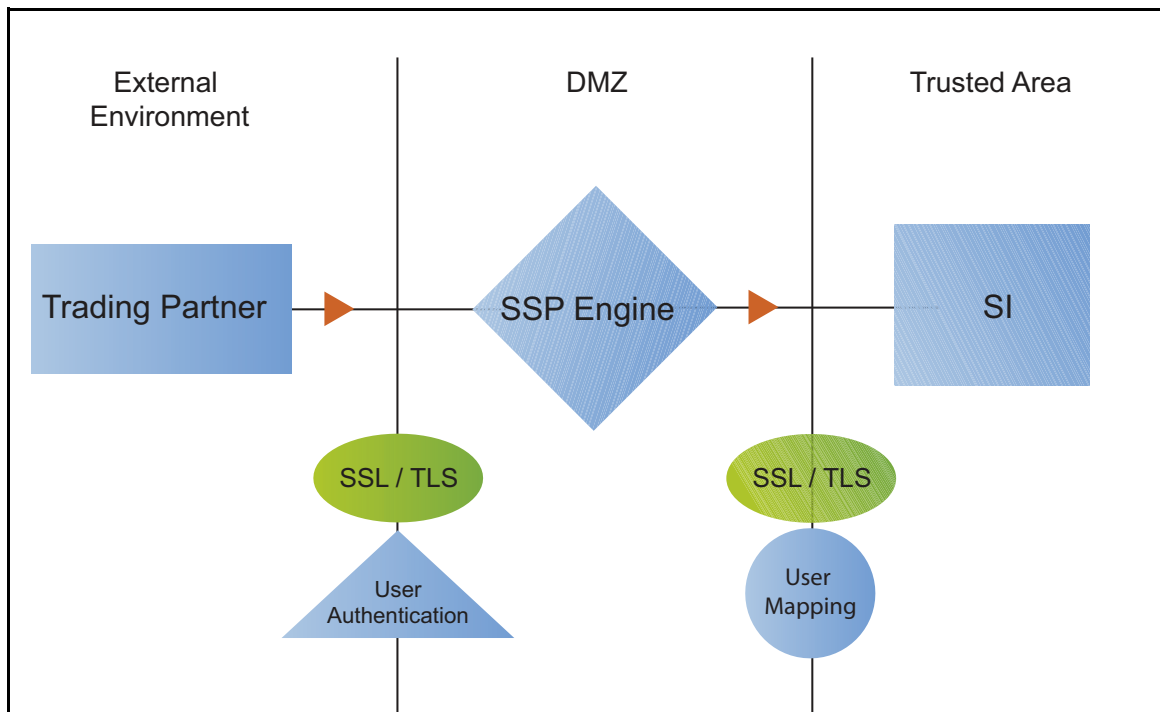
Ensure that the engine is configured to use the user store containing the user credentials.

To add user information to the local user store:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree and select a user store to modify.
3. From the User Store Configuration panel, click New.
4. Specify values for the following:
 - ◆ User Name
 - ◆ Password
 - ◆ Confirm Password
5. Click Save.

Provide Sterling Integrator Credentials to the Outbound FTP Node Using the Netmap

This scenario builds on the Basic FTP Configuration by enabling the use of user credentials from the netmap to connect to the outbound Sterling Integrator connection. Following is an illustration of the security features supported in this scenario:



When an inbound trading partner connects to SSP, its credentials are replaced with credentials stored in the netmap. The replacement credentials are then used to connect to the outbound FTP server. This method uses SSP security features to prevent trading partners from knowing the credentials used to connect to the outbound Sterling Integrator server. The outbound Sterling Integrator server must have a user definition that accepts the user ID and password provided.

After you configure the environment to use credentials defined in the netmap, test the configuration by establishing a session initiated by an FTP client to an Sterling Integrator server. Refer to *Test the Inbound and Outbound FTP Connections* on page 162 for more information on testing the configuration described in this scenario.

Provide Credentials for the Outbound FTP Node Using the Netmap Worksheet

In this scenario, edit the netmap and policy you created in the Basic FTP Configuration to provide user credentials stored in SSP to connect to the outbound Sterling Integrator connection.

Collect the following information so you can match the SSP configuration with the Sterling Integrator server configuration. Use the information on this worksheet as you edit the outbound node definition, and be sure to select the netmap and policy you created in the Basic FTP Configuration.

Configuration Manager Field	Feature	Value
User ID	User ID used to connect to the Sterling Integrator server. (Must also be defined at the Sterling Integrator server.)	
Password	Password to connect to the Sterling Integrator server. (Must also be defined at the Sterling Integrator server.)	

Connect to the Outbound FTP Server Using Credentials from the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to store the user ID and password to connect to the outbound Sterling Integrator server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and SSP uses information stored in the netmap to connect to the outbound FTP server.

Before you configure this option:

- Ensure the user ID and password are defined on the Sterling Integrator server.
- Obtain the user ID and password.

To configure validation for the outbound connection using credentials stored in the netmap:

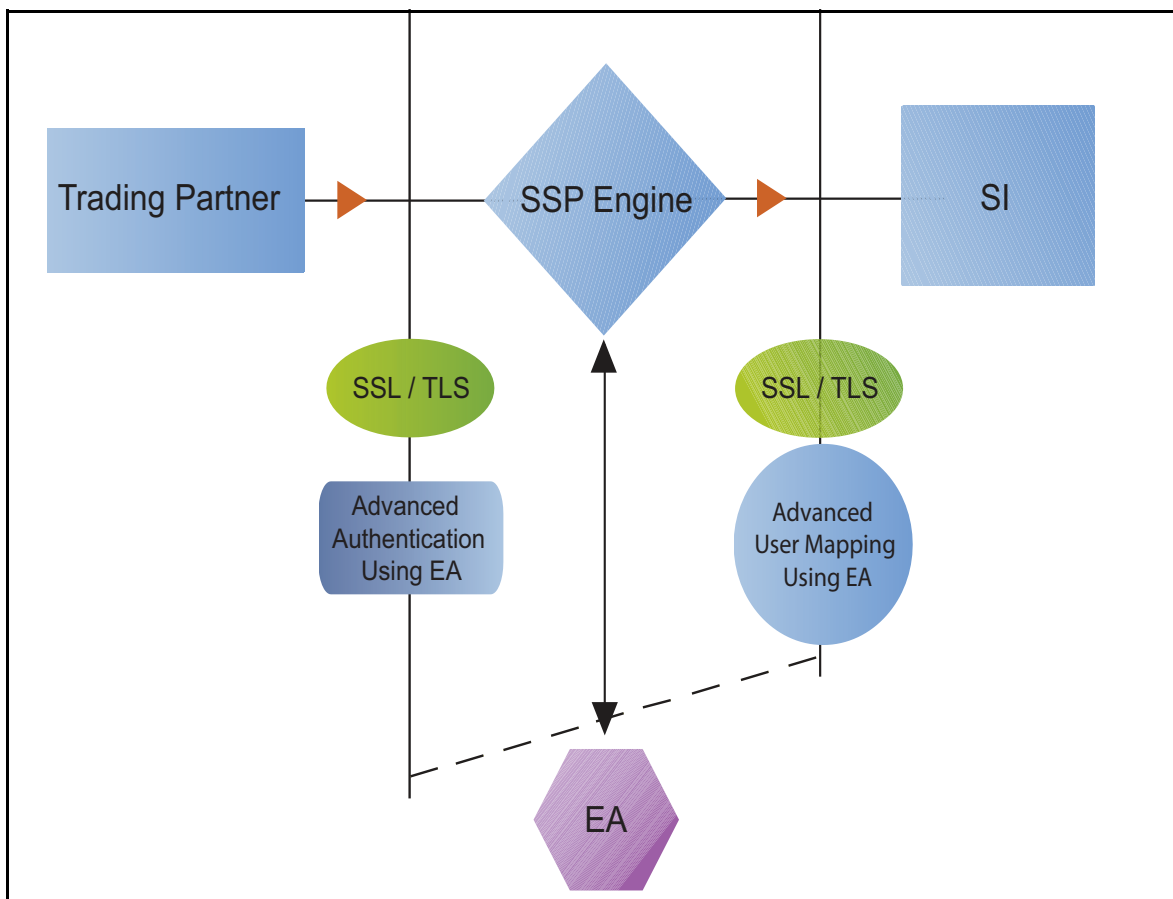
1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select the FTP netmap to modify.
3. Click the Outbound Nodes tab.
4. Select the outbound node to modify and click Edit.
5. Click the Advanced tab.

6. Type values in the following fields for connecting to the Sterling Integrator server:
 - ◆ User ID
 - ◆ Password
7. Click Save.
8. Expand the Policies tree and select the policy to modify.
9. On the FTP Policy Configuration panel, click the Advanced tab.
10. From the User Mapping: Internal User ID list, select From Netmap.
11. Click Save.

Test the configuration to ensure that this feature is working.

Strengthen Authentication of an FTP Node Using EA

Use EA to provide a more advanced method of securing the inbound or the outbound connection, such as, authenticating certificate information or user credentials presented by the inbound node, or performing user ID and password mapping for the internal credentials. The following illustrates the security features enabled in this scenario:



Authenticate an Inbound FTP Certificate or User Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. Following are some of the functions that EA can perform:

- Validate certificates, including dates and signatures
- Verify the presence of X.509 v3 extensions
- Enforce minimum key length requirements
- Check certificates against certificate revocation lists (CRLs)
- Perform LDAP queries

The EA definition determines the options that are enabled.

Manage Connection Requirements to the Outbound FTP Server Using EA

For a higher level of security when connecting to the outbound server, use information stored in an LDAP database. To use information in an LDAP database, you configure EA. You can use EA to map a user ID and password provided by an inbound connection to a unique user ID and password that is not exposed to the external node.

Authenticate an Inbound FTP Certificate or User Using EA Worksheet

Use the following worksheet to identify the information needed to authenticate a trading partner using information in EA. Update the policy you created in the Basic FTP Configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the inbound certificate?	(Yes or No)
Certificate Authentication - External Authentication Profile	If yes, identify the EA certificate validation definition.	
User Authentication - Through External Authentication	Will you validate user information?	(Yes or No)
User Authentication - External Authentication Profile	If yes, identify the EA user validation definition.	

Authenticate the Inbound FTP Node Using EA

To authenticate certificate information or user information about the inbound node against information stored in an external database, you must configure EA. After you configure EA to enable certificate validation or user authentication, use this procedure to configure SSP to use the authentication method you defined in EA.

Before you configure SSP to use EA to authenticate an inbound node authentication, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

The policy associated with the inbound node has enabled client authentication.

The public keys for SSP have been sent to the EA server and imported into the EA key store.

The EA server connection has been configured in SSP.

To configure authentication of an inbound node using EA:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Configure one or more of the following options:
 - ◆ To validate the certificate presented by the inbound node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and identify the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.
 - ◆ To validate the user, enable Through External Authentication and identify the name of the profile defined in EA in the External Authentication Profile field.
5. Click Save.

You can now associate this policy with the inbound node on which you want to perform user authentication using information stored in an LDAP server.

Connect to Outbound FTP Server Using EA Worksheet

Use this worksheet to configure a stronger outbound connection using information from an LDAP database.

Configuration Manager Field	Feature	Value
User Certification Through External Authentication	Will you validate user information?	Yes
External Authentication Profile	If yes, identify the EA user validation definition	
Destination Service Name	Identify the destination server that can be accessed by the outbound node, when using EA to map a user ID and password. Valid values are 1-255 alphanumeric characters and certain special characters. The following characters are not allowed: ! @ # % ^ * () + ? , < > { } [] ; " ' .	

Connect to the Outbound Node Using Information Stored in EA

If you store user credentials in an external database accessed by EA, use this procedure to configure SSP to use these credentials to connect to the secure outbound server.

Before you configure this option:

- Configure a user validation definition in EA.

- Obtain the name of the EA definition.

- Configure the EA server to allow connections from SSP.

- Ensure that the policy associated with the inbound node has enabled client authentication.

- Ensure that the public keys for SSP have been sent to the EA server and imported into the EA key store.

To configure the use of credentials from EA:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Enable the User Authentication Through External Authentication option.
5. Type the name of the definition you defined in EA in the External Authentication Profile field.
6. Deselect the Local User Store option.
7. From the Internal User ID field, select From External Authentication.
8. Click Save.
9. Expand the Netmaps tree and click the HTTP netmap to modify.
10. On the HTTP Netmap panel, click the Outbound Nodes tab.
11. Select the node to edit and click Edit.
12. Click the Advanced tab.
13. Identify the destination service name to use to connect the outbound node when using EA in the Destination Service Name field.
14. Click OK and click Save.

Test the Inbound and Outbound FTP Connections

To verify that the engine can receive and initiate communication sessions, you have to establish a connection between an FTP client and the engine, initiate a session from the engine to the Sterling Integrator server in the trusted zone, and review the SSP audit log for the results.

Note: Configuration files must be available at the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

- Establish an FTP session initiated by a trading partner using an FTP client

- Initiate an outbound session to an Sterling Integrator server on behalf of the FTP client connection

To verify the communications sessions:

1. Make sure the engine is running.
2. Initiate an FTP client session to the Sterling Integrator server in your trusted zone.
3. View the Inbound Node Log and the Outbound Node Log.
4. Confirm that the data transfer was successful, as illustrated in the sample log below:

Sample Inbound Node Log

```
11 Sep 2010 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http Sterling IntegratorD=1
SNAME=user.company.com SIP=10.20.200.100 SPORT=40134 SSP104I Session: 1 -
Session Proceeding after Node match: Any
11 Sep 2010 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=user.company.com SIP=10.20.200.100 SPORT=40134 DNAME=lunar.company.com
DIP=10.20.246.42 DPORT=10054 SUID=admin DUID=admin SSP102I Session: 1 -
Control:ServerAgent Connection closed (CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

Sample Outbound Node Log

```
11 Sep 2010 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=user.company.com SIP=10.20.246.121 SPORT=40134 SSP104I Session: 1 -
Session Proceeding after Node match: Any

11 Sep 2010 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=user.company.com SIP=10.20.200.100 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.200.40 DPORT=10054 SUID=admin
DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Additional FTP Configuration Options

Additional FTP configuration options are available for the following features:

- Route an Outbound FTP Connection to Alternate Sterling Integrator Servers
- Define a Passive Data Outbound Port Range for an FTP Reverse Proxy Adapter
- Define a Passive NAT Address for an FTP Reverse Proxy Adapter
- Define an Active Data Outbound Port Range for an FTP Reverse Proxy Adapter
- Use IP address from PASV response for outbound data connections

Route an Outbound FTP Connection to Alternate Sterling Integrator Servers

When you configured the adapter, you identified the Sterling Integrator server to connect to by selecting one of the outbound node connections defined in the netmap. For each outbound node definition, you can identify up to three alternate outbound nodes to connect to if the primary Sterling Integrator server is not available.

Two methods of configuring alternate Sterling Integrator server routing are available.

Select an Sterling Integrator server from the drop-down list. Using this method, you first configure an outbound node definition in the netmap for each alternate Sterling Integrator server you want to use. Each connection uses the security and External Authentication settings defined in the outbound node definition.

Select IP address/port from the drop-down list and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap, and each alternate connection uses the security and External Authentication settings defined in the primary node definition.

If you configure alternate Sterling Integrator server definitions in the outbound node definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2. If this connection is unsuccessful, SSP tries to connect to the third alternate, Node 3. If the connection to this node is unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Outbound Node tab and select the node to modify.
4. Click the Advanced tab.
5. Do one of the following:
 - ◆ To identify an alternate node that is defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.
 - ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:
 - a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.
 - b. Provide the IP address and port number for the alternate outbound node.
6. Click OK.
7. Click Save.

Define a Passive Data Outbound Port Range for an FTP Reverse Proxy Adapter

Two modes can be used to open the FTP data channel: active mode and passive mode. The mode used on the inbound connection is determined by the client. SSP always uses passive mode for the outbound connection.

In active mode, the inbound FTP node sends a port command identifying the data channel listen port on which SSP needs to connect. You identify the port numbers to use for an active data connection in the Active Data Outbound Port Range field.

In passive mode, the FTP client sends a PASV command and SSP starts a listener to receive the data connections from the client. You identify a port number range to use to start the listener that receives connections.

To define a passive data outbound port range:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter to modify.
3. Click the Advanced tab.
4. Type a value to use for the passive data outbound port range in the Passive Data Listening Port Range field.
5. Click Save.

Define a Passive NAT Address for an FTP Reverse Proxy Adapter

When a PASV command is sent to SSP from an inbound FTP client, the host and port number to which the inbound FTP client needs to connect for the data channel is returned. When SSP is behind a firewall, the host address of SSP is not visible to the inbound FTP client. To ensure that the client can obtain this information, define the passive NAT address.

Define this value if the client cannot directly connect to the proxy, such as when using a static network address translation (NAT).

If you are using a remote external perimeter server with the FTP reverse proxy adapter and the perimeter server is also behind a firewall using static network address translation, identify the name or IP address of the computer running the external perimeter server.

To define a passive NAT address:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter to modify.
3. Click the Advanced tab.
4. Type a value to use for the passive NAT address in the Passive NAT Address field.
5. Click Save.

Define an Active Data Outbound Port Range for an FTP Reverse Proxy Adapter

Two modes are available to open the FTP data channel: active mode and passive mode. The mode used on the inbound connection is determined by the client. SSP always uses passive mode for the outbound connection.

In active mode, the inbound FTP node sends a port command identifying the data channel listen port on which the proxy needs to connect. You identify the port numbers to use for an active data connection in the Active Data Outbound Port Range field.

In passive mode, the FTP client sends a PASV command to SSP and SSP starts a listener to receive the data connections from the client. You identify a port number range that can be used to start the listener to receive connections.

To define an active data outbound port range for an FTP reverse proxy adapter:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter to modify.
3. Click the Advanced tab.
4. Type a value to use for the active data outbound port range in the Active Data Outbound Port Range field.
5. Click Save.

Use IP Address from a PASV Response For Outbound Data Connections

As a security measure, SSP ignores PASV response and uses the same IP address as the initial control channel for all data channel connections. If VIPI (virtual IP address) is used by the outbound server, the data connections may be a different IP address than the original connection. Complete this procedure to allow data channel connections to use different IP addresses.

To allow SSP to use an IP address from a PASV response:

To define an active data outbound port range for an FTP reverse proxy adapter:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter to modify.
3. Click the Advanced tab.
4. Enable the field Use IP from PASV Response.
5. Click Save.

HTTP Reverse Proxy Configuration

The HTTP configuration scenarios describe how to configure HTTP protocol connections to and from the engine.

Note: Configuration must be available on the engine before communication sessions with Sterling Integrator can be established.

Organization of the HTTP Configuration Scenarios

The first scenario instructs you on how to configure a basic configuration. Each successive scenario adds an additional security feature to the basic configuration. After configuring each scenario, test the connection to ensure that you have correctly configured it. You determine your security needs and configure the security features applicable for your environment.

The following scenarios help you configure and test SSP for HTTP protocol connections to the Sterling Integrator server:

- Create a basic HTTP configuration
- Add SSL/TLS support
- Perform user authentication using the local user store
- Provide outbound credentials using the netmap

The remaining configuration scenarios require Sterling External Authentication Server (EA), an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to the EA to enforce the following advanced security features:

- Authenticate an inbound certificate or user using EA
- Manage connection requirements to the outbound server using EA

Additional procedures are provided to instruct you on how to configure the following features:

- Block common exploits
- Rewrite URLs in HTML content to route inbound connections through SSP
- Define alternate nodes for failover support


Complete Scenario Worksheets

Before you begin configuring SSP for each HTTP connection scenario, gather the information on the worksheet provided with the scenario. You use this information as you configure each feature. Complete worksheets as follows:

- Provide a value for each SSP feature listed. Fields listed in the worksheet are required.
- Accept default values for fields not listed in the worksheet.
- Note the Configuration Manager field(s) where you will specify the value.

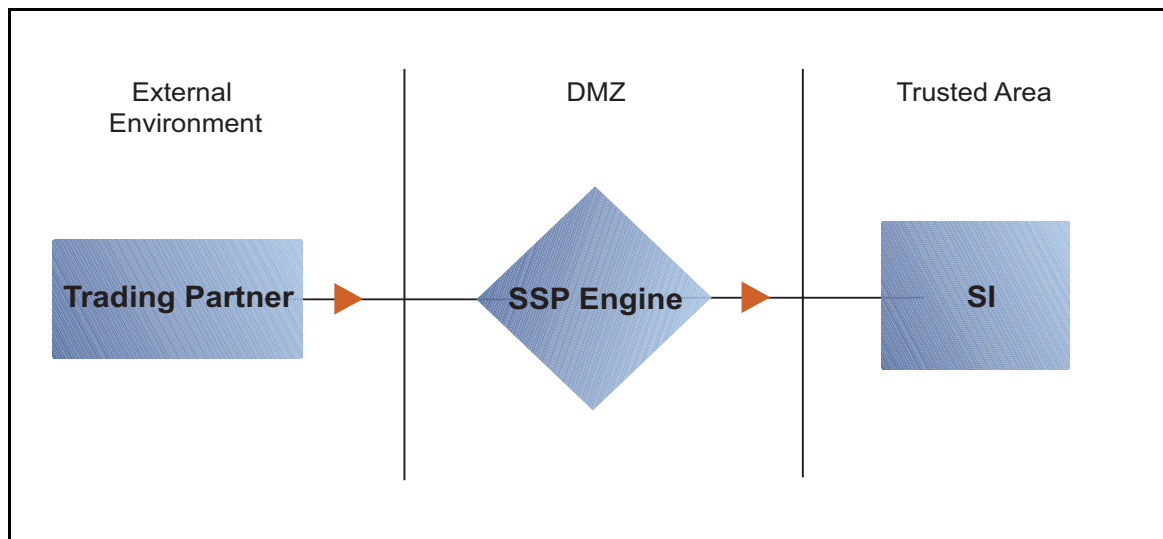
Complete and Test HTTP Configuration Scenarios

Work through the sequence of HTTP configuration scenarios in the order they are presented to add additional security features. Be sure to test each feature before you add the next feature to the configuration. Before you move SSP into production, ensure that you have configured and tested all of the security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed:  To view more information about the error, hover over the icon.

Create a Basic HTTP Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection from a trading partner to the Sterling Integrator server as shown in the following diagram. You accept default values when configuring this scenario. As a result, no authentication occurs in SSP and credentials presented by the inbound node are passed through to the Sterling Integrator server.



After you configure SSP, validate the configuration by initiating an HTTP connection from the trading partner. For more information on testing the configuration, see *Test the Inbound and Outbound HTTP Connections* on page 187.

Complete the following tasks to define a basic HTTP configuration:

- Create a policy
- Define inbound and outbound connections in a netmap
- Define an HTTP adapter

Basic HTTP Configuration Worksheet

Before you configure SSP for HTTP connections, gather the information on the Basic HTTP Configuration Worksheet. You use this information as you configure a basic HTTP connection for SSP. After you configure SSP for HTTP connections, validate the configuration by initiating an HTTP connection from the inbound node.

HTTP Policy

Create a basic policy. In a later HTTP configuration scenario, you edit this policy to add security features.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy.	_____

HTTP Netmap (Inbound and Outbound Connections)

Create a netmap that contains connection information for the nodes connecting to and from SSP: the trading partner (inbound node) and the Sterling Integrator server (outbound node). You will also associate the basic security policy you create with the inbound node.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name.	

Inbound Trading Partner Information

Inbound Node Name	Trading partner name (name to assign to inbound node definition).	
Peer Address Pattern	Host name or IP address pattern.	* (Specifying * for this value allows all inbound nodes configured on the Sterling Integrator server as trading partners to connect to the Sterling Integrator server. Use this value for testing purposes. To create a more specific node definition, see <i>Define Inbound HTTP Node Connection Definitions</i> on page 174 <i>Define Inbound HTTP Node Connection Definitions</i> .)
Policy	Name of policy you create.	This value is selected from a pull-down list.

Configuration Manager Field	Feature	Value
Outbound Sterling Integrator Server Connection		
Node Name	Outbound Sterling Integrator server node name.	
Primary Destination Address	Host name or IP address to connect to the outbound Sterling Integrator server.	_____
Primary Destination Port	Port number to connect to the outbound Sterling Integrator server.	_____

HTTP Adapter

Create an HTTP adapter that defines information necessary to establish HTTP connections to and from SSP. When you are configuring the adapter, select the basic netmap and the outbound Sterling Integrator server you define in the netmap definition.

Configuration Manager Field	Feature	Value
Adapter Name	Adapter name.	
Listen Port	Listen port to use for inbound connections.	
Netmap	Netmap to associate with the adapter.	
Standard Routing Node	Name of the outbound node corresponding to the Sterling Integrator server where inbound connections are routed.	
Engine	Engine to run on.	

Create an HTTP Policy

The HTTP policy defines how you impose controls to authenticate a trading partner trying to access an Sterling Integrator server over the public Internet.

To define a policy:

1. Click Configuration from the menu bar.
2. Click Actions > New Policy > HTTP Policy.
3. Type a Policy Name.
4. Click Save.

Create an HTTP Netmap

You define inbound connection information for your external trading partners and outbound connection information for the Sterling Integrator server that SSP connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define inbound and outbound nodes:

1. Click Configuration from the menu bar.
2. Click Actions > New Netmap > HTTP Netmap.
3. Type a Netmap Name.
4. To define an inbound node definition, click the Inbound Nodes tab and click New.
5. Specify the following values:
 - ◆ Inbound Node Name
 - ◆ Peer Address Pattern
 - ◆ Policy

Note: If you have not defined a policy, click the green plus sign to define one.

6. Click OK.
7. To define an outbound node definition, click the Outbound Nodes tab and click New.
8. Specify the following values:
 - ◆ Outbound Node Name
 - ◆ Primary Destination Address
 - ◆ Primary Destination Port
9. Click OK.
10. Click Save.

Define the HTTP Adapter Used for the Connection

An HTTP adapter definition specifies system-level communications information necessary for HTTP connections to and from SSP. You can create multiple adapter definitions.

Before you begin this procedure, create the following definitions:

A netmap to associate with the adapter.

An engine definition to associate with the adapter. Refer to *Install or Upgrade SSP on UNIX or Linux* or *Install or Upgrade SSP on Windows* for instructions.

To define an HTTP adapter:

1. Click Configuration from the menu bar.
2. Click Actions > New Adapter > HTTP Reverse Proxy.

3. Specify values for the following:
 - ◆ Adapter Name
 - ◆ Listen Port
 - ◆ Netmap
 - ◆ Standard Routing Node
 - ◆ Engine
4. Click Save.

What You Defined with the Basic HTTP Configuration Scenario

Creating connections to Sterling Integrator servers on behalf of nodes external to your trusted zone requires that you organize information about the trading partners and the Sterling Integrator server in a policy, a netmap, and an adapter definition. You created these items when you defined the Basic HTTP Configuration. The next step is testing the configuration prior to configuring additional security features. Before you test the configuration, be sure that:

The Sterling Integrator server has an active HTTP server adapter configured to listen for the port specified in the outbound node definition

The user ID and password provided by the inbound node are defined at the Sterling Integrator server

Refer to *Test the Inbound and Outbound HTTP Connections* on page 187 for information about testing the HTTP Reverse Proxy Configurations outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Variations on the Basic HTTP Configuration

After you confirm that the communications sessions you established using the basic HTTP configuration were successful, you may want to validate sessions using other types of inbound trading partner definitions before you add complexity to the security configuration. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

Inbound HTTP Trading Partner Node Definitions

You can modify the inbound trading partner node definitions as follows:

- Define a specific IP address
- Define a wildcard peer pattern
- Define an IP/subnet pattern

Define HTTP Connection Requirements Between SSP and Inbound Nodes

You define connection requirements between SSP and inbound nodes by creating inbound node definitions. Refer to your company security requirements to determine how tightly to define what parameters an inbound node must provide to allow a connection.

You can define an inbound node definition as generic or as specific as your security environment requires. In the strictest environment, you define a specific node definition that allows only one individual inbound connection to use the definition. In an environment where you trust in the inbound node connections, you can identify a pattern of IP addresses and create an inbound definition that allows all inbound connections that match the pattern to connect to SSP.

Methods of defining inbound nodes include:

Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32. SSP also supports individual host names. They must match the value returned by a reverse DNS lookup.

Define an inbound node entry that allows all nodes that match an IP/Subnet address pattern. Patterns include:

Match the first 16 bits of an IP address pattern. For example, 10.20.0.0/16 allows all IP addresses that begin with 10.20.* to connect to SSP.

Match the first 8 bits of an IP address pattern. For example, 10.0.0.0/8 allows all IP addresses that begin with 10.* to connect to SSP.

Define an inbound node entry that allows all inbound nodes that match a wildcard host name pattern. When a connection is attempted and you have defined a wildcard host name pattern definition, a reverse DNS lookup is performed on the IP address of the inbound connection. The DNS name is compared to the wildcard pattern. Wildcard patterns include:

* enables a match on any number of characters. For example, *.a.com allows a connection from b.a.com but not from a.bc.com. * allows all inbound nodes to successfully connect to SSP.

? enables a match on one character. For example, a?.com allows a connection from a.b.com but not from a.bc.com.

You can define more than one inbound node definition and use a combination of the node definition methods. However, be sure to order the definitions from most specific to least specific. When an inbound node connection is attempted, SSP compares the IP address of the inbound node to the first inbound node definition. If it matches, a connection is established. If it does not match, SSP checks the next inbound node definition until a match is found. If no match is found, the connection is terminated.

Inbound HTTP Connection Definition Worksheet

Use the following worksheet to identify the information needed to configure inbound node definitions for groups of inbound nodes that match a pattern or for specific inbound nodes.

Configuration Manager Field	Define Inbound Trading Partner Information	Value
-----------------------------	--	-------

Note: If you define a single node and definitions for multiple nodes using pattern matching, ensure that you order the definitions from most specific to least specific because SSP processes them in the order in which they are listed.

Inbound Node Name	Trading Partner Name.
Policy Name	Policy to associate with the inbound trading partner.

For a Single Node

Peer Address Pattern	<p>IP address</p> <p>Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. SSP supports host name. An example definition is a.b.com.</p> <p>A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32.</p>
----------------------	---

For Multiple IP Addresses Using IP/Subnet Pattern

Peer Address Pattern	<p>Peer Address IP/Subnet Pattern Options:</p> <ul style="list-style-type: none"> ◆ Match first 16 bits of IP address with pattern, for example, 10.20.0.0/16 matches 10.20.* ◆ Match first 8 bits of IP address with pattern, for example, 10.0.0.0/8 matches 10.*
----------------------	---

For Multiple Nodes Using Wildcard Peer Address Pattern to Validate Inbound DNS

Peer Address Pattern	<p>Wildcard Peer Address Pattern:</p> <ul style="list-style-type: none"> ◆ * enables a match on any number of characters, for example, *.a.com matches b.a.com but not a.b.com ◆ ? enables a match on any one character, for example, a?.com matches a.b.com but not a.bc.com
----------------------	---

Define Inbound HTTP Node Connection Definitions

This procedure instructs you how to modify the basic HTTP configuration to add inbound node definitions for a group of nodes with similar information, and definitions that limit access to one specific inbound node. It assumes that you have already configured an adapter. Gather a list of all inbound trading partners, including names and IP addresses.

To define inbound connection definitions:

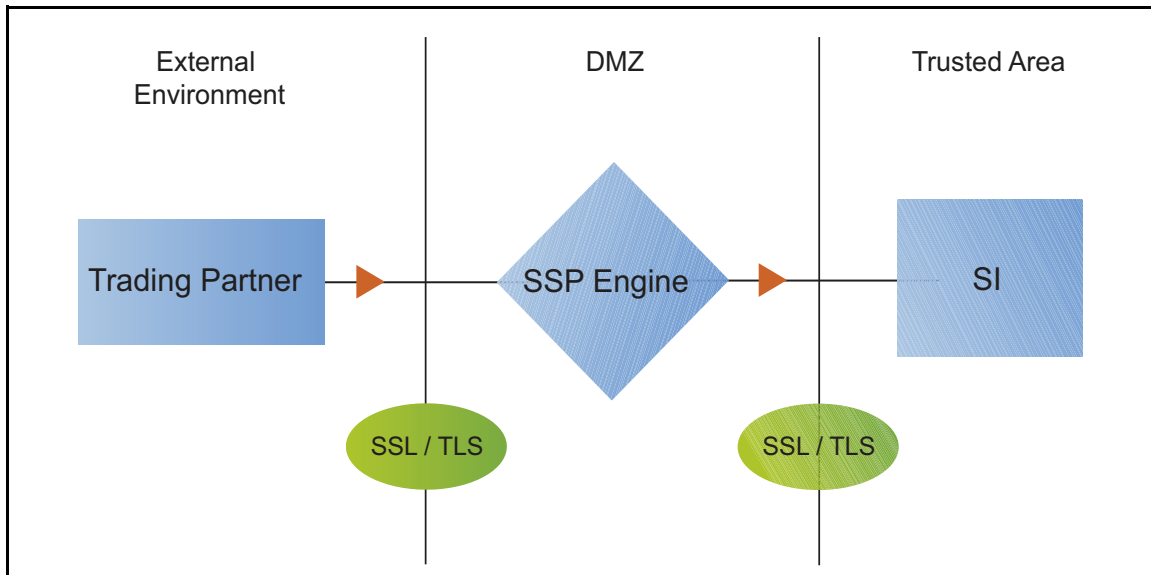
1. Identify patterns that can be used to define groups of inbound nodes.
2. Decide if you need to define a trading partner connection for any individual IP addresses.

3. Click Configuration from the menu bar.
4. Expand the Netmaps tree and select the netmap to modify.
5. Click New to add a new inbound node definition.
6. Using the information you defined on the Inbound Connection Definition Worksheet, provide the following information and click Save:
 - ◆ Inbound Node Name
 - ◆ Peer Address Pattern
 - ◆ Policy
7. Repeat step 6 for every group of connections and for every individual IP address connection you want to define.
8. If necessary, reorder the node definitions in the netmap. Order definitions from most specific to least specific since they will be evaluated in order.
 - a. Click the radio button beside the inbound node definition to move.
 - b. Click Move Up or Move Down until the node definition is in the correct order.
9. Click Save.

Establish a session initiated by an HTTP client to an Sterling Integrator server to test the configuration.

Add SSL/TLS Support for an HTTP Connection

This scenario builds on the Basic HTTP Configuration by enabling security for the inbound and outbound nodes you defined in the netmap. Following is a diagram to illustrate the addition of SSL or TLS to the inbound and the outbound node connections.



Note: Before you configure SSL or TLS support, you must check in your certificates. Refer to Manage Certificates for SSL/TLS Transactions with Trading Partners.

To add SSL/TLS support to the netmap for the inbound and outbound nodes, define the following options for the connections:

- Protocol
- Cipher suites
- Stores and certificates

To effectively configure and test this scenario:

1. Add SSL/TLS support to the inbound node definition first and establish a session initiated by an HTTP client to an Sterling Integrator server.
2. Then, add SSL/TLS support to the outbound node definition and establish a session initiated by an HTTP client to an Sterling Integrator server.

SSL/TLS Support for HTTP Worksheet

Before you add SSL/TLS support to the connection information you created in the Basic HTTP Configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Secure Inbound HTTP Connection

Select the security setting and cipher suites to be used to secure the connection. To configure client authentication, enable this option. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Inbound Node Name	Name of inbound node to add security to.	Select an inbound node definition from the list
Security Setting	Security protocol to use.	(SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Enable Client Authentication	Do you want to require that the inbound connection present its certificate for SSL or TLS client authentication?	(Yes or No)
Trust Store	If client authentication is enabled, identify the trust store where the certificate is stored.	
CA Certificates/Trusted Root	Name of CA certificate/trusted root (if client authentication is enabled).	
Key Store	The database where the keys and system certificates you want to use are stored.	
Key/System Certificate	Name of SSP system certificate presented to the inbound connection during the handshake.	
Available Cipher Suites Selected Cipher Suites	Select the ciphers to enable by moving them from the Available Ciphers to the Selected Ciphers field.	

Secure the Outbound HTTP Connection

Select the security setting and cipher suites to be used to secure the connection. Select the trusted certificate to use to validate the server certificate. If the server requires client authentication, you must specify a server certificate. If the server requires client authentication, you specify a key/system certificate.

Configuration Manager Field	Feature	Value
Outbound Node Name	Name of outbound node to add security to.	Select a node definition from the list.
Security Setting	Security protocol to use.	(SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Trust Store	The trust store where the certificate is stored.	

Configuration Manager Field	Feature	Value
CA Certificates/Trusted Root	Identify the certificate to use to secure the outbound connection.	
Key Store	Key store where the Key/System Certificate is stored.	
Key/System Certificate	System certificate used to validate the server.	
Available Ciphers Selected Ciphers	Cipher suites to enable.	

Secure the Inbound HTTP Connection Using the SSL or TLS Protocol

The first step in strengthening security is to secure the communications channel. This procedure describes how to enable the TLS or SSL protocol for the inbound connection to authenticate SSP to the trading partner initiating the connection. To require that SSP authenticate the inbound node, enable client authentication.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP Cert Stores.

To enable the TLS or SSL protocol:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Click the Inbound Nodes tab.
4. Select an inbound node to modify, and click Edit.
5. Click the Security tab, and then click Secure Connection to enable security.
6. Select values for the following:
 - ◆ Security Setting
 - ◆ Key Store
 - ◆ Key/System Certificate
 - ◆ Available Cipher Suites
 - ◆ Selected Cipher Suites
7. To enable client authentication:
 - a. Click Enable Client Authentication.
 - b. Select the trust store where the CA certificate or trusted root certificate is stored.
 - c. Select the CA Certificates/Trusted Root certificate to use.

Note: Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

8. Click OK.
9. Click Save.

Establish a session initiated by an HTTP client to an Sterling Integrator server to test the configuration.

Secure the Outbound HTTP Connection Using the SSL or TLS Protocol

If the Sterling Integrator server has enabled the use of SSL or TLS to secure the connection, you must enable TLS or SSL protocol in the SSP outbound node configuration. This procedure describes how to enable the TLS or SSL protocol to authenticate the Sterling Integrator server to SSP when establishing an outbound connection.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP cert stores.

To enable the TLS or SSL protocol:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Click the Outbound Nodes tab.
4. Select an outbound node to modify, and click Edit.
5. Click the Security tab, and then click Secure Connection to enable security.
6. Select the following security options for the node:
 - ◆ Security Setting
 - ◆ Trust Store
 - ◆ CA Certificate/Trusted Root

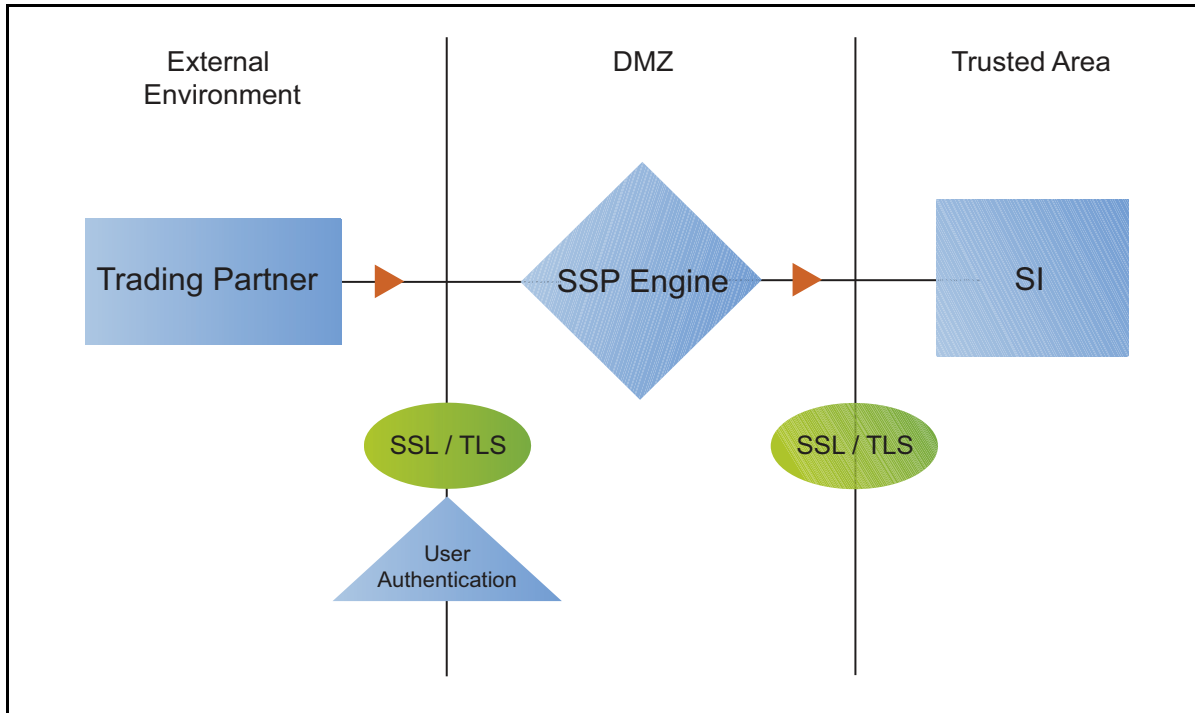
Note: Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

- ◆ Available Ciphers
 - ◆ Selected Ciphers
7. If the Sterling Integrator server requires client authentication, select the key store and key/system certificate to present to the Sterling Integrator server during the SSL/TLS handshake.
 8. Click OK.
 9. Click Save.

Establish a session initiated by an HTTP client to an Sterling Integrator server to test the configuration.

Add Local User Authentication to the HTTP Connection

This scenario builds on the Basic HTTP Configuration by adding user authentication to the inbound connection using information defined in the local user store. Following is an illustration of the security options enabled for this scenario:



The user ID and password presented by the inbound node are authenticated against the information stored in the local user store. The values must match before a connection is established. You must add this information to the local user store before you can test this scenario.

Adding user authentication to the inbound connection defined in the Basic HTTP Configuration involves enabling user authentication and specifying information about the trading partner.

After you configure user authentication using the local user store information, validate the configuration by establishing a session initiated by an HTTP client to an Sterling Integrator server.

HTTP Inbound Connection (Local User Authentication) Worksheet

Before you add user authentication to the inbound connection you created in the Basic HTTP Configuration scenario, gather the information on the HTTP Inbound Connection (Local User Authentication) Worksheet. Use this information as you configure user authentication for the inbound connection.

In this scenario, you edit the policy you created in the HTTP Basic Configuration scenario and enable user authentication. You also add a user ID and password for the trading partner to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	
User Authentication	Method to use to authenticate the inbound node.	Through local user store
User Store	Name of the user store you create.	
User Name	Name of the user you define in the User Store.	
Password Confirm Password	The password value to use to validate the inbound connection.	

Enable Local User Authentication to an HTTP Inbound Connection

You can strengthen the security of inbound connections by enabling local user authentication. This procedure describes how to configure the use of the local user store to validate an inbound connection.

Note: Check the netmap to ensure that the policy you edit is associated with the inbound nodes you want to authenticate.

To add user authentication for an inbound connection:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select the policy you created in the basic configuration.
3. Click the Advanced tab.
4. Enable the User Authentication Through Local User Store option.
5. Click Save.

Add Credentials to the Local User Store for an HTTP Connection

If you enable user authentication through the local user store, you have to add user information to the local user store to be validated by SSP during an inbound HTTP client connection.

Before you begin this procedure:

Enable user authentication for the inbound connection.

Ensure that the engine is configured to use the user store that contains the user credentials.

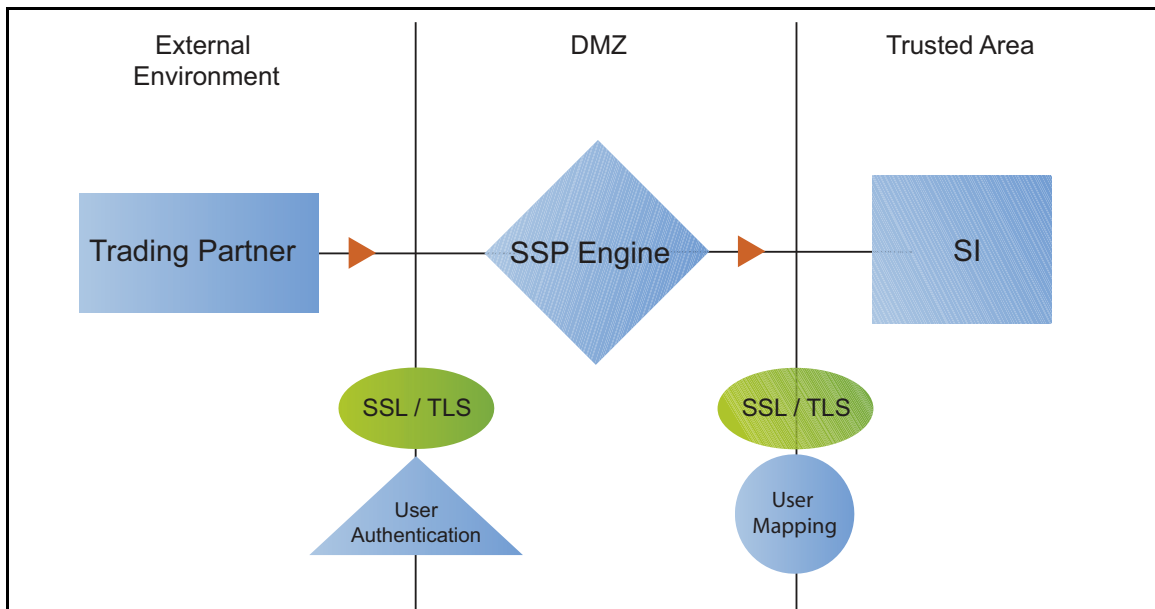
To add user information to the local user store:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree and select a user store to modify.
3. From the User Store Configuration panel, click New.
4. Specify values for the following:
 - ◆ User Name
 - ◆ Password
 - ◆ Confirm Password
5. Click OK.
6. Click Save.

Establish a session initiated by an HTTP client to an Sterling Integrator server to test the configuration.

Provide Credentials to the Outbound HTTP Node Using the Netmap

This scenario builds on the Basic HTTP Configuration by enabling the use of user credentials from the netmap to connect to the outbound Sterling Integrator connection. Following is an illustration of the security features supported in this scenario:



If you configure user mapping using the netmap, an inbound trading partner connects to SSP and provides one set of credentials. Its credentials are replaced with credentials stored in the netmap. The replacement credentials are then used to connect to the outbound secure server. This method uses SSP security features to prevent trading partners from knowing the credentials used to connect

to the outbound Sterling Integrator server. The outbound Sterling Integrator server must have a user definition that accepts the user ID and password provided.

After you configure the environment to use credentials defined in the netmap, test the configuration by establishing a session initiated by an HTTP client to an Sterling Integrator server. Refer to *Test the Inbound and Outbound HTTP Connections* on page 187 for more information on testing the configuration described in this scenario.

Connect to the Outbound HTTP Server Using Credentials from the Netmap Worksheet

In this scenario, edit the netmap and the policy you created in the Basic HTTP Configuration to provide user credentials stored in SSP to connect to the outbound Sterling Integrator connection.

Collect the following information so you can match the SSP configuration with the Sterling Integrator server configuration. Use the information on this worksheet as you edit the outbound node definition, and be sure to select the netmap and policy you created in the Basic HTTP Configuration.

Configuration Manager Field	Feature	Value
User ID	User ID used to connect to the Sterling Integrator server. (Must also be defined at the Sterling Integrator server)	
Password	Password to connect to the server. (Must also be defined at the Sterling Integrator server)	

Configure Name and Password to Connect to the Outbound HTTP Server in the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to store the user ID and password to connect to the outbound Sterling Integrator server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and SSP uses information stored in the netmap to connect to the outbound HTTP server.

Before you configure this option:

- Ensure the user ID and password are defined on the Sterling Integrator server
- Obtain the user ID and password

To configure validation for the outbound connection using credentials stored in the netmap:

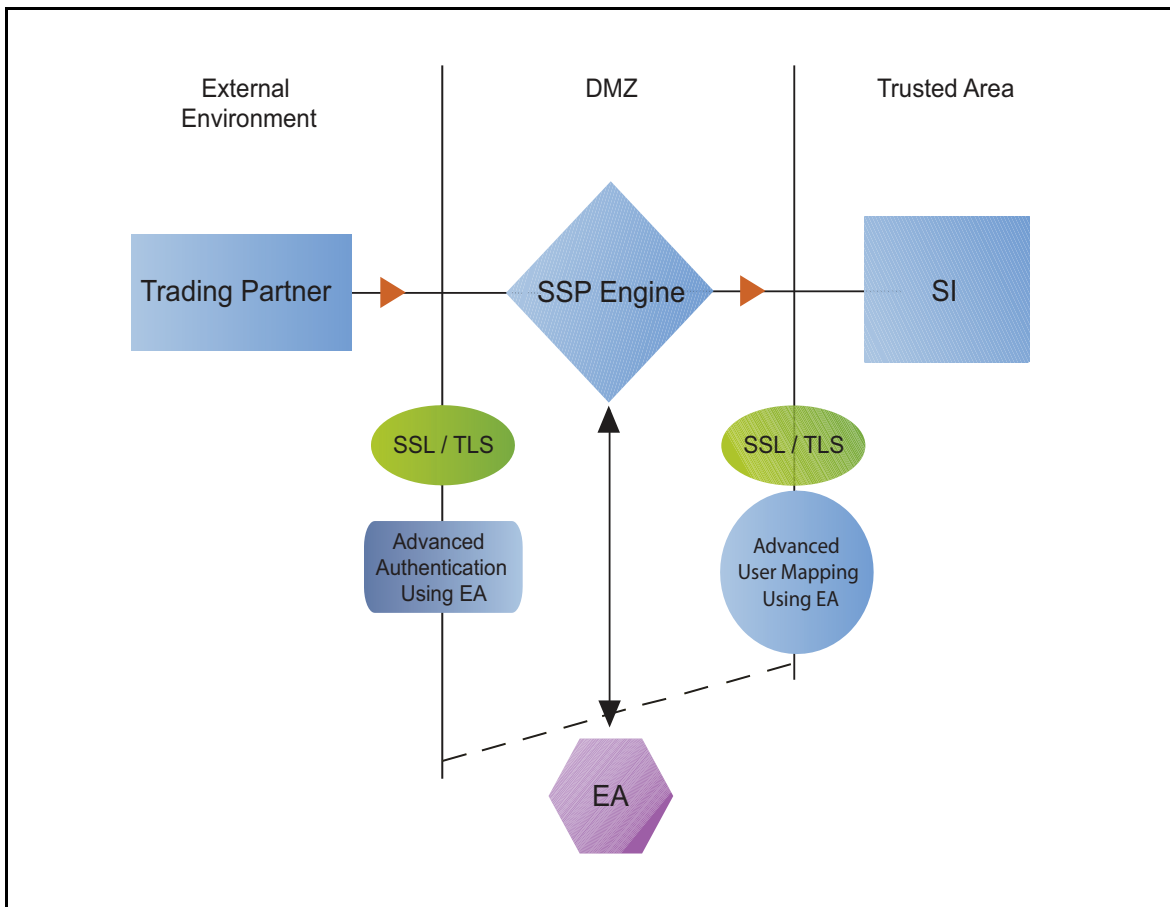
1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Click the Outbound Nodes tab.
4. Select the outbound node to modify and click Edit.
5. Click the Advanced tab.

6. Type the following values to be used to connect to the Sterling Integrator server:
 - ◆ User ID
 - ◆ Password
7. Click OK.
8. Click Save.
9. Expand the Policies tree and select the policy to modify.
10. On the Policy Configuration panel, click the Advanced tab.
11. From the User Mapping: Internal User ID list, select From Netmap.
12. Click Save.

Test the configuration to ensure that the updated configuration is working.

Strengthen Authentication for an HTTP Connection Using EA

To provide a more advanced method of securing the inbound or the outbound connection, use EA. Use EA to authenticate certificate information or user credentials presented by the inbound node or to perform user ID and password mapping for the internal credentials. The following illustrates the security features enabled in this scenario.



Authenticate an Inbound HTTP Certificate or User Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. Following are some of the options EA can perform:

- Validate certificates, including dates and signatures
- Verify the presence of X.509 v3 extensions
- Enforce minimum key length requirements
- Check certificates against certificate revocation lists (CRLs)
- Perform LDAP queries

The EA definition determines which options are enabled.

Manage Connection Requirements to the Outbound HTTP Server Using EA

For a higher level of security when connecting to the outbound server, use information stored in an LDAP database to connect to the outbound server. To use information in an LDAP database, you configure EA. EA can map a user ID and password provided by an inbound connection to a user ID and password that is not exposed to the external node.

Authenticate an Inbound HTTP Certificate or User Using EA Worksheet

Use the following worksheet to specify the information needed to authenticate a trading partner with information in EA. Update the policy you created in the Basic HTTP Configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the inbound certificate?	(Yes or No)
Certificate Authentication - External Authentication Profile	If yes, identify the EA certificate validation definition.	
User Authentication - Through External Authentication	Will you validate user information?	(Yes or No)
User Authentication - External Authentication Profile	If yes, identify the EA user validation definition.	

Authenticate the Inbound HTTP Node Using EA

To authenticate certificate information or user information about the inbound node against information stored in an LDAP database, you must configure EA. After you configure EA to enable certificate validation or user authentication, use this procedure to configure SSP to use the authentication method you defined in EA.

Before you configure SSP to use EA to authenticate an inbound node, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

- The policy associated with the inbound node has enabled client authentication.

The public keys for SSP have been sent to the EA server and imported into the EA keystore. The EA server connection has been configured in SSP. Refer to *Configure SSP for Sterling External Authentication Server (EA)*.

To configure authentication of an inbound node using EA:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy to modify.
3. On the HTTP Policy Configuration panel, click the Advanced tab.
4. To validate the certificate presented by the inbound node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and identify the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.
5. To validate a user from EA:
 - a. Enable User Authentication Through External Authentication field.
 - b. Type the name of the definition you defined in EA in the User Authentication External Authentication Profile field.
 - c. Deselect the Through Local User Store option.
 - d. Select From External Authentication in the User Mapping:Internal User ID field.
6. Click Save.

You can now associate this policy with the inbound node on which you want to perform user authentication using EA.

Connect to the Outbound HTTP Server Using EA Worksheet

Use this worksheet to identify information required to configure a stronger outbound connection using information in an LDAP database:

Configuration Manager Field	Feature	Value
User Certification Through External Authentication	Will you validate user information against LDAP?	Yes
External Authentication Profile	If yes, identify the EA user validation definition.	
Destination Service Name	Identify the destination server that can be accessed by the outbound node, when using EA to map a user ID and password. Valid values are 1-255 alphanumeric characters and certain special characters. The following characters are not allowed: ! @ # % ^ * () + ? , < > { } [] ; " ' .	

Connect to the Outbound HTTP Server Using Information Stored in LDAP

If you store user credentials in an LDAP database, use this procedure to configure SSP to use these credentials to connect to the secure outbound server.

Before you configure this option:

Configure a definition in EA and obtain the name of the EA definition.

Configure the EA server to allow connections from SSP.

Ensure that the policy associated with the inbound node has enabled client authentication.

Ensure that the public keys for SSP have been sent to the EA server and imported into the EA trust store.

To configure the use of credentials from an LDAP database:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Enable the User Authentication Through External Authentication field.
5. Type the name of the definition you defined in EA in the User Authentication External Authentication Profile field.
6. Deselect the Local User Store option.
7. Select User ID/Password from External Authentication in the User Mapping:Internal User ID field.
8. Click Save.
9. Expand the Netmaps tree and click the HTTP netmap to modify.
10. On the HTTP Netmap panel, click the Outbound Nodes tab.
11. Select the node to edit and click Edit.
12. Click the Advanced tab.
13. Identify the destination service name to use to connect the outbound node when using EA in the Destination Service Name field.
14. Click OK and click Save.

Test the Inbound and Outbound HTTP Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between an HTTP client and the engine, initiate a session from the engine to the Sterling Integrator server in the trusted zone, and review the SSP audit log for the results.

Note: Configuration files must be available on the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

Establish an HTTP session initiated by a trading partner using an HTTP client

Initiate an outbound session to an Sterling Integrator server on behalf of the HTTP client connection

To verify the communications sessions:

1. Make sure the engine is running.
2. Initiate an HTTP client session to the Sterling Integrator server in your trusted zone.
3. View the Inbound Node Log and the Outbound Node Log.
4. Confirm that the data transfer was successful, as shown in the following sample audit log output.

Sample Inbound Node Log

```
11 Sep 2010 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1 SNAME=user.company.com
SIP=10.20.200.100 SPORT=40134 SSP104I Session: 1 - Session Proceeding after Node
match: Any
11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1 SNAME=user.company.com
SIP=10.20.200.100 SPORT=40134 DNAME=dname.company.com DIP=10.20.246.42 DPORT=10054
SUID=admin DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

Sample Outbound Node Log

```
11 Sep 2010 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1 SNAME=user.company.com
SIP=10.20.200.100 SPORT=40134 SSP104I Session: 1 - Session Proceeding after Node
match: Any
11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1 SNAME=user.company.com
SIP=10.20.200.100 SPORT=40134 DNAME=dname.company.com DIP=10.20.200.40 DPORT=10054
SUID=admin DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

If your session was unsuccessful, review the log information to determine the likely cause of failure and the corrective action to take.

Additional HTTP Configuration Options

Additional HTTP configuration options are available for the following features:

- Block common exploits
- Change the commands that are allowed or blocked
- Rewrite URLs in HTML content to route inbound connections through proxy
- Define alternate nodes for failover support

Block Common Exploits

When a connection from an inbound HTTP node to SSP is attempted, you can enable the ability to scan the URL requested and look for commonly occurring exploits.

If block common exploits is enabled, HTTP requests cannot contain the following characters or strings, which are commonly used on attacks on HTTP servers:

```
|
.
\
<?
\u0000
```

You can change the values that are blocked in the policy.

To enable the capability to block common exploits:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Enable the Block Common Exploit Strings field.
5. Click Save.

Change the Values to Block in a URL String

When a connection from an inbound HTTP node to SSP is attempted, you can enable the ability to scan the URL requested and look for commonly occurring exploits.

If block common exploits is enabled, HTTP requests cannot contain the characters or strings, identified in the graphic above. You can change the characters that are blocked. To change the blocked strings:

1. Click Configuration from the menu bar.
1. Expand the Policy tree and click the HTTP policy to modify.
2. On the HTTP Policy Configuration panel, click the Advanced tab.
3. Add a new value, or delete or edit an existing value, by changing the values in the Block Common Exploit Strings field.
4. Click Save.

Map a URL in HTML Content from the Outbound Server

HTTP Reverse Proxy HTML rewriting allows you to replace the URL links submitted by an HTTP client to the HTTP server with URL links to SSP. If the HTTP server has web pages with links to other web pages on the same host, you must map all URL connections in order for the links to work.

Before you configure this option, create a netmap definition. Create an outbound node definition for each URL containing a host and port.

Configure HTTP Rewrite to Support the Sterling Integrator Dashboard

To communicate with the Sterling Integrator dashboard, two connections must be established to the outbound Sterling Integrator server: one connection to the Sterling Integrator base port and one to the Sterling Integrator base port + 33.

To configure this environment:

1. Define two outbound nodes in the netmap: Definition 1 configures a connection to the Sterling Integrator host and base port. Definition 2 configures a connection to the Sterling Integrator host and base port + 33.
2. Add mapping values to the netmap definition for both URL connections.
3. Configure two HTTP Reverse Proxy adapters: one to route connections to the Sterling Integrator host and base port (Definition 1) and another to the Sterling Integrator host and base port + 33 (Definition 2). Use the same netmap with both adapter definitions. For each adapter, select a different outbound node to route connections to in the Standard Routing Node field.

For example, assume SSP is installed and running on the host, proxy_host and HTTP Reverse Proxy adapter 1 is configured to listen on the port, adapter1_port. It uses the outbound node defined as Sterling Integrator base port on a host called si_host. HTTP Reverse Proxy adapter 2 listens on the port, adapter2_port and uses an outbound node defined as Sterling Integrator baseport + 33 (the dashboard default port).

To configure this environment, define the following URL rewrite values in the netmap definition:

Server URL	Proxy URL
http://<si_host>:<baseport>	http://<proxy_host>:<adapter1_port>
http://<si_host>:<baseport+33>	http://<proxy_host>:<adapter2_port>

Configure HTML Rewrite

To configure HTML rewrite:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Make sure you have two outbound node definitions: one for the Sterling Integrator server and its base port and another for Sterling Integrator base port + 33. To define an outbound node definition:
 - a. Click the Outbound Nodes tab and click New.
 - b. Specify the following values:
 - Outbound Node Name
 - Primary Destination Address
 - Primary Destination Port
4. Click OK.

5. On the HTTP Netmap Nodes panel, click the HTML Rewrite tab.
6. Click New.
7. Enable the Support HTML Rewrite field.
8. Type the URL path for the outbound server in the Server URL field.
9. Type the URL path for the proxy in the Proxy URL field. Refer to *Configure HTTP Rewrite to Support the Sterling Integrator Dashboard* on page 190 and the table of values for instructions on the URL values to define for the Sterling Integrator dashboard.
10. Click Save.
11. Repeat steps 3 through 10 for all HTML Rewrite options you want to configure.
12. To reorder the HTML rewrite definitions:
 - a. Click the radio button beside the URL routing definition to reorder.
 - b. Click Move Up or Move Down until the item is in the correct order.
13. Click Save.
14. Expand the Adapters tree and click the adapter to modify.
15. Enable Support HTML Rewrite.
16. Click Save.

Test the configuration to ensure that the HTML rewrite is configured correctly.

Note: If the following message is written to the `secureproxy.log` file, correct your URL definition:

HTML Rewrite proxy URL Map entry is not a valid URI.

Define Alternate Nodes for Failover Support for an Outbound HTTP Connection

If you are using standard routing to connect to an Sterling Integrator server in the secure zone, you define a primary Sterling Integrator server to connect to in the adapter. For each outbound node definition, you can identify up to three alternate outbound nodes to connect to when the primary Sterling Integrator server is not available.

Two methods of configuring alternate Sterling Integrator server routing are available.

Select a previously defined outbound node from the drop-down list on the Advanced tab. To configure this method, you first configure an outbound node definition in the netmap for each alternate Sterling Integrator server you want to use. Each connection uses the security and other settings defined for that outbound node in the netmap.

Select IP address/port from the drop-down list on the Advanced tab and enter values for the IP address and port. If you use this method you do not have to define the alternate outbound nodes in the netmap, and each alternate connection shares the security and other settings defined in the primary node definition.

If you configure alternate Sterling Integrator server definitions in the outbound node definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2. If this connection is unsuccessful, SSP

tries to connect to the third alternate, Node 3. If the connection to this node is unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Click the Outbound Nodes tab.
4. Select the outbound node to modify and click Edit.
5. Click the Advanced tab.
6. Do one of the following:
 - ◆ To identify an alternate node defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.
 - ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:
 - a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.
 - b. Provide the IP Address and Port number for the alternate outbound node
7. Click OK.
8. Click Save.

SFTP Reverse Proxy Configuration

The SFTP configuration scenarios describe how to configure SFTP protocol connections to and from the engine.

Note: Configuration information must be available on the engine before communication sessions with Sterling Integrator can be established.

Organization of the SFTP Configuration Scenarios

The first scenario instructs you on how to configure a basic configuration. Each successive scenario adds another security feature to the basic configuration. After adding a security feature, test the connection to ensure that you have correctly configured it. You determine your security needs and configure the security features applicable for your environment.

The following scenarios help you configure and test SSP for SFTP protocol connections to the SFTP server:

- Create a basic configuration
- Perform user authentication using the local user store
- Provide user mapping using the netmap

The remaining scenarios require EA, an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to the EA to enforce the following advanced security features:

- Authenticate an inbound user using EA
- Manage connection requirements to the outbound server using EA

Additional procedures instruct you how to define alternate nodes for failover support.


Complete SFTP Scenario Worksheets

Before you configure SSP for SFTP, gather the information on the worksheet provided with the scenario. You use this information as you configure each feature. Complete worksheets as follows:

- Provide a value for each SSP feature listed. Fields listed in the worksheet are required.
- Accept default values for fields not listed in the worksheet.
- The worksheet identifies the Configuration Manager field where you specify each value.

Complete and Test SFTP Configuration Scenarios

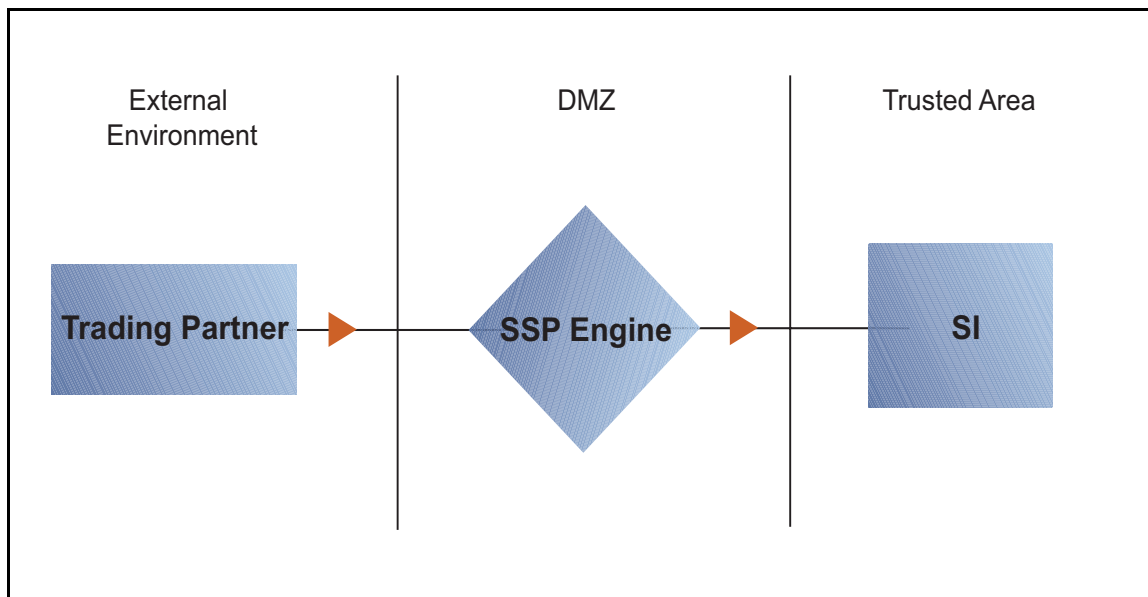
Work through the sequence of SFTP configuration scenarios in the order in which they are presented to add and test security features. Be sure to test each feature before you add the next feature to the configuration. Before you move SSP into production, ensure that you have configured and tested all of the security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed:  To view more information about the error, hover over the icon.

Create a Basic SFTP Configuration

This scenario contains all the information and tools to configure SSP to establish a basic connection from a trading partner to the SFTP server as shown in the following diagram. You are configuring the minimum requirements to allow you to test the connections and ensure that communications sessions can be established between the inbound node and SSP, and to the outbound SFTP node. The basic configuration requires that SSP present its key to the inbound node for authentication and that the SFTP server present its key to SSP for authentication. It does not configure user authentication. After you create and test the basic SFTP configuration and all connections are working, you then add user authentication.

You accept default values when configuring this scenario. As a result, user credentials presented by the inbound node are used to connect to the outbound SFTP server.



After you configure the basic SFTP configuration, validate it by initiating an SFTP connection from the trading partner. For more information on testing the configuration, see *Test the Inbound and Outbound Connections* on page 210 *Test the Inbound and Outbound Connections*.

Complete the following tasks to define a basic SFTP configuration:

- Create a policy
- Define inbound and outbound connections in a netmap
- Define an SFTP adapter

Basic SFTP Configuration Worksheet

Before you configure SSP for SFTP connections, gather the information on the Basic SFTP Configuration Worksheet. You use this information as you configure a basic SFTP connection for

SSP. After you configure SSP for SFTP connections, validate the configuration by initiating an SFTP connection from the inbound node.

SFTP Policy

Create a basic policy. The default authentication method is password authentication. However, the password is not authenticated in the basic configuration because you do not select an authentication mechanism. Instead, it is passed through to the outbound node for authentication. In a later SFTP configuration scenario, you add the configuration information needed to authenticate an inbound node.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy.	

SFTP Netmap (Inbound and Outbound Connections)

Create a netmap that contains connection information for the nodes connecting to and from SSP: the trading partner (inbound node) and the Sterling Integrator SFTP server (outbound node). For the outbound node, you must identify the host name and IP address to connect to the node as well as the known host key to use for server authentication and the ciphers or message authentication codes (MACs) to use to encrypt the data. You also associate the basic policy you create with the inbound node.

Note: You must have SSH keys to authenticate SSP to the inbound node (local host keys) and to authenticate the outbound SFTP server to SSP (known host keys). Create a key store for the keys and check the keys into the key store. Refer to *Manage Local Host Key Stores and Keys* on page 292 for instructions on creating a local host key store and add a key to the key store. Refer to *Manage Known Host Key Stores and Keys* on page 297 for instructions on creating the known host key store and importing the key.

If SSP is required by the SFTP server to present its user key for authentication, you must have SSH keys for the local user for this authentication exchange. Refer to *Manage Local User Key Stores and Keys* for instructions on creating the local host key store and importing the key.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name.	
Inbound Trading Partner Information		
Inbound Node Name	Trading partner name (name to assign to inbound node definition).	(No spaces allowed.)

Configuration Manager Field	Feature	Value
Peer Address Pattern	Host name/IP address pattern.	* (Specifying * for this value allows all inbound nodes configured on the SFTP server as trading partners to connect to the SFTP server. To define a more specific node definition, see <i>Define SFTP Connection Requirements Between SSP and Inbound Nodes</i> on page 200Define SFTP Connection Requirements Between SSP and Inbound Nodes.)
Policy	Name of policy you create. (Select it from the pull-down list.)	

Outbound SFTP Server Connection

Outbound Node Name	Outbound SFTP server node name.	
Primary Destination Address	Host name/IP address of SFTP server.	
Primary Destination Port	Port number to connect to SFTP server.	
Known Host Key Store	Name of the key store where the known host key is stored.	
Known Host Key	Location and name of the public key presented to SSP by the outbound SFTP server during authentication.	

SFTP Adapter

Create an SFTP adapter that defines information necessary to establish SFTP connections to and from SSP. When you configure the adapter, select the basic netmap and outbound SFTP server in the netmap definition and the local host key that SSP presents to its clients.

Configuration Manager Field	Feature	Value
Adapter Name	Adapter name.	
Listen Port	Listen port to use for inbound connections.	
Netmap	Netmap to associate with the adapter.	
Standard Routing Node	Name of the outbound node corresponding to the Sterling Integrator server where inbound connections are routed.	

Configuration Manager Field	Feature	Value
Engine	Engine to run on.	
Startup Mode	How the adapter is started. auto starts the adapter as soon as it is pushed to the engine. manual requires that the adapter be manually started.	
Local Host Key Store	Name of the key store where the local host key is stored.	
Local Host Key	Location and name of the private part of the key presented by SSP to the inbound connection during authentication.	
Available Cipher Suites	Cipher suites to enable.	
Selected Cipher Suites	(Be sure to match the configuration of the SFTP client.)	
Available Cipher Suites	Cipher suites to enable.	
Selected Cipher Suites	(Be sure to match the configuration of the SFTP client.)	
Available Key Exchange	Key exchange to enable.	
Selected Key Exchange	(Be sure to match the configuration of the SFTP client.)	

Create an SFTP Policy

The SFTP policy defines how you impose controls to authenticate a trading partner trying to access an SFTP server over the public Internet. The basic policy does not enable any security features. You add user authentication to the policy definition in later scenarios.

To define a policy:

1. Click Configuration from the menu bar.
2. Click Actions > New Policy > SFTP Policy.
3. Specify a name for the policy in the Policy Name field.
4. Click Save.

Create an SFTP Netmap

You define inbound connection information for your external trading partners and outbound connection information for the SFTP server SSP connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

The SFTP protocol requires that the server authenticate itself to the client.

Inbound connection—Server authentication for the inbound connection requires that SSP use its private key to verify its identity to the inbound connection. Before you can configure

authentication of SSP, you must configure a local host key store and add the private key to the local host key store. You must also send the public key to the inbound trading partner. Refer to *Manage Local Host Key Stores and Keys* on page 292 for instructions. The keys used to authenticate SSP to the inbound node connection are configured in the adapter definition.

Outbound connection—Server authentication for the outbound connection requires that the SFTP server present its public key to SSP. SSP must use the public key to validate the server connection. Before you can configure authentication of the SFTP server, you must configure a known host key store and add the public key received from the SFTP server to this key store. Refer to *Manage Known Host Key Stores and Keys* on page 297 for instructions.

For authentication of the SFTP server connection, you must determine what ciphers are allowed for encryption and what MACs are allowed for message integrity protection. These MACs and ciphers must also include the required settings from the inbound nodes, the outbound node, and all keys checked into the key stores. You also determine the order of preference for both the ciphers and the MACs. Communicate with the SFTP server administrator to ensure that your configuration matches the SFTP server configuration.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define inbound and outbound nodes:

1. Click Configuration from the menu bar.
2. Click Actions > New Netmap > SFTP Netmap.
3. Type a name for the netmap in the Netmap Name field.
4. To define an inbound node definition:
 - a. Click New.
 - b. Specify the following values:
 - Inbound Node Name
 - Peer Address Pattern
 - Policy
 - c. Click OK.
5. To define an outbound node definition:
 - a. Click the Outbound Nodes tab and click New.
 - b. Specify the following values:
 - Outbound Node Name
 - Primary Destination Address
 - Primary Destination Port
 - Known Host Key Store
 - Known Host Key
 - c. Click the Security tab.
 - d. Specify the following values:
 - Available Cipher Suites
 - Available MAC Suites

- Available Key Exchange
 - e. If necessary, reorder the selected cipher suites, MAC suites, and key exchanges.
 - f. Click Ok.
6. Click Save.

Define the Adapter for the SFTP Connection

An SFTP adapter definition specifies both the system-level communications information necessary to establish SFTP connections to and from SSP and the local host key used to validate SSP to an inbound connection. Because the SFTP protocol requires that SSP present its key to the inbound node for authentication, you must configure the adapter with the local host key store and the local host key to present to the inbound connection. Before you can configure the adapter, create a local host key store and a local host key. Refer to *Manage Local Host Key Stores and Keys* on page 292 for instructions.

You must also determine what ciphers are allowed for encryption and what MACs are allowed for message integrity protection, as well as the order of preference for both the ciphers and the MACs. Communicate with the administrator of the inbound node to ensure that your configurations match.

You can create multiple adapter definitions.

Before you begin this procedure, create the following definitions:

A netmap to associate with the adapter

An engine definition to associate with the adapter. Refer to *Install or Upgrade SSP on UNIX or Linux* on page 55 or *Install or Upgrade SSP on Windows* on page 67 for instructions.

To define an SFTP adapter:

1. Click Configuration from the menu bar.
2. Click Actions > New Adapter > SFTP Reverse Proxy.
3. Specify values for the following:
 - ◆ Adapter Name
 - ◆ Listen Port
 - ◆ Netmap
 - ◆ Standard Routing Node
 - ◆ Engine
 - ◆ Local Host Key Store
 - ◆ Local Host Key
4. Click the Security tab.
5. Specify values for the following fields:
 - ◆ Available Cipher Suites
 - ◆ Available MAC Suites
 - ◆ Available Key Exchange

6. If necessary, reorder the selected cipher suites, MAC suites, or key exchange algorithms.

Note: If you change one of the following values, you must restart the adapter before the change takes effect: listen port, local host key, selected cipher suites, selected MAC suites, key exchange, compression, maximum sessions, session timeout, inbound perimeter server, outbound perimeter server, or external authentication perimeter server.

7. Click Save.

What You Defined with the Basic SFTP Configuration Scenario

Creating secure connections to SFTP servers on behalf of nodes external to your trusted zone requires that you organize information about the trading partners and the SFTP server in a policy, a netmap, and an adapter definition. You created these items when you defined the Basic SFTP Configuration. Be sure to test the Basic SFTP Configuration before you configure additional security features. Refer to *Test the Inbound and Outbound Connections* on page 210 for information about testing the SFTP reverse proxy configurations outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify these items to configure more complex authentication measures.

Variations on the Basic SFTP Configuration

After you confirm that the communications sessions you established using the basic SFTP configuration were successful, you may want to validate sessions using other types of inbound trading partner definitions before adding complexity to the security configuration. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

Inbound Trading Partner Node Definitions

You can modify the inbound trading partner node definitions as follows:

- Define a specific IP address
- Define a wildcard peer pattern
- Define an IP/subnet pattern

Define SFTP Connection Requirements Between SSP and Inbound Nodes

You define connection requirements between SSP and inbound nodes by defining inbound node definitions. Refer to your company security requirements to determine how tightly to define the parameters that an inbound node must provide to allow a connection.

You can create inbound node definitions to allow only one individual inbound connection, or you can identify a pattern of IP addresses and create an inbound definition to allow inbound connections matching the pattern to connect to SSP. Methods of defining inbound nodes are as follows:

- Create an entry for an individual inbound node and define the inbound node IP address to connect to SSP. Only connections from that IP address are allowed. A single IP Address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32. SSP also supports individual host names. They must match the value returned by a reverse DNS lookup.

Create an inbound node entry that allows all nodes that match an IP/Subnet address pattern. Patterns include:

Match the first 16 bits of an IP address pattern. For example, 10.20.0.0/16 allows all IP addresses that begin with 10.20.* to connect to SSP.

Match the first 8 bits of an IP address pattern. For example, 10.0.0.0/8 allows all IP addresses that begin with 10.* to connect to SSP.

Define an inbound node entry that allows all inbound nodes that match a wildcard host name pattern. When a connection is attempted and you have defined a wildcard host name pattern definition, a reverse DNS lookup is performed on the IP address of the inbound connection. The DNS name is compared to the wildcard pattern. Wildcard patterns include:

Asterisk (*) enables a match on any number of characters. For example, *.a.com allows a connection from b.a.com but not from a.bc.com. Using only the * allows all inbound nodes to successfully connect to SSP.

Question mark (?) enables a match on one character. For example, a?.com allows a connection from a.b.com but not from a.bc.com.

You can define more than one inbound node definition and use a combination of the node definition methods. However, be sure to order the definitions from most specific to least specific. When an inbound node connection is attempted, SSP compares the IP address of the inbound node to the first inbound node definition. If it matches, a connection is established. If it does not match, SSP checks the next inbound node definition until a match is found. If no match is found, the connection is terminated.

Inbound SFTP Connection Definition - Worksheet

Use the following worksheet to identify the information needed to configure inbound node definitions specific inbound nodes or groups of inbound nodes that match a pattern.

Configuration Manager Field	Define Inbound Trading Partner Information	Value
Note: If you define a single node and definitions for multiple nodes using pattern matching, order the definitions from most specific to least specific. SSP processes them in the order in which they are listed.		
Inbound Node Name	Trading Partner Name.	_____
Policy	Policy to associate with the inbound trading partner.	_____
For a Single Node		
Peer Address Pattern	IP address/32 or hostname. Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. SSP supports host name. An example definition is a.b.com. A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32.	

Configuration Manager Field	Define Inbound Trading Partner Information	Value
For Multiple IP Addresses Using IP/Subnet Pattern		
Peer Address Pattern	Peer Address IP/Subnet Pattern Options.	
For Multiple Nodes Using Wildcard Peer Address Pattern to Validate Inbound DNS		
Peer Address Pattern	Wildcard Peer Address Pattern.	

Define Inbound Node Connection Definitions

This procedure instructs you how to modify the basic SFTP configuration to add inbound node definitions for a group of nodes with similar information, and definitions that limit access to one specific inbound node. It assumes that you have already configured an adapter. Gather a list of all inbound trading partners, including names and IP addresses.

To define inbound connection definitions:

1. Identify patterns that can be used to define groups of inbound nodes.
2. Decide if you need to define a trading partner connection for any individual IP addresses, to increase security.
3. Click Configuration from the menu bar.
4. Expand the Netmaps tree and click the netmap to modify.
5. Click New to add a new inbound node definition.
6. Using the information you defined on the Inbound Connection Definition Worksheet, provide the following information and click Save:
 - ◆ Inbound Node Name
 - ◆ Peer Address Pattern
 - ◆ Policy
7. Repeat step 6 for every group of connections and for every individual IP address connection you want to define.
8. If necessary, reorder the node definitions in the netmap. Order definitions from most specific to least specific since they will be evaluated in order.
 - a. Click the radio button beside the inbound node definition to move.
 - b. Click Move Up or Move Down until the node definition is in the correct order.
9. Click OK.
10. Click Save.

Authenticate an Inbound SFTP Node Against Information Stored in the Local User Store

The Create a Basic SFTP Configuration scenario does not authenticate the inbound node. For additional security, you may configure the authentication method to use for the inbound node. You can choose from the following user authentication methods:

Authenticate the password presented by the inbound node against information stored in the local user store.

Authenticate the key presented by the inbound node against information stored in the authorized user key store.

Authenticate either the password or the key presented by the inbound node using information stored in the local user store or the authorized user key store.

Authenticate both the password and the key presented by the inbound node using information stored in the local user store and the authorized user key store.

Authenticate a password using information stored in the EA.

The following scenarios build on the Create a Basic SFTP Configuration scenario by adding user authentication of the inbound node using information from the local user store. Determine which authentication method you want to enable and then complete the procedure to implement it. Refer to *Strengthen the SFTP User Authentication Using EA* on page 207 for instructions on configuring user authentication using EA.

Add Local Authentication to an Inbound Node Worksheet

Before you add user authentication to the inbound connection you created in the Basic Configuration scenario, gather the information on the Add Local Authentication to an Inbound Node Worksheet. Use this information as you configure user authentication for the inbound connection.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	
Required Authentication Method	Method to use for the inbound node. Options include: <ul style="list-style-type: none"> ◆ Password ◆ Key ◆ Password and Key ◆ Password or Key 	
Internal User ID	The source to use to for the internal user ID.	Pass-Through or Netmap
Name	Name to assign to the user you create.	
Password Confirm Password	If you are authenticating the user-supplied password, identify the password value to use to validate the inbound password.	

Add Local Authentication to the Inbound Node Using Password Information

This scenario builds on the Basic SFTP Configuration by adding user authentication to the inbound connection. It compares a password presented by the inbound node to information defined in the local user store. You must add the password information to the local user store before you can test this scenario. Refer to *Manage CM User Accounts* on page 257 for instructions.

To add support for password authentication:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select the policy to modify.
3. Click the Advanced tab.
4. Select Password as the Required Authentication Method.
5. Enable the User Authentication Mechanism: Through Local User Store option.
6. Click Save.

Authenticate an Inbound Node Using Key Information

This scenario builds on the Basic SFTP Configuration by adding inbound user authentication using a key. This authentication method requires that credentials for the Sterling Integrator server be defined in the netmap since only the password can be passed through to the Sterling Integrator server. You must add the key information to the user definition before you can test this scenario. Refer to *Add SSH Keys to a User Account* on page 261 for instructions.

To add support for key authentication:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy.
3. Click the Advanced tab.
4. Select Key as the Required Authentication Method.
5. Enable the User Authentication Method: Through Local User Store option.
6. In the Internal User ID field, select Netmap.
7. Click Save.
8. Expand the netmap tree and open the netmap to edit.
9. Click the Outbound Nodes tab.
10. Select the outbound node to edit and click Edit.
11. Click the Advanced tab.
12. Type the user ID and password or key to use to connect to the outbound Sterling Integrator server.
13. Click OK.
14. Click Save.

Authenticate an Inbound Node by Comparing Either a Password or a Key to the Local User Store

This scenario builds on the Basic SFTP Configuration by adding support for either key or password authentication of the inbound connection. The inbound node may present a key or a password. Only one must be authenticated for a communications session to be established. This authentication method requires that credentials for the Sterling Integrator server be defined in the netmap, since only a password can be passed through to the Sterling Integrator server.

You must add the password and key information to the user definition in the local user store before you can test this scenario. Refer to *Create an Engine User Account* on page 261 for instructions on creating a user account and assigning a password. Refer to *Add SSH Keys to a User Account* on page 261 for instructions on adding a key to a user account definition.

To add support for either password or key authentication:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy.
3. Click the Advanced tab.
4. Select Password or Key as the Required Authentication Method.
5. Enable the User Authentication Mechanism: Through Local User Store option.
6. In the Internal User ID field, select Netmap.
7. Click Save.
8. Expand the netmap tree and open the netmap to edit.
9. Click the Outbound Nodes tab.
10. Select the outbound node to edit and click Edit.
11. Click the Advanced tab.
12. Type the user ID and password to use to connect to the outbound Sterling Integrator server.
13. Click OK.
14. Click Save.

Authenticate an Inbound Node by Comparing Both a Password and a Key to the Local User Store

This scenario builds on the Basic SFTP Configuration by adding support for both key and password authentication of the inbound connection. The inbound node must present both a key and a password and both must be authenticated for a communications session to be established.

You must add the password and key information to the user definition in the local user store before you can test this scenario. Refer to *Create an Engine User Account* on page 261 for instructions on creating a user account and assigning a password. Refer to *Add SSH Keys to a User Account* on page 261 for instructions on adding a key to a user account definition.

To add support for password and key authentication:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and select a policy.

3. Click the Advanced tab.
4. Select Password and Key as the Required Authentication Method.
5. Enable the User Authentication Method: Through Local User Store option.
6. In the Internal User ID field, select Pass-through.
7. Click Save.

After you configure user authentication using both key and password information, validate the configuration by establishing a session initiated by an SFTP client to an SFTP server.

Provide User Mapping Using the Netmap

This scenario builds on the Basic SFTP Configuration by enabling the use of different user credentials for the outbound connection to the SFTP server.

If you configure this option, the credentials presented by the inbound trading partner are not used to connect to the SFTP server. Credentials stored in the netmap are used to connect to the SFTP server. This method prevents trading partners from accessing the actual credentials used to connect to the internal SFTP server.

After you configure the use of alternate credentials to connect to the SFTP server using information from the netmap, test the configuration by establishing a session initiated by an SFTP client to a SFTP server. Refer to *Test the Inbound and Outbound Connections* on page 210 for more information on testing the configuration described in this scenario.

Provide User Mapping Using the Netmap - Worksheet

In this scenario, edit the netmap and the policy you created in the basic configuration to strengthen the outbound connection by providing user credentials and a mapping method to use to secure the outbound connection to the SFTP server.

Collect the following information so you can match the SSP configuration with the SFTP server configuration. Use the information on this worksheet as you edit the outbound node definition, and be sure to select the netmap and policy you created in the Basic Configuration.

Configuration Manager Field	Feature	Value
Netmap	Name of netmap to modify.	
Policy	Name of policy to modify.	
User ID	User ID to connect to the SFTP server (Defined at the SFTP server).	
Password	Password to connect to the SFTP server (Defined at the SFTP sever).	
Local User Key Stores	The name of the key store where the key to authenticate SSP to the outbound connection is stored.	

Configuration Manager Field	Feature	Value
Local User Key	The local user key to use to authenticate SSP to the outbound connection.	

Connect to the Outbound Server Using Credentials from the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to prevent the user ID and password, or the key provided by the trading partner, from being used to connect to the server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and uses information stored in the netmap to connect to the outbound server.

Before you configure this option:

- Ensure that a user ID and password or key are defined for the outbound connection on the SFTP server

- Obtain the user ID and password.

To configure validation for the outbound connection using credentials stored in the netmap:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and click the netmap to modify.
3. Click the Outbound Nodes tab.
4. Select the outbound node to modify and click Edit.
5. Click the Advanced tab.
6. Type the values to use to connect to the SFTP server:
 - ◆ User ID
 - ◆ Password
 - ◆ Local User Key Stores
 - ◆ Local User Key
7. Click Save.
8. Expand the Policies tree and click the policy to modify.
9. On the Policy Configuration panel, click the Advanced tab.
10. From the User Mapping: Internal User ID list, select Netmap.
11. Click Save.

Strengthen the SFTP User Authentication Using EA

This scenario builds on the basic SFTP configuration by adding user and password authentication or user and key authentication using information defined in EA. To provide a more advanced method of securing the SFTP connection, use EA.

Authenticate an Inbound SFTP User or Key Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines the options that are enabled. EA will return a user ID, password, and routing name for a local user key stored on SSP. Refer to Sterling External Authentication Server documentation library for the functions that can be performed in EA.

Authenticate an Inbound SFTP User or Key Using EA Worksheet

Use the following worksheet to identify the information needed to authenticate a trading partner user ID, password, or key using EA. Update the policy you created in the Basic Configuration for this scenario.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	
Required Authentication Method	Method to use to authenticate the inbound node. Options include: <ul style="list-style-type: none">◆ Password◆ Key◆ Password and Key◆ Password or Key	
User Authentication Mechanism - Through External Authentication	Enable this option because you will validate user information using EA.	
User Authentication Profile	If you are authenticating a user ID and password, type the name of the profile defined in EA used to authenticate the user.	
Key Authentication Profile	If you are authenticating the user ID and key, type the name of the profile defined in EA to authenticate the key.	

Authenticate the Inbound User ID and Password Using EA

To authenticate the user ID and password provided by the inbound node against information stored in an LDAP database, you must configure EA. After you configure EA to enable user authentication, use this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP, obtain the name of the EA definition and ensure that the EA server connection has been configured.

To configure authentication of an inbound node password using EA:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Select Password or Password or Key in the Required Authentication Method field.

5. Enable User Authentication Mechanism - Through External Authentication and type the name of the user authentication definition you defined in EA in the User Authentication Profile field.
6. Deselect the Through Local User Store option.
7. Click Save.

You can now associate this policy with a inbound node for which you want to perform user authentication using EA.

Authenticate the Inbound User ID and Key Using EA

To authenticate key information about the inbound node against information stored in an LDAP database, you must configure EA. After configuring EA, use this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP, obtain the name of the EA definition and ensure that the EA server connection has been configured.

To configure authentication of an inbound node password using EA:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Select Key in the Required Authentication Method field.
5. Enable Key Authentication Mechanism - Through External Authentication and type the name of the key authentication definition you defined in EA in the Key Authentication Profile field.
6. Deselect the Through Local User Store option.
7. Click Save.

You can now associate this policy with a inbound node for which you want to perform key authentication using EA.

Strengthen the Outbound SFTP Connection With EA User Mapping

This scenario builds on the basic SFTP configuration by adding user or key mapping using information defined in EA. To provide a more advanced method of securing an SFTP connection, use EA to map a user ID and password or user key presented by the inbound node to login credentials stored in EA. The mapped login credentials are used to connect to the outbound server in the secure zone.

Manage SFTP User Mapping Using EA

For a higher level of security when connecting to the outbound server, use information stored in an LDAP database to connect to the outbound server. To use information in an LDAP database, you configure EA. You can use EA to map a user ID, password, or key provided by an inbound connection to a user ID, password, or key that is not exposed to the external node.

Perform User Mapping Using EA in an SFTP Environment Worksheet

Use this worksheet to identify the information needed to authenticate a trading partner user ID, password, or key using EA. Update the policy you created in the Basic Configuration for this scenario.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	
Internal User ID	The source to use to for the internal user ID.	External Authentication

Connect to the Outbound SFTP Node Using Information Stored in LDAP

If you store user credentials in an LDAP database, use this procedure to configure SSP to use these credentials to connect to the secure outbound server.

Before you configure this option:

Configure a SSH key authentication definition in EA and obtain the name of the EA definition.

Configure the EA server to allow connections from SSP.

Ensure that the public keys for SSP have been sent to the EA server and imported into the EA trust store.

Configure SSP for user authentication through EA. Refer to *Strengthen the SFTP User Authentication Using EA* on page 207.

To configure the use of a password or a key from the LDAP database:

1. Click Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Select From External Authentication in the User Mapping:Internal User ID field.
5. Click Save.

Test the Inbound and Outbound Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between an SFTP client and the engine, initiate a session from the engine to the SFTP server in the trusted zone, and review the SSP audit log for the results.

Note: Configuration files must be available at the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

Establish a session initiated by a trading partner using an SFTP client

Initiate an outbound session to an SFTP server on behalf of the client connection

To verify the communications sessions:

1. Make sure the engine is running.
2. Initiate a client session to the SFTP server in your trusted zone from a trading partner.
3. View the log file at the client to ensure that the connection from the inbound node to SSP was successful.
4. View the log file of the engine to ensure that the connection to SSP was successful.

Route an Outbound Connection to Alternate SFTP Servers

When you configure an SFTP adapter, you define a primary SFTP server to connect to. For each outbound node definition, you can identify up to three alternate outbound nodes to connect to if the primary SFTP server is not available.

Two methods of configuring alternate SFTP server routing are available.

Select an SFTP server from the drop-down list. To configure this method, you first configure an outbound node definition in the netmap for each alternate SFTP server. Each alternate connection uses the security and advanced settings defined for the outbound node in the netmap.

Select IP address/port from the drop-down list and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap. Each alternate connection uses the security and advanced settings defined in the primary node definition.

If you configure alternate SFTP server definitions in the outbound node definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2. If this connection is unsuccessful, SSP tries to connect to the third alternate, Node 3. If the connection to this node is unsuccessful, the inbound connection is aborted.

To configure alternate outbound connections:

1. Click Configuration from the menu bar.
2. Expand the Netmap tree and click the netmap to modify.
3. Click the Outbound Nodes tab.
4. Select the outbound node to modify and click Edit.
5. Click the Advanced tab.
6. Do one of the following:
 - ◆ To identify an alternate node that is defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.
 - ◆ To configure an alternate node that is not in the netmap and use the security setting defined in the primary node definition:
 - a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.
 - b. Provide the IP Address and Port number for the alternate outbound node
7. Click OK.

8. Click Save to save the netmap updates.

PeSIT Proxy Configuration

SSP and PeSIT Overview

Sterling Secure Proxy (SSP) acts as an application proxy between Connect:Express and PeSIT nodes. It provides a high level of data protection between external PeSIT connections and your internal network. Define an inbound node definition for each trading partner connection from outside the company and outbound node definition for every company server to which SSP will connect.

SSP provides reverse proxy services for Connect:Express servers when the trading partners initiate sessions to Connect:Express servers in the trusted zone. SSP provides forward proxy services for Connect:Express servers when the node in the trusted zone initiates a session to a server at a remote trading partner.

SSP provides these services for Connect:Express and PeSIT nodes in a manner similar to the way it provides these services for other protocols.

The PeSIT configuration scenarios describe how to configure PeSIT protocol connections to and from the SSP engine using Configuration Manager.

Supported PeSIT Software

The following software is supported for use with the SSP PeSIT Proxy Adapter:

- Connect:Express for z/OS (formerly OS/390) version 4.2.2 or later

- Connect:Express for UNIX version 1.4.4 or later

- Connect:Express for Windows version 3.0.5 or later

Organization of the PeSIT Configuration Scenarios

The first scenario instructs you how to do a basic setup. Each successive scenario adds an additional security feature to the basic configuration. After you go through each scenario, test the connection to ensure that it is correctly configured. You determine your security needs and configure the security features applicable to your environment.

The scenarios include the following:

- Create a basic PeSIT configuration

- Add SSL/TLS support

- Configure PNODE-based routing

- Add local Logon ID authentication

- Provide outbound credentials using the netmap

The remaining configuration scenarios require Sterling External Authentication Server (EA), an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to EA to enforce the following advanced security features:

- Authenticate an inbound certificate or user using EA

Configure logon ID mapping to the SNODE using EA

Configure certificate-based routing

Additional procedures are provided to instruct you how to configure the following features:

Define alternate nodes for failover support

Enable action based on protocol errors

Block a PeSIT command from a PNODE

Complete Scenario Worksheets

Before you perform each PeSIT configuration, gather the information on the provided worksheet. You use this information as you configure each feature. Complete worksheets as follows:


Enter a value for each listed SSP feature. Fields listed in the worksheet are required.

Accept default values for fields not listed in the worksheet.

The worksheet identifies the Configuration Manager field where you will specify each value.

Complete and Test Configuration Scenarios

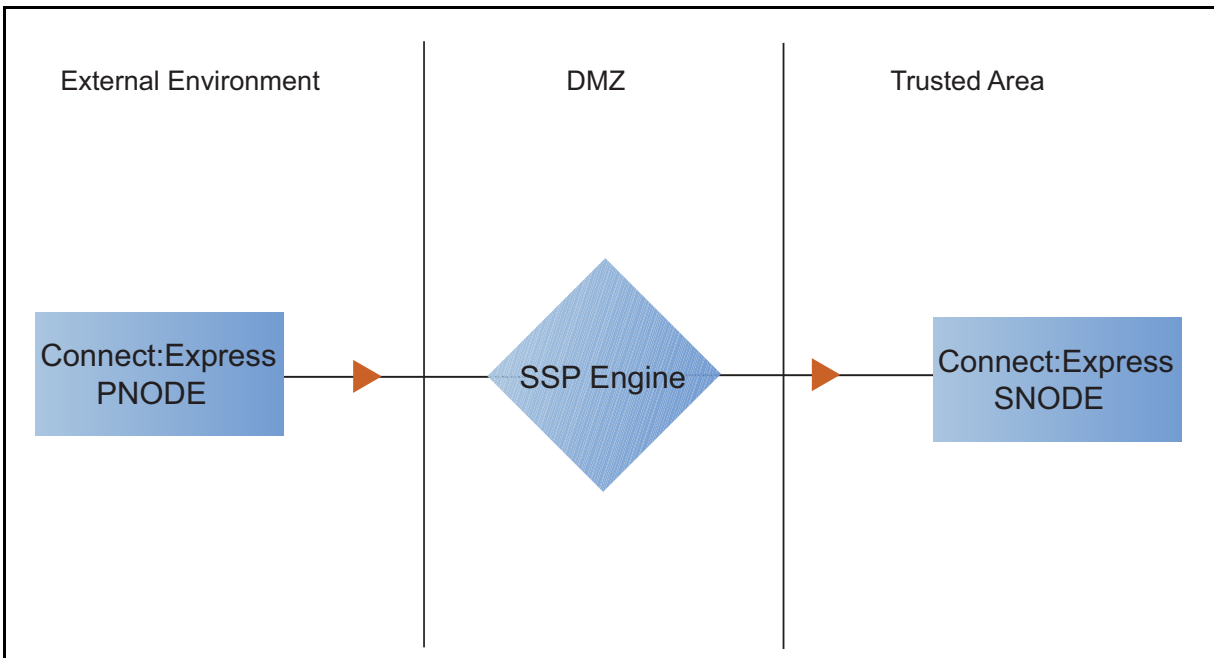
Work through the sequence of PeSIT configuration scenarios in the order in which they are presented to add security features. Be sure to test each feature before you add the next one to the configuration. Before you move SSP into production, ensure that you have configured and tested all security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: . To view more information about the error, hover over the icon.

Create a Basic PeSIT Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection between PeSIT servers. Using default values, the PNODE presents a Logon ID to connect to the SNODE without EA. As a result, no authentication occurs in SSP and the logon ID presented by the PNODE is used to connect to the SNODE. This scenario assumes that all nodes are Connect:Express nodes.

The basic configuration uses standard routing to route connections to the node you define in the adapter. You are instructed on how to configure PNODE routing, mixed routing, and certificate-based routing in later scenarios.



Before you configure a PeSIT connection, make sure that an engine has been configured. Refer to *Install or Upgrade SSP on UNIX or Linux* on page 55 or *Install or Upgrade SSP on Windows* on page 67 for instructions.

After you configure SSP, validate the configuration by initiating a PeSIT connection from the PNODE. For more information on testing the configuration, see *Test the PeSIT Connections* on page 233.

Complete the following tasks to define a basic PeSIT configuration:

- Create a policy
- Define PeSIT nodes in a netmap
- Define a PeSIT adapter

Basic PeSIT Configuration Worksheet

Before you configure SSP for PeSIT connections, gather the information on the basic PeSIT configuration Worksheet. You use this information as you configure a basic PeSIT connection for SSP.

Policy

Create a basic policy. In a later PeSIT configuration scenario, you edit this policy to add security features to it.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy	

Netmap (All PeSIT Nodes)

Create a netmap that contains connection information for the nodes connecting to and from SSP. For each node, associate a policy with the node.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name	
PeSIT Node Definition		
Node Name	Name to assign to the PeSIT node definition	
PeSIT Server Address	Host name or IP address of the PeSIT server	
PeSIT Port	Listening port number of the PeSIT server	
Policy	Name of policy you create (Select from a pull-down list.)	
Node Name	Name to assign to the PeSIT node definition	
PeSIT Server Address	Host name or IP address of the PeSIT server	
PeSIT Port	Listening port number of the PeSIT server	
Policy	Name of policy you create (Select from a pull-down list.)	
Node Name	Name to assign to the PeSIT node definition	
PeSIT Server Address	Host name or IP address of the PeSIT server	
PeSIT Port	Listening port number of the PeSIT server	
Policy	Name of policy you create (Select from a pull-down list.)	

PeSIT Adapter

Create a PeSIT adapter that defines information necessary to establish PeSIT connections to and from SSP. When configuring the adapter, select the basic netmap and the PeSIT server where connections are routed and defined in the netmap definition.

Configuration Manager Field	Feature	Value
Name	Adapter name	
Listen Port	Listen port to use for inbound connections	
Netmap	Netmap to associate with the adapter	
SNODE Netmap Entry	Name of PeSIT node where the connection is routed	
Engine	Engine to run the PeSIT adapter on	

Create a Basic PeSIT Policy

The policy defines how you impose controls to authenticate a PeSIT PNODE trying to communicate with a PeSIT SNODE over the public Internet. The basic policy does not enforce any controls over the defined node. You add security controls when you define more advanced security settings.

To define a basic policy:

1. Select Configuration from the menu bar.
2. Click Actions > New Policy > PeSIT Policy.
3. Type a Policy Name.
4. Click Save.

Create a PeSIT Netmap

You define connection information for every PeSIT node that communicates using SSP. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define PeSIT nodes:

1. Select Configuration from the menu bar.
2. Click Actions > New Netmap > PeSIT Netmap.
3. Type a Netmap Name.
4. To define a PeSIT node definition, click New.
5. Specify the following values:
 - ◆ Node Name
 - ◆ PeSIT Server Address or hostname
 - ◆ PeSIT Server Port (listening port)

- ◆ Policy
6. Click OK.
 7. Repeat steps 4 through 6 for each node you want to define. Define at least one PNODE and at least one SNODE in order to establish a connection between two PeSIT nodes.
 8. Click Save.

Define the PeSIT Adapter Used for the Connection

A PeSIT adapter definition specifies system-level communications information necessary for PeSIT connections through SSP.

Before you begin this procedure, create a netmap and an engine to associate with the adapter.

To define a PeSIT adapter:

1. Select Configuration from the menu bar.
2. Click Actions > New Adapter > PeSIT Proxy.
3. Specify values for the following:
 - ◆ Name
 - ◆ Listen Port
 - ◆ Netmap
 - ◆ SNODE Netmap Entry
 - ◆ Engine
4. Click Save.

What You Defined with the Basic PeSIT Configuration Scenario

Creating connections between PeSIT nodes when routing them through SSP requires that you organize information about the PeSIT nodes in a policy, a netmap, and an adapter definition. You created these items when you defined the basic PeSIT configuration. The next step is to test the configuration to ensure that the connections work. Before you test the configuration, be sure that:

The Connect:Express SNODE server has a definition in its partner's directory for the Connect:Express or PeSIT PNODE. The definition must use the IP address of the SSP server. The local name must be the SNODE name.

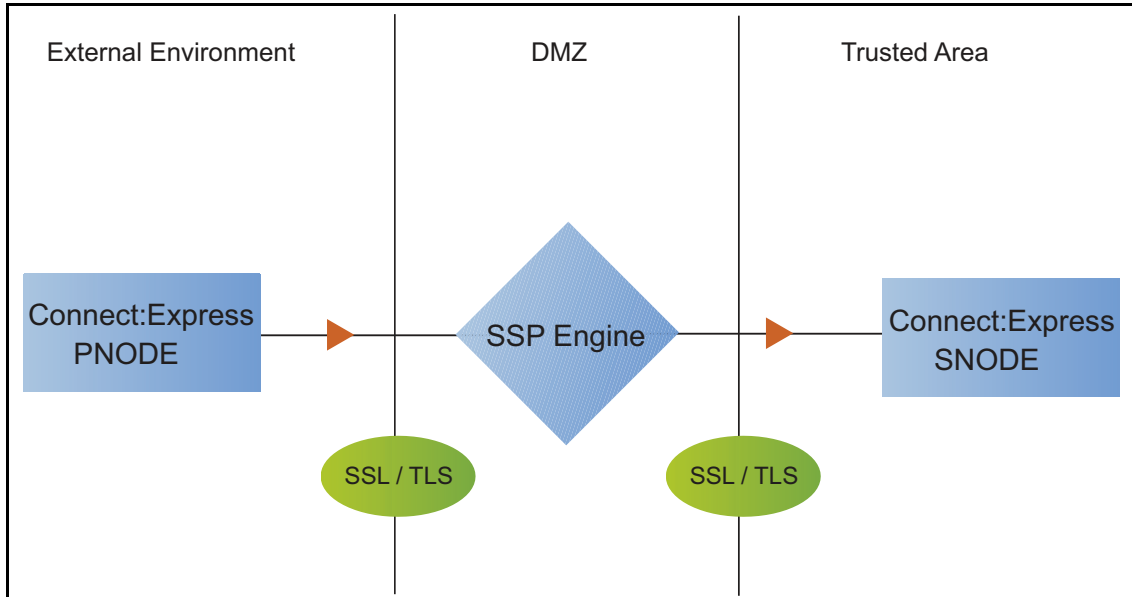
The Connect:Express PNODE server has a definition in its partner's directory for the Connect:Express or PeSIT SNODE, using the IP address and port of the SSP server. The local name must be the PNODE name.

Refer to *Test the PeSIT Connections* on page 233 for information about testing the PeSIT proxy configuration outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Add SSL/TLS Support

This scenario builds on the basic PeSIT configuration by enabling security for the nodes you defined in the netmap.



Adding SSL/TLS support to the netmap for the nodes involves selecting the following options for the connections:

- SSL or TLS Protocol
- Cipher suites
- Certificate stores and certificates

Add SSL/TLS support to the PNODE and the SNODE definitions. Set up SSL/TLS parameter files at both the SNODE and the PNODE servers. Obtain certificates for both sessions and check them into the certificate store. Then, test the connection.

Note: This procedure assumes you have checked in your certificates. Refer to *Manage Certificates for SSL/TLS Transactions with Trading Partners* on page 471, for more information.

SSL/TLS Support Worksheet

Before you add SSL/TLS support to the connection information you created in the basic PeSIT configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Select the security setting and cipher suites to be used to secure the connection. To require that the certificate common name be validated in a certificate presented, enable this option and identify the common name value to check. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Node Name	Name of the node to add security to, from the nodes you've already defined.	
Use SSL	Enable this option to enable security checking	Enabled
Verify Common Name	Enable this option to enable common name checking. This is optional.	Enabled/Disabled
Certificate Common Name	Value of common name in certificate presented, if Common Name Checking is enabled.	
Security Setting	Security protocol to use. Options include SSL or TLS.	
Enable Client Authentication	Do you want to require the inbound connection to present its certificate for SSL or TLS client authentication?	
Trust Store	Name of the store for the CA certificate or trusted root certificate	
CA Certificates/Trusted Root	Name of CA certificate/trusted root	
Key Store	Name of the store for the key or system certificate is stored	
Key/System Certificate	Name of the SSP system certificate presented to the PeSIT server	
Available Cipher Suites	Select the ciphers to enable by moving them from the Available Cipher Suites to the Selected Cipher Suites field	

Secure the PeSIT Connection Using the SSL or TLS Protocol

The first step to strengthen security is to secure the communications channel. This procedure describes how to enable the SSL or TLS protocol for the PeSIT connections to and from SSP in a netmap you created in the basic configuration. To require that SSP perform common name checking, enable this option and identify the common name in the configuration.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP Cert Store. Refer to *Manage Certificates for SSL/TLS Transactions with Trading Partners* on page 471 for instructions.

To enable the SSL or TLS protocol:

1. Select Configuration from the menu bar.

2. Expand the Netmaps tree and select a netmap to modify.
3. Select a node to modify, and click Edit.
4. Click the Security tab, and then click Use SSL to enable security.
5. To enable common name checking:
 - a. Click Verify Common Name.
 - b. Type the certificate common name in the Certificate Common Name field.
6. Select values for the following:
 - ◆ Security Setting
 - ◆ Key Store
 - ◆ Key/System Certificate
 - ◆ Available Ciphers
 - ◆ Selected Ciphers
7. To enable client authentication:
 - a. Click Enable Client Authentication.
 - b. Select the Trust Store where the certificate you want to use is located.
 - c. Select the CA Certificates/Trusted Root to use to authenticate the certificate presented by the inbound node.

Note: Be sure to highlight the certificate to select. If only one certificate is displayed in the field, it is not selected until you highlight it.

8. Click OK.
9. Click Save.

Establish a session initiated by a Connect:Express PNODE to test the configuration.

Variation on the Add SSL/TLS Support Configuration

After you confirm that the communications session you established using the Add SSL/TLS Support scenario was successful, you may want to further modify your configuration. After testing the SSL/TLS configuration, you can configure the environment to allow the inbound and outbound sessions to use different levels of encryption.

Allow Different Levels of Encryption for the Inbound and Outbound Node

In a PeSIT environment where SSP is not being used, one session is established between an SNODE and a PNODE. In the SSP environment, a session break is created; therefore, two sessions are established: one between the PNODE and SSP and another between SSP and the SNODE. To use the same protocol on both sessions, use the default settings.

Complete this procedure to define one protocol for the inbound node and a different protocol for the outbound node. This function is useful when you want to secure the inbound connection but allow a nonsecure session between SSP and the outbound node.

To enable different levels of encryption for the inbound and the outbound connection:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree, and select the adapter you want to modify.
3. Click the Advanced tab.
4. Enable the Inbound and outbound sessions can have different levels of encryption option.
5. Click Save.

Configure PNODE-Based Routing

The basic configuration uses standard routing to determine where a connection is routed. If you configure standard routing, all sessions through an adapter are routed to the same connection. To allow a PNODE to determine what SNODE it connects to, configure PNODE-based routing. For PNODE-based routing, you must configure a node definition in the netmap for the PNODE and for all the SNODEs you will route to.

PNODE-based Routing Worksheet

This scenario builds on the basic PeSIT configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic PeSIT configuration scenario, gather the information on the PNODE-based Routing Worksheet. You use this information as you configure PNODE-based routing.

Make sure that you have a node definition for the PNODE and for every node where the connection is routed in the netmap you select.

Configuration Manager Field	Feature	Value
Name	Adapter name	
Routing Type	Routing type to use for this connection	PNODE-specified

Configure PNODE-based Routing

To configure a PeSIT adapter to use PNODE-based routing:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree and select the adapter you want to modify.
3. Select PNODE-specified in the Routing Type field.
4. Click Save.

Configure Mixed Routing

Mixed routing allows a PNODE to determine what SNODE it connects to. If the PNODE does not identify what SNODE to connect to, mixed routing then routes to the SNODE identified in the SSP configuration. Before PNODE-based routing can be implemented, you must configure a node definition in the netmap for the PNODE and the SNODE.

Mixed Routing Worksheet

This scenario builds on the basic PeSIT configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic PeSIT configuration scenario, gather the information on the Mixed Routing Worksheet. You use this information as you configure PNODE-based routing.

Make sure that you have a node definition for the PNODE and for the node where the connection is routed in the netmap you select.

Configuration Manager Field	Feature	Value
Name	Adapter name	
Routing Type	Routing type to use for this connection	PNODE-specified and then Standard
SNODE Netmap Entry	SNODE to route connections to	

Configure PNODE Specified and Then Standard (Mixed) Routing

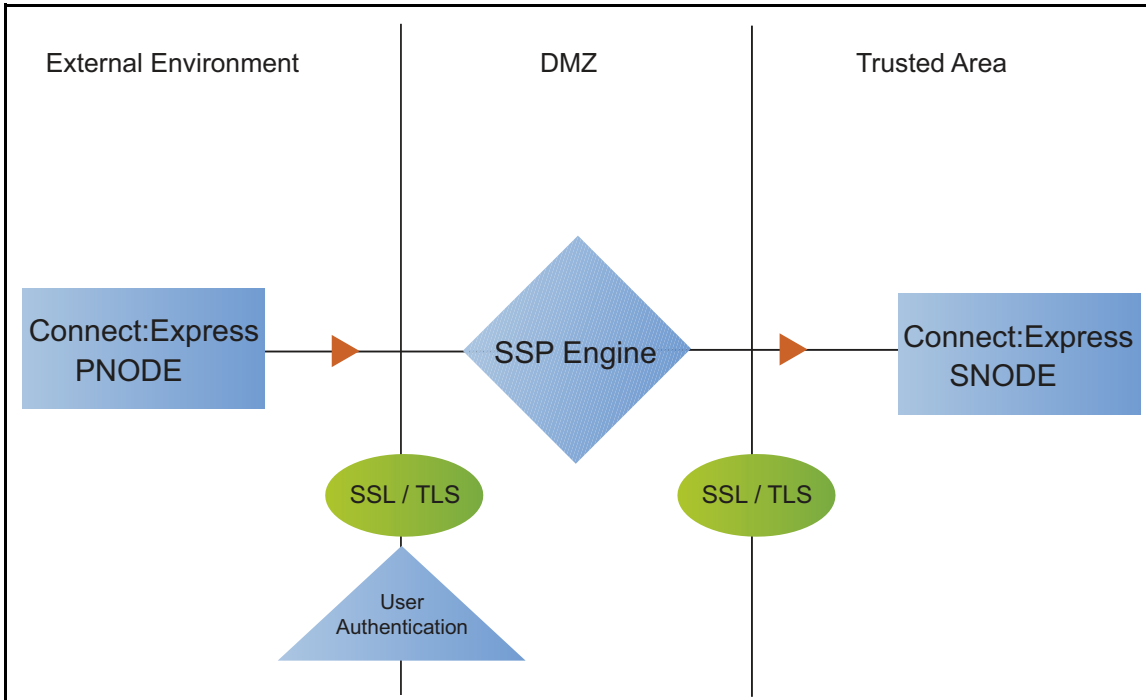
To configure a PeSIT adapter to use PNODE specified and then standard (mixed) routing:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree and select the adapter you want to modify.
3. Select PNODE-Specified, then Standard (mixed) in the Routing Type field.
4. Select the SNODE to route connections to in the SNODE Netmap Entry field.
5. Click Save.

Add Local User Authentication to a PeSIT Connection

This scenario builds on the basic PeSIT configuration by adding local user authentication to the PNODE connection using information defined in the local user store. The logon ID and password presented by the PNODE are authenticated against information stored in the local user store. The

values must match before a connection is established. You must add this information to the local user store before you can test this scenario.



Adding logon ID authentication to the PNODE connection defined in the basic PeSIT configuration involves enabling logon ID authentication and specifying information about the PNODE.

After you configure local logon ID authentication, validate the configuration by establishing a session initiated by a Connect:Express PNODE.

PeSIT PNODE Connection (Local LogonID Authentication) Worksheet

Before you add local logonID authentication to the PNODE connection you created in the basic PeSIT configuration scenario, gather the information on the PeSIT PNODE Connection (Local LogonID Authentication) Worksheet. Use this information as you configure logonID authentication for the PNODE connection.

In this scenario, you edit the policy you created in the PeSIT basic configuration scenario and enable logonID authentication. You also add a logonID and password for the PeSIT PNODE to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node	
LogonID Authentication	Method to use to authenticate the inbound node	Through local user store
User Store	Name of the user store you create	

Configuration Manager Field	Feature	Value
User Name	Name of the user you define in the User Store	
Password Confirm Password	The password value to use to validate the inbound connection	

Add User Authentication to the PeSIT Inbound Connection

You can strengthen the security of PeSIT PNODE connections by enabling local logonID authentication. This procedure describes how to configure local logonID authentication.

To add local logonID authentication for a PNODE connection:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and select a policy.
3. Click the Advanced tab.
4. Enable the LogonID Authentication Through Local User Store option.
5. Click Save.

Add Credentials to the Local User Store

If you enable logon ID authentication through the local user store, you also add logonID information to the local user store that is validated by SSP during a PeSIT client connection.

Before you begin this procedure:

Enable logon ID authentication for the inbound connection.

Ensure that the engine is configured to use the user store that contains the user credentials.

To add user information to the local user store:

1. Select Credentials from the menu bar.
2. Click User Stores to expand the list of user stores.
3. Select the default user store called defUserStore.
4. From the User Store Configuration panel, click New.
5. Specify values for the following:
 - ◆ User Name
 - ◆ Password
 - ◆ Confirm Password
6. Click OK.
7. Click Save.

Provide Credentials to the Outbound PeSIT Node Using the Netmap

This scenario builds on the Basic PeSIT Configuration by enabling the use of Logon credentials from the netmap to connect to the outbound PeSIT connection. Following is an illustration of the security features supported in this scenario:

When an inbound trading partner connects to SSP, its credentials are replaced with credentials stored in the netmap. The replacement credentials are then used to connect to the outbound PeSIT server. This method uses SSP security features to prevent trading partners from knowing the credentials used to connect to the outbound Connect:Express. The outbound Connect:Express must have a partner definition that accepts the LogonID and password provided.

After you configure the environment to use credentials defined in the netmap, test the configuration by establishing a session initiated by a Connect:Express client to a Connect:Express server. Refer to *Test the PeSIT Connections* on page 233 for more information on testing the configuration defined in this scenario.

Provide Credentials for the Outbound PeSIT Node Using the Netmap - Worksheet

In this scenario, edit the netmap and policy you created in the Basic PeSIT Configuration to provide user credentials stored in SSP to connect to the outbound PeSIT connection.

Collect the following information so that you can match the SSP configuration with the Connect:Express server configuration. Use the information on this worksheet as you edit the outbound node definition. Select the netmap and policy you created in the Basic PeSIT Configuration.

Configuration Manager Field	Feature	Value
Logon ID	Partner ID to use to connect to the Connect:Express server. (Must also be defined at the Connect:Express server.)	
Password	Password to use to connect to the Connect:Express server. (Must also be defined at the Connect:Express server.)	

Connect to the Outbound PeSIT Server Using Credentials from the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to store the Logon ID and password to connect to the outbound Connect:Express server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and SSP uses information stored in the netmap to connect to the outbound Connect:Express server.

Before you configure this option:

- Ensure that the Logon ID and password are defined on the Connect:Express server.
- Obtain the Logon ID and password.

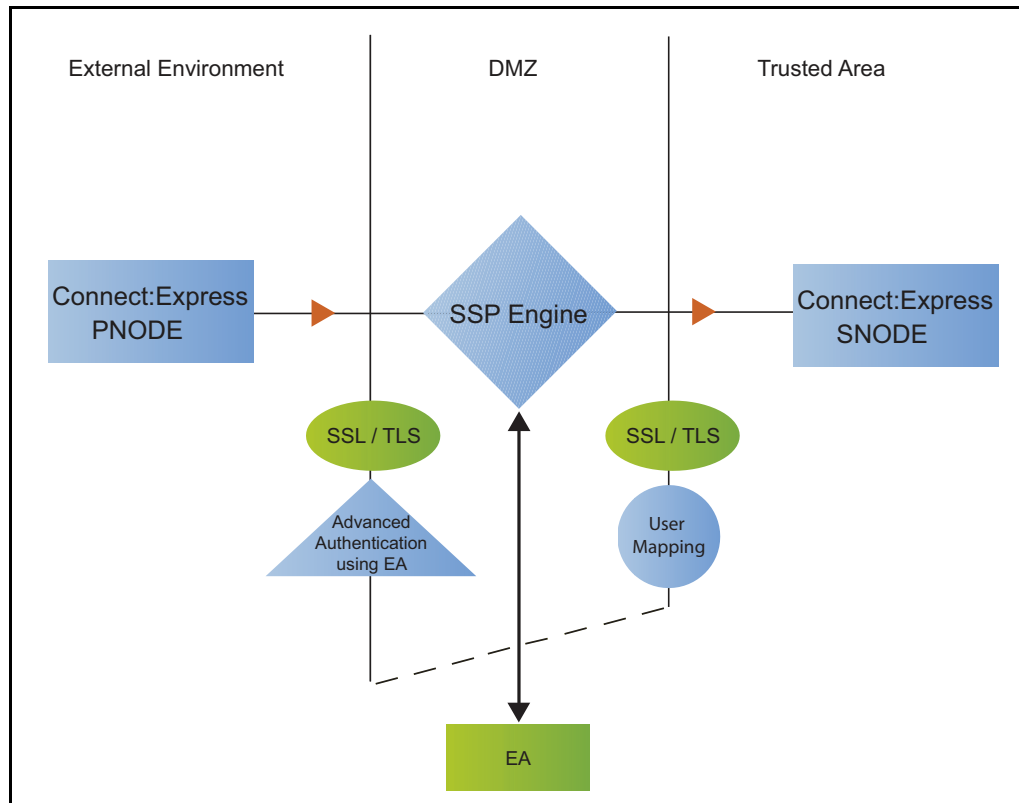
To configure validation for the outbound connection using credentials stored in the netmap:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select the PeSIT netmap to modify.
3. Select the Connect:Express server node to modify and click Edit.
4. Click the Advanced tab.
5. Type values in the following fields for connecting to the Sterling Integrator server:
 - ◆ Logon ID
 - ◆ Password
6. Click Save.
7. Expand the Policies tree and select the policy to modify.
8. On the PeSIT Policy Configuration panel, click the Advanced tab.
9. From the LogonID Mapping: Internal LogonID list, select Replace LogonID with Netmap LogonID.
10. Click Save.

Test the configuration to ensure that this feature is working.

Strengthen LogonID Authentication Using EA

This scenario builds on the basic PeSIT configuration by adding logonID authentication to the PNODE connection using information defined in EA. To provide a more advanced method of securing a PeSIT connection, use EA to authenticate certificate information or logonID credentials presented by the inbound node or to perform logonID and password mapping.



Authenticate an Inbound Certificate or LogonID Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines the options that are enabled. Refer to the Sterling External Authentication Server documentation library for the functions that can be performed in EA.

Authenticate a Certificate or LogonID Using EA - Worksheet

Use the following worksheet to identify the information needed to authenticate a PeSIT connection using information in EA. Update the policy you created in the basic PeSIT configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the certificate presented by the PNODE?	(Yes or No)

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Profile	If yes, provide the EA certificate validation definition.	
User Authentication - Through External Authentication	Will you validate user information?	(Yes or No)
User Authentication - External Authentication Profile	If yes, provide the EA user validation definition.	

Authenticate a PeSIT Certificate or LogonID Using EA

To authenticate certificate information or logonID information about the PeSIT node against information stored in an LDAP database, you must configure EA. After you configure EA to enable certificate or logonID authentication, complete this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP to use EA to authenticate a node connection, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

- The public keys for SSP have been sent to the EA server and imported into the EA keystore.

- The EA server connection has been configured in SSP.

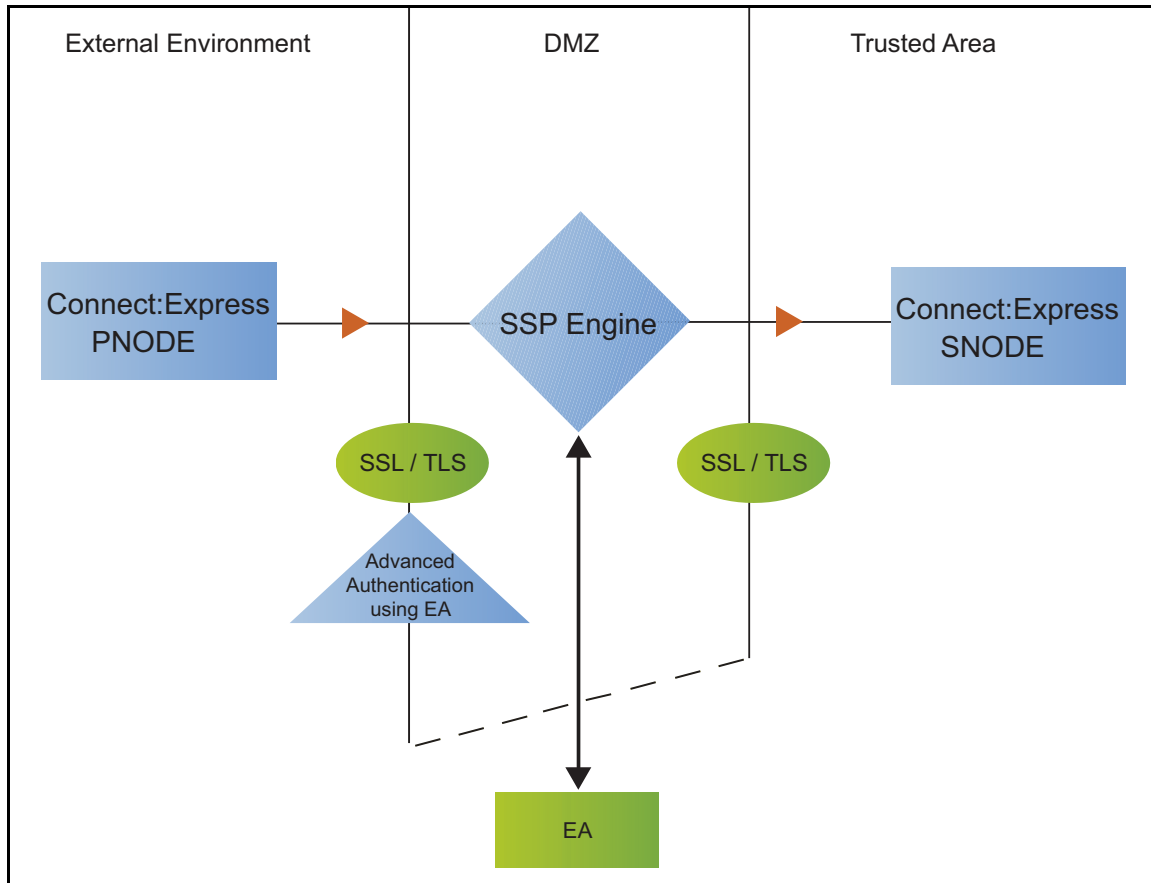
To configure authentication of a PeSIT node using EA:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Configure one or more of the following options:
 - ♦ To validate the certificate presented by the node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and enter the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.
 - ♦ To enable logonID authentication through EA, enable LogonID Authentication - Through External Authentication and type the name of the definition you defined in EA in the LogonID Authentication - External Authentication Profile field.
5. Deselect the Through Local User Store option.
6. Click Save.

You can now associate this policy with a PeSIT node where you want to perform logonID authentication using information stored in an LDAP database.

Strengthen the Connection to the SNODE With LogonID Mapping

This scenario builds on the basic PeSIT configuration by adding logonID mapping using information defined in Sterling External Authentication Server (EA). To provide a more advanced method of securing a PeSIT connection, use EA to map a PNODE logonID and password to login credentials stored in EA. The mapped login credentials are then used to connect to the SNODE.



Perform LogonID Mapping Using EA - Worksheet

Use this worksheet to identify the logonID mapping method to enable for the SNODE connection with information in EA:

Configuration Manager Field	Feature	Value
Replace LogonID with Userid mapped in External Authentication	The PNODE requires a LogonID to access the SNODE. The LogonID provided is replaced with a value defined in EA.	Enabled

Configuration Manager Field	Feature	Value
Destination Service Name	Name of the service. If no value is provided, the SNODE is used as the service name.	

Perform LogonID Mapping Using Information Stored in EA

If you store user credentials in an LDAP database, use this procedure to map a logonID and password to information stored in EA.

Destination Service Name needs to be selected on the Advanced tab of the Netmap Node screen of the PNODE. If Destination Service Name is not provided, the SNODE name is used.

Before you configure this option:

- Configure a definition in EA.
- Obtain the name of the EA definition.
- Configure a connection between EA and the engine.

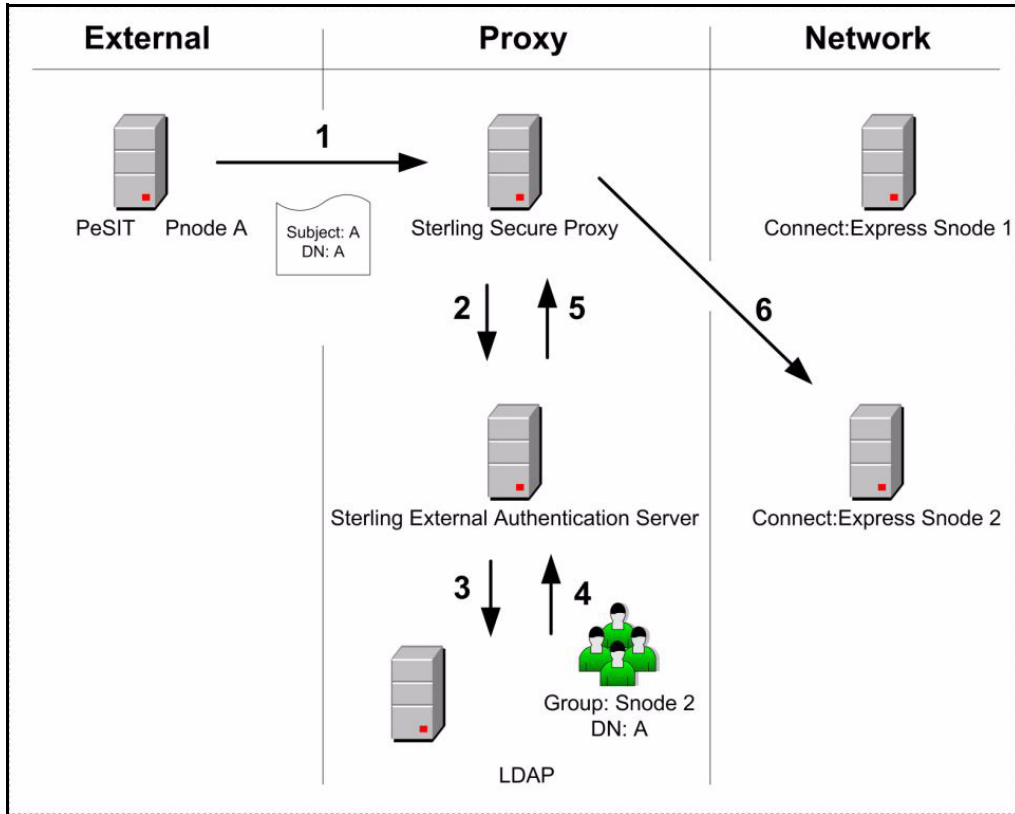
To configure logonID mapping:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. To enable logonID authentication through EA, enable the LogonID Authentication Through External Authentication option and type the name of the definition you defined in EA in the External Authentication Profile field.
5. Select Replace LogonID with LogonID mapped in External Authentication.
6. Click Save.
7. In the Configuration panel, expand the Netmap option and click the netmap to modify.
8. Select the PNODE to modify and click Edit.
9. Click the Advanced tab.
10. Type the name of the service in the Destination Service Name field. If no value is provided, the SNODE name is used as the service name.
11. Click OK.
12. Click Save.

Configure Certificate-Based Routing

This scenario builds on the basic PeSIT configuration by configuring certificate-based routing. Certificate-based routing uses a routing name returned by EA. It is associated with the subject distinguished name found in the PNODE certificate. SSP uses this routing name to determine the SNODE where the incoming SSP connection is routed. To perform certificate-based routing, modify an adapter you defined in the basic PeSIT configuration.

The following diagram illustrates the certificate-based routing function:



Summary of Certificate-Based Routing

Following are the steps performed during certificate-based routing:

1. The PNODE passes a certificate chain during an SSL/TLS session. This certificate includes several attributes, such as subject and distinguished name (DN).
2. SSP passes the certificate chain to Sterling External Authentication Server (EA).
3. Using the configuration parameters in a certificate validation request, EA attempts to match PNODE certificate attributes to the LDAP server and requests the associated routing value.
4. LDAP returns the routing value to EA.
5. EA passes the routing value to the SSP engine.
6. SSP routes the PNODE request to the SNODE using the routing value.

Configure Certificate-Based Routing in SSP

Before you test certificate-based routing, you must create a certificate validation request in EA that includes an attribute query definition called Routing Names. This attribute query definition is created to retrieve a routing name value using certificate attributes as search criteria. You must also configure a connection between SSP and EA.

Refer to *Configure SSP for Sterling External Authentication Server (EA)* on page 273 for instructions.

To configure certificate-based routing:

1. Select Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter you want to modify.
3. Select Certificate-based in the Routing Type field.
4. Click Save.
5. Click the Netmap navigation panel, expand the Netmap tree, and select the PeSIT adapter that contains the SNODE where the connection are routed.
6. Select the node to modify and click Edit.
7. Type the routing value to be returned from the LDAP server in the Routing Name field. The routing name must exactly match the routing value returned from the LDAP server. This routing name identifies the SNODE for routing the PNODE request.
8. Click OK.
9. Click Save.
10. Configure SSP to enable certificate authentication using EA. Refer to *Authenticate an Inbound Certificate or LogonID Using EA* on page 228.

Test the PeSIT Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between a Connect:Express PNODE and the engine, initiate a session from the engine to the Connect:Express SNODE in the trusted zone, and review the SSP log for the results.

This procedure enables you to verify that the engine can:

- Establish a Connect:Express session between a PNODE and SSP

- Initiate a session to a Connect:Express SNODE on behalf of the Connect:Express PNODE connection

To verify the communications sessions:

1. View the `secureproxy.log`.

2. Confirm that the sessions were established, as shown in the following example.

```
23 avr. 2010 13:39:37,459 INFO [ProxyNearScheduler-Thread-9]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP103I
Session started from Peer Address: PNODE1.company/10.20.10.80
23 avr. 2010 13:39:37,459 INFO [ProxyNearScheduler-Thread-9]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=120000000007 SSP104I
Session Proceeding after Node match: INSERVER
23 avr. 2010 13:39:37,459 INFO [ProxyNearScheduler-Thread-9]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=120000000007 SSE1831I
Authentication mechanism: no authentication.
23 avr. 2010 13:39:37,475 INFO [ProxyNearScheduler-Thread-3]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=120000000007 SSE0103I
Connecting to server.
23 avr. 2010 13:39:37,475 INFO [ProxyNearScheduler-Thread-3]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=120000000007 SSP0237I
Attempting outbound connection with
10.20.129.3/InetSocketAddress-host:/10.20.129.3-port:4004 ...
23 avr. 2010 13:39:37,631 INFO [ProxyFarScheduler-Thread-12]
> : Bytes Received: 646 [at: 7.398711524695777E-4 MBPS]
> Bytes Sent: 402 [at: 4.604151753758053E-4 MBPS]
>
> : Bytes Received: 402 [at: 4.710017574692443E-4 MBPS]
> Bytes Sent: 646 [at: 7.568834212067955E-4 MBPS]
>
23 avr. 2010 13:39:44,459 INFO [ProxyFarScheduler-Thread-14]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=12400000000744407 SSE0112I
Session ended.
```

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Additional PeSIT Configuration Options

Additional PeSIT configuration options support the following features:

- Define alternate nodes for failover support
- Record an error message or shutdown a connection based on protocol errors
- Block PeSIT command from a PNODE

Define Alternate Nodes for Failover Support

If you are using standard routing to connect to a Connect:Express server in the secure zone, you identify a primary server to connect to in the adapter. The primary nodes are defined in the netmap. For each PNODE definition in the netmap, you can identify up to three alternate outbound nodes to connect to if the primary Connect:Express server is not available.

Two methods of configuring alternate server routing are available.

Select a previously defined outbound node from the drop-down list on the Netmap - Advanced tab. To configure this method, you first configure an outbound node definition in the netmap for each alternate node you want to use. Each connection uses the security and External Authentication settings defined for that outbound node in the netmap.

Select IP address/port from the drop-down Node list on the Advanced tab and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap, and each alternate connection shares the security and EA settings defined in the primary node definition.

If you configure alternate server definitions in the PNODE definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2 and then to the third alternate, Node 3. If all are unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. Select Configuration from the menu bar.
2. Expand the Netmaps tree and click the netmap to modify.
3. Select the node to modify and click Edit.
4. Click the Advanced tab.
5. Do one of the following:
 - ◆ To identify an alternate node that is defined in the netmap and use its security settings, select the outbound node name from the drop-down list.
 - ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:
 - a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.
 - b. Provide the IP Address and Port number for the alternate outbound node.
6. Click OK.
7. Click Save.

Record an Error Message or Shut Down a Connection Based on Protocol Errors

To write a warning message to the log file or shut down a connection when a protocol violation occurs during a file transfer, enable this function in the Policy definition.

To enable an action based on a protocol error:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and select the policy to modify.
3. Select the action to take on a protocol error in the Protocol Error Action field.
4. Click Save.

Block PeSIT Command from a PNODE

This scenario builds on the basic PeSIT configuration by adding the capability to prevent a PeSIT command from being executed.

To prevent a PeSIT command from being executed:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Transfer Directions tab.
4. Click on one of the following commands to disable the command:
 - ◆ Receive a File Allowed (SELECT)
 - ◆ Send a File Allowed (CREATE)

5. Click Save.

Change the Logging Levels

The following configuration events are written to the engine audit log:

All fields from an initial engine configuration received from the Configuration Manager

Changed fields from an engine configuration update from the Configuration Manager

Inbound connections received for all protocols

Inbound handshakes completed for the FTP, HTTP, PeSIT, and Connect:Direct protocols

Inbound login successes and failures for the FTP, HTTP, and SFTP protocols

Outbound connections established for all protocols

Outbound handshakes completed for the FTP, HTTP, PeSIT, and Connect:Direct protocols

Outbound login successes and failures for the FTP, HTTP, and SFTP protocols

When you configure a PeSIT node, the logging level for the node is set to None and no log is created. You can change the logging level to one of the following options: ERROR to write error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

Use Perimeter Servers to Manage PeSIT Communications

You can use a perimeter server with SSP to manage inbound and outbound PeSIT and Connect:Express communications. Configure perimeter servers for PeSIT and Connect:Express nodes the same way you configure perimeter servers for Connect:Direct nodes. Refer to *Configure Perimeter Servers to Manage SSP Communications* on page 263.

Manage Your SSP Configuration

After you set up your SSP configuration, refer to *Manage Your SSP Configuration* on page 483. The following sections describe PeSIT-specific considerations:

Modify Properties in an Adapter Definition

Adapters are configured with default settings. Use this procedure to modify a property. For FTP and HTTP adapters, the properties and default values are displayed. To change a property, type a new value for the property key. For SFTP, Connect:Direct, and PeSIT adapters, the properties are not displayed.

Refer to the field level help for a description of the properties. To change a property, type the property name and its key value.

To modify an adapter property:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and click the adapter to modify.
3. Click the Properties tab.

4. Click New to add a new property definition.
5. For each property, specify values for the following:
 - ◆ Key
 - ◆ Value
6. Click Save.

Copy a PeSIT Node

To quickly create a PeSIT node definition, you can copy an existing definition and make the changes necessary to create a new item.

To copy a PeSIT node:

1. Click Configuration from the menu bar.
2. Expand the Netmap tree and click the PeSIT netmap where the node is defined.
3. Click the radio button beside the node to copy and click Copy.

A new node is created and renamed to *CopyofItemName* where *ItemName* is the name of the original node you created.

4. Modify the node definition as necessary.
5. Click OK.
6. Click Save.

Manage Configuration Manager

Use the procedures in this section to modify CM settings.

Change the Password for a CM User

You configured users who can access the CM and defined a password for each user. If these user want change their CM password, provide them with the following procedure.

1. Open Internet Explorer.
2. Type the logon address as follows: `https://hostname:port/SSPDashboard` or `https://ipaddress:port/SSPDashboard`
3. On the logon screen, type the user ID and password.
4. Click **New Password**.
5. Type the new password in the New password and Confirm password fields.
6. Click **Confirm**.

Change the CM Passphrase on UNIX or Linux

To change the passphrase defined for CM at installation.

1. Navigate to `install_dir/bin`, where `install_dir` is the CM installation directory.
2. To stop CM, type the following command and press **Enter**:

```
./stopCM.sh
```

3. Type the passphrase defined for CM and press **Enter**.
4. At the administrator ID prompt, type the administrator ID and press **Enter**.
5. At the password prompt, type the password and press **Enter**. CM stops.
6. Type the following command, and press **Enter**.

```
./changePassphrase.sh
```

7. Type the current passphrase and press **Enter**.
8. Type a new passphrase and press **Enter**. Retype the new passphrase and press **Enter**.
Your new passphrase is effective the next time you start CM.

Change the CM Passphrase in Windows

To change the passphrase defined for CM at installation:

1. Click **Start>Run>Browse**.
2. Browse to the `install_dir\bin` directory, where `install_dir` is the CM installation directory.
3. To stop CM, double-click the following file:

```
stopCM.bat
```

4. Click **OK**.
5. Type the CM passphrase and press **Enter**.
6. At the administrator ID prompt, type the administrator ID and press **Enter**.
7. At the password prompt, type the password and press **Enter**.
CM stops.
8. Double-click the following file:

```
changePassphrase.bat
```

9. Click **OK**.
10. Type the current passphrase and press **Enter**.
11. Type the new passphrase and press **Enter**. Retype the new passphrase and press **Enter**.
Your new passphrase is effective the next time you start CM.

Change the Listen Port on CM

To change the listen port on UNIX, Linux, or Windows:

1. Stop CM.
2. Navigate to *install_dir*\bin, where *install_dir* is the CM installation directory.
3. Type the following command:

```
configureaccepter port =
```

where *nnnn* is the port to listen on.

If the command is successful, a response similar to the following is displayed:

```
Acceptor configuration updated:
name      = Secure
secure   = true
port      = nnnn
address   = (default)
timeout   = 30000
enabled   = true
```

All changes to the listen accepters take effect the next time CM is started.

Modify the Timeout Value for a CM Session

By default, a CM session times out after 30 minutes.

To change the CM session timeout value on UNIX, Linux, or Windows:

1. Open the web.xml file in the `install_dir\app\jetty\webservices\webapps\SSPDashboard\WEB-INF` directory.
2. Change the following parameter to identify how many minutes before a session time out.

```
<session-timeout>30</session-timeout>
```

3. Save the file.

Modify the Listener Settings for CM

When you install SSP, you define the IP address and port that CM uses to listen for secure connections from the engine.

To change the IP address and port used for secure connections:

1. Select **System** from the menu bar.
2. Click **Actions>System Settings**.
3. Change the values in the **IPAddress** and **Secure Listener Port** fields.
4. Click **Save**.

Modify Security Settings for CM

Use this procedure to modify the security information used during a secure connection from CM to the web server. You must export the certificate information and add it to the engine setup.

Note: This procedure does not include all steps necessary to configure security settings for CM. Refer to [Manage Certificates Between SSP Components](#) to configure security settings.

To modify security settings for CM:

1. Select **System** from the menu bar.
2. Click **Actions>System Settings**.
3. Click the Security tab.
4. Change the values in the **Key/System Certificate** and **Cipher Suites** fields.
5. Click **Save**.

Modify Logging for Sessions Between CM and the Web Server

To modify the logging level for sessions between CM and the web server:

1. Select **System** from the menu bar.
2. Expand the System Settings tree and click `CMSysSystemSettings`.
3. Click the Globals tab.
4. Modify the logging level.
5. Click **Save**.

Modify Connection Settings for Sessions Between CM and the Web Server

To modify the connection settings for sessions between CM and the web server:

1. Select **System** from the menu bar.
2. Click **Actions>System Settings**.
3. Click the Globals tab.
4. Modify one or more of the connection values.
5. Click **Save**.

Unlock a CM Component

Use the Lock Manager to unlock CM components. A component may become locked if it is already being edited by another user or if the browser is closed without logging out of CM.

To unlock a CM component:

1. Select **System** from the menu bar.
2. Click **Lock Manager**.
3. In the show field, select the component to unlock.
4. To limit the list, select the protocol used in the component.
5. Click **Unlock Selected**.

Modify the Timeout Value for a CM Session

When you configure your environment using CM, a session times out if it is idle for 30 minutes.

To change the timeout value:

1. Open web.xml in *install_dir*\conf\jetty\webservices\webapps\SSPDashboard\WEB-INF.
2. Change the following parameter to define how long to wait before a session times out.

```
<session-timeout>30</session-timeout>
```

3. Save the file.

Uninstall CM from UNIX or Linux

When you uninstall CM, configuration files and logs remain in the *SSPCM_install_dir*/conf, *SSPCM_install_dir*/logs, and apps/jetty/JettyConfigDef.xml directories.

To remove CM:

1. Stop CM.
2. Navigate to the *SSPCM_install_dir*/UninstallerData directory.
3. Type the following command, and press **Enter**:

```
Uninstall_Sterling_Secure_Proxy_Configuration_Manager_V3.3.01
```

Uninstall CM from Windows

When you uninstall CM, configuration files and logs files remain in the *SSPCM_install_dir*\conf and *SSPCM_install_dir*\logs directories.

To remove CM:

1. Stop CM.
2. Click **Start > Programs > Sterling Secure Proxy V3.3.01**.
3. Click **Uninstall Configuration Manager**.
4. Click **Uninstall**.
5. Click **Done**.

Manage SSP Engines



Use the procedures in this section to manage an engine.

View Configured Engines

Use the monitoring function in CM to view all configured engines.

To view configured engines:

1. Click **Monitoring** from the menu bar.
2. Click **Engine Status (All)**. A list of all configured engines is displayed, including the status. Status is displayed as follows:

- ◆  Engine is running
- ◆  Engine is not running

The following information is displayed for each engine:

- ◆ Engine Name
- ◆ Last Configuration Pushed
- ◆ Message
- ◆ CM Version
- ◆ Eng. Version

Change the Engine Passphrase on UNIX or Linux

To change the passphrase defined for the engine at installation:

1. Navigate to *install_dir/bin*, where *install_dir* is the engine installation directory.
2. To stop the engine, type the following command and press **Enter**:

```
./stopEngine.sh
```

3. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
4. Type the following command and press **Enter**:

```
./changePassphrase.sh
```

5. Type the current passphrase and press **Enter**.
6. Type a new passphrase and press **Enter**. Retype the new passphrase and press **Enter**.
Your new passphrase is effective the next time you start the engine.

Change the Engine Passphrase on Windows

To change the passphrase defined for the engine at installation:

1. Click **Start>Run>Browse**.
2. Double-click the following file in the *install_dir\bin* directory:

```
stopEngine.bat
```

3. Click **OK**.
4. Type the engine passphrase and press **Enter**.

The engine stops.

5. Double-click the following file:

```
changePassphrase.bat
```

6. Click **OK**.
 7. Type the current passphrase and press **Enter**.
 8. Type the new passphrase and press **Enter**. Retype the new passphrase and press **Enter**.
- Your new passphrase is effective the next time you start the engine.

Configure the Refresh Interval Between CM and Engines

The Engine Status Page provides information on engines, including when configuration files are pushed to the engine and the version of the files at CM and at the engine. CM polls engines every 30 seconds and updates the information displayed in the Monitoring display.

To change how often CM polls its engines for status information:

1. Click **Monitoring** from the menu bar.
2. Click **Engine Status (All)**. A list of all configured engines is displayed.
3. Type how often to poll engines in the **Refresh Interval** (secs) field.
4. Click **Save**.

Note: The new polling interval is not implemented immediately. The previous polling interval must expire before the new value is implemented. For example, if the polling interval is 50 seconds and you change the value to 15 seconds, the new value of 15 seconds is implemented after 50 seconds.

Update the Monitor Display of Engine Information

Use the Engine Status Page for information on engines, including when configuration files were pushed to the engine, the version of the configuration files at CM and at the engine. CM polls engines every 30 seconds and updates information displayed in the Monitoring display. Use this procedure to immediately poll all engines and update the information displayed.

To poll all engines and obtain configuration information:

1. Click **Monitoring** from the menu bar.
2. Click **Engine Status (All)**. A list of all configured engines is displayed.
3. Click **Refresh**.

Manually Send a Configuration File to an Engine

Adapters are configured at CM. The configuration is then sent to the engine the next time CM polls it. The version of the configuration file saved at the engine and the version at CM is displayed. The version should be the same at the engine and CM. If not, either wait for CM to poll the engine or manually push the configuration file to the engine. The engine must be running to push a configuration file.

Note: Only one CM can be used to configure an engine. If you attempt to send configuration files to an engine from more than one CM, you generate configuration errors.

To manually send the configuration file to an engine:

1. Click **Monitoring** from the menu bar.
2. Click **Engine Status (All)**. A list of all configured engines is displayed.
3. Select the engine where you want to push a configuration.
4. Click **Push Config**.

Change the Listen Port for an Engine

You can change the listen port defined at installation on Windows, UNIX, or Linux. Stop the engine before you change the listen port. Refer to *Change the Engine Passphrase on UNIX or Linux* on page 245 for instructions.

To change the listen port on an engine:

1. From *install_dir/bin*, where *install_dir* is the engine installation, type the following command:

```
configureAcceptor port=nnnn
```

2. Type the system passphrase.

If the command is successful, a response similar to the following is displayed:

```
Acceptor configuration updated:
name      = Secure
secure   = true
port     = nnnn
address  = (default)
timeout  = 30000
enabled  = true
```

Changes to the listen acceptor ports take effect the next time the engine is started.

Change the IP Address for an Engine

If you have multiple NIC cards on an engine, you can route traffic through the IP addresses associated with them. For each NIC card, perform this procedure to associate the IP address of the NIC card with an engine.

After you change the IP address for an engine, create an engine definition that uses the same IP address. Refer to *Install or Upgrade SSP on UNIX or Linux* on page 55 or *Install or Upgrade SSP on Windows* on page 67 for instructions.

To specify the IP bind address of the NIC card:

1. From `\install_dir\bin` where `install_dir` is the engine directory, type the following command:

```
configureAcceptor address=IPaddress
```

2. Type the system passphrase for the engine and press **Enter**.

If the command is successful, a response similar to the following is displayed:

```
Acceptor configuration updated:
name      = Secure
secure   = true
port     = nnnn
address  = (default)
timeout  = 30000
enabled  = true
```

Change the Logging Level for an Engine

When you configure an engine, the logging level for the engine is set to Error by default. Error logging level writes all error messages for the engine to the log.

To change the logging level for an engine:

1. Click Configuration from the menu bar.
2. Highlight the engine to modify.
3. Click the Advanced tab.
4. Select the logging level in the Engine Logging Level field.
5. Click Save.

Uninstall the Engine from UNIX or Linux

When you uninstall CM or the engine, configuration files and logs remain in the *SSPEngine_install_dir/conf* and *SSPEngine_install_dir/logs* directories.

To remove the engine:

1. Stop the engine.
2. Navigate to the *SSPEngine_install_dir/UninstallerData* directory.
3. Type the following command, and press **Enter**:

```
Uninstall_Sterling_Secure_Proxy_Engine_V3.3.01
```

Uninstall the Engine from Windows

When you uninstall the engine, configuration files and log files remain in the *install_dir/conf* and *install_dir/logs* directories. The file *apps\jetty\JettyConfigDef.xml* remains.

To remove the engine:

1. Stop the engine.
2. Click **Start > Programs > Sterling Secure ProxyV3.3.01**.
3. Click **Uninstall Engine**.
4. Click **Uninstall**.
5. Click **Done**.

Modify the Heap Size

If you determine that your system is running slowly, you can change the heap size on either the engine or the configuration manager (CM) to improve performance.

Modify Engine Heap Size

When you install the SSP engine, the heap size is set to a default size of 512MB.

Follow these instructions for changing the engine heap size.

Modify Engine Heap Size on UNIX or Linux

To modify the engine heap size on UNIX or Linux:

1. From *install_dir*/bin, open the startEngine.sh file.
2. Modify the MAXHEAP parameter:
3. Save the file.

Modify Engine Heap Size on Windows

If you run the engine as a Windows service, modify the engine heap size as follows:

1. From the *install_dir*\bin directory, open the SSPengine\$.lax file.
2. Modify the following parameter to the preferred value:

```
lax.nl.java.option.java.heap.size.max=536870912
```

3. Save the file.

If you run the engine from the command line, modify the engine heap size as follows:

1. From the *install_dir*\bin directory, open the startEngine.bat file.
2. Modify the MAXHEAP parameter to the preferred value:

```
MAXHEAP=256m
```

3. Save the file.

Modify Configuration Manager Heap Size

When you install the CM, the heap size is set to 268435456 (256 MB). If your system is slow, you can modify the heap size to improve performance.

Modify the CM Heap Size on UNIX or Linux

To modify the CM heap size on UNIX or Linux:

1. From the *install_dir*/bin directory, open the startCM.sh file.

2. Modify the following line to increase the heap size:

```
lax.nl.java.option.java.heap.size.max=268435456
```

3. Save the file.

Modify the CM Heap Size on Windows

If you run the CM as a Windows service, modify the CM heap size as follows:

1. From the `install_dir\bin` directory, open the `SSPcm$.lax` file.
2. Modify the following parameter to the preferred value:

```
lax.nl.java.option.java.heap.size.max=268435456
```

3. Save the file.

If you run the CM from the command line, modify the CM heap size as follows:

1. From the `install_dir\bin` directory, open the `startCM.bat` file.
2. Modify the following line to increase the heap size:

```
lax.nl.java.option.java.heap.size.max=268435456
```

3. Save the file.

Manage Adapters

After you configure adapters and are in a production environment, use the procedures in this chapter to monitor adapter activity and stop or start an adapter.

Monitor Configured Adapters

Use the monitoring function in CM to view and monitor adapters configured for an engine.

To view and monitor adapters:

1. Click **Monitoring** from the menu bar.
2. Expand the Engine Status (All) tree.
3. Click the engine where the adapters you want to monitor is running.

The following information about each adapter is displayed:

- ◆ Adapter Name
- ◆ Type
- ◆ Port
- ◆ Message

Stop an Adapter from CM

To stop an adapter from CM:

1. Click **Monitoring** from the menu bar.
2. Expand the Engine Status tree.
3. Click the engine where the adapter is running.
4. Select the adapter to stop and click **Stop**.

Start an Adapter from CM

To start an adapter:

1. Click **Monitoring** from the menu bar.
2. Expand the Engine Status tree.
3. Click the engine where the adapter is defined.
4. Select the adapter to start and click **Start**.

Manage User Accounts and Passwords

Two types of user accounts can be created in SSP: CM user accounts and SSP engine user accounts. CM user accounts control access to the SSP user interface. Engine user accounts control which users can send data through SSP. Password policies can be associated with both a CM user account and engine user account to help enforce your company's security policies. Some of the options in the password policy do not apply to engine users. CM user accounts also include role-based security to provide varying levels of access to users within the organization. SSP can be configured to perform user authentication based on information defined in a user account.

Use the information in the following topics to manage password policies and user accounts:

- Manage Password Policies
- Manage CM User Accounts
- Manage User Stores and User Accounts

Manage Password Policies

Password policies are sets of security decisions you make and apply to different user accounts according to security policies in your company. These choices include items such as the number of days a password is valid and the maximum and minimum length of a password.

Use password policies to streamline security operations when adding new users. Instead of adding individual policies for each user, you create one password policy and apply it to all users who require the same access.

A password policy is applied to a new user or when the password is changed on an existing user.

You can apply a password policy only to internal user accounts. This provides you the greatest flexibility in maintaining security policies.

For example, a password policy named Test may have the following password settings:

- Valid for 10 days
- Requires a minimum of 10 characters and maximum of 20 characters
- Requires default password change after the initial log in
- Maintains three passwords in history so the user cannot reuse them
- Must use at least two special characters

In this example, the system administrator gives the user a user name and password. The user logs in to SSP and is prompted to change the password. If the user fails to provide a password with at least 10 and no more than 20 characters, or without at least two special characters, SSP prompts the user for corrections. After all conditions in the password policy are met, the new password is saved and the user is allowed access.

Each user account can have only one password policy associated with it, but one password policy can be applied to multiple user accounts.

Create a Password Policy

You create a password policy to assign to user accounts. You do not have to associate a password policy with a user account, but doing so helps manage your security by streamlining your security operations. A user account can have only one password policy.

To create a password policy:

1. Click Advanced from the menu bar.
2. Click Actions > New Password Policy.
3. Specify values for the following:
 - ◆ Password Policy Name (no spaces allowed)
 - ◆ Days Valid
 - ◆ Minimum Length
 - ◆ Maximum Length
 - ◆ Keep in History
4. To enforce the policy of using at least two special characters in passwords, enable Must contain special characters.
5. Click Save.

You can now edit and delete password policies and assign them to user accounts.

Edit a Password Policy

To edit a password policy:

1. Click Advanced from the menu bar.
2. Expand the Password Policies tree.
3. Click the password policy to edit.
4. Edit the values you want to change. You cannot edit the policy name.
5. Click Save.

Copy a Password Policy

To copy a password policy:

1. Click Advanced from the menu bar.
2. Expand the Password Policies tree.
3. Click the password policy to copy.
4. Click Actions > Copy Selected.
5. Type a name for the new policy.
6. Edit any values you want to change.
7. Click Save.

Delete a Password Policy

To delete a password policy:

1. If the password policy is associated with a user:
 - a. Click Credentials from the menu bar.
 - b. Expand the User Stores tree.
 - c. Select the user store that contains the user definition.
 - d. Select the user to edit and click Edit.
 - e. Remove the password policy to delete from the Password Policy ID field.
 - f. Click OK.
 - g. Click Save.
2. Click Advanced from the menu bar.
3. Expand the Password Policies tree and click the password policy to delete.
4. Click Actions > Delete Selected.
5. Click Delete.

Manage CM User Accounts

CM accounts are assigned a user role: Admin or Operator. Admin users can create and update user accounts and have full access to all configuration options in CM. Operator users have read-only access to accounts and cannot access system functions. Operator users can, however, change their passwords from the login screen.

In addition to role-based security, you can assign password policies to user accounts. Use the default CM user account called admin access CM to create user accounts.

This section includes the following procedures:

- Create a CM User Account
- Edit a CM User Account
- Copy a CM User Account
- Delete a CM User Account

Create a CM User Account

To create a CM user account:

1. Click System from the menu bar.
2. Click Actions > New CM User.
3. Specify the following values for the user account:
 - ◆ User Name (no spaces allowed)
 - ◆ Password
 - ◆ Confirm Password

4. Select the user role to assign to the user account from the User role list: Admin or Operator.
5. To enforce a password policy for this account, select a password policy from the list.
6. To require that the user change the password after the first logon, enable Password Requires change.
7. Click Save.

Edit a CM User Account

To edit a CM user account:

1. Click System from the menu bar.
2. Expand the CM Users tree.
3. Select the user account to edit.
4. Edit the user properties as needed. The User Name cannot be edited.
5. Click Save.

Copy a CM User Account

You can copy a CM user account to create a new user account.

To copy an account:

1. Click System from the menu bar.
2. Expand the CM Users tree.
3. Select the user account to copy and click Actions > Copy Selected.
4. Type a name for the account.
5. Edit the user properties as needed.
6. Click OK.
7. Click Save.

Delete a CM User Account

You can delete a CM user account as needed to maintain the security of SSP.

To delete a user account:

1. Click System from the menu bar.
2. Expand the CM Users tree.
3. Select the user account to delete.
4. Click Actions > Delete Selected.

Manage User Stores and User Accounts

Create user accounts for users who need to access SSP for file transfer. You can create SSP user accounts in the default user store, `defUserStore`, or you can create a new user store to manage groups of users.

For users who communicate using the SSH protocol and who use multiple keys to authorize users, identify the key store where keys are stored and the user record containing the key.

Before you begin:

If you plan to use password policies for user accounts, configure the password policies prior to configuring user accounts.

If you plan to perform local user authentication using SSH keys for SFTP inbound connections, import SSH keys into the SSH key stores. For more information on importing keys into the SSH key stores, see [Manage SSH Keys for SFTP Transactions](#).

This section includes the following procedures:

- Create a User Store
- Copy a User Store
- Delete a User Store
- Create an Engine User Account
- Add SSH Keys to a User Account
- Edit an Engine User Account
- Copy an Engine User Account
- Delete an Engine User Account

Create a User Store

To create a user store:

1. Click Credentials from the menu bar.
2. Click Actions > New User Store.
3. Specify a user store name in the User Store Name field.
4. If desired, change the default values for the following fields:
 - ◆ User Lockout Duration
 - ◆ User Lockout Threshold
5. Click New to add a user account to the user store. You must create at least one user account in the user store before you can save it.
6. Specify the following values for the user account:
 - ◆ User Name
 - ◆ Password
 - ◆ Confirm Password
7. To enforce a password policy for this account, select a password policy from the list.

8. If desired, provide the following information for the user:
 - ◆ First Name
 - ◆ Last Name
 - ◆ Email Address
 - ◆ Pager
 - ◆ Manager ID
9. Click OK.
10. Click Save.

Modify the User Account Locking Value in the User Store

A user account is locked if the user tries to log in to SSP and is unsuccessful, the number of times defined in the User Lockout Threshold field. A login is unsuccessful if the user provides an invalid user ID or password. Internal errors, such as a failure by EA to connect to LDAP server, is not a login failure.

To modify the user account locking value:

1. Click Credentials from the menu bar.
2. Expand the User Stores in the left navigation bar.
3. Click the user store name to open.
4. Change the default value for the User Lockout Threshold field.
5. Click OK.
6. Click Save.

Copy a User Store

To copy a user store:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Select the user store to copy.
4. Click Actions > Copy Selected.
5. Type a name for the new user store.
6. Edit the properties as needed.
7. Click Save.

Delete a User Store

To delete a user store:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.

3. Select the user store to delete. The default user store cannot be deleted.
4. Click Actions > Delete Selected.
5. Click Delete.

Create an Engine User Account

Create a user account to provide access to the engine. To create an engine user account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store to which you want to add a user account.
4. Click New.
5. Specify the following values for the user account:
 - ◆ User Name (no spaces allowed)
 - ◆ Password
 - ◆ Confirm Password
6. To enforce a password policy for this account, select a password policy from the list.
7. Click OK.
8. Click Save.

Add SSH Keys to a User Account

To perform local user authentication for a user account that will be used to access SSP for SFTP connections, you can associate SSH keys with that account.

To add SSH keys to a user account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store to where the user account is stored.
4. Select the user account to add the SSH key to and click Edit.
5. Click the Advanced tab.
6. Select an SSH Authorized User Key Store from the list or click + to create a new User Key Store. Refer to *Create a User Store* on page 259.
7. Select the SSH Authorized User Keys that can be used by this user. Use Shift + Ctrl to select multiple keys.
8. Click OK.
9. Click Save.

Edit an Engine User Account

To edit an engine user account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store to edit.
4. Select the user account to edit and click Edit.
5. Edit the user properties.
6. Click OK.
7. Click Save.

Copy an Engine User Account

You can copy an engine user account to create a new user account.

To copy an account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store that contains the user account to copy.
4. Select the user account to copy and click Copy.
5. Type a name for the account.
6. Edit the user properties as needed.
7. Click OK.
8. Click Save.

Delete an Engine User Account

You can delete a user account as needed to maintain the security of SSP.

To delete a user account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store that contains the user account to delete.
4. Select the user account to delete.
5. Click Delete.
6. Click Save.

Configure Perimeter Servers to Manage SSP Communications

A perimeter server is used by SSP to manage inbound and outbound TCP communication. This software tool enables you to manage the communications flow between outer layers of your network and the TCP-based transport adapters. Perimeter servers can be used to restrict areas where TCP connections are initiated: from more secure areas to less secure areas.

During the SSP installation, a perimeter server is installed. This perimeter server is referred to as the local perimeter server. You can use this default local perimeter server to restrict connections or you can install other perimeter server instances as needed. You can install additional perimeter servers on different computers or you can install different instances on the same computer, if you want to use different network cards for inbound and outbound traffic. A perimeter server requires a perimeter server definition in SSP.

After you install and configure a remote perimeter server, you need to map how the perimeter server is used: inbound, outbound, or External Authentication. Refer to *Map Perimeter Servers* on page 271.

Before you configure remote perimeter servers in SSP, complete the installation procedures outlined in *Install a Remote Perimeter Server*.

Topics include:

- Typical Installation

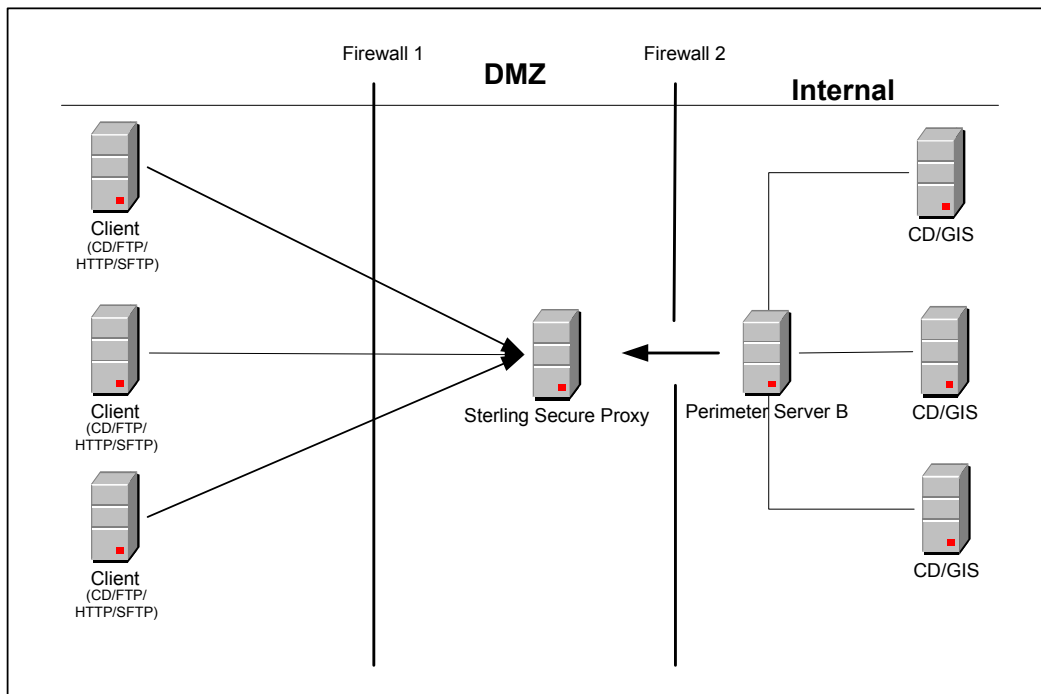
- Define a remote perimeter Server for a Less Secure Environment

- Configure and Edit a remote perimeter Server Definition When Installed in a More Secure Network

- Map Perimeter Servers

Typical Installation

The following figure illustrates a typical SSP installation with perimeter servers:



The preceding figure shows the following:

1. The persistent connection is established from the perimeter server in the internal trusted network to SSP in the DMZ. This allows for only an outbound hole to be configured in the Firewall 2 (no inbound hole is needed with this configuration)
2. SSP has an HTTP server adapter configured for two scenarios, one secure HTTP (HTTPS) and the other non-secure HTTP.
3. Two trading partners with separate host and port numbers are configured to communicate with SSP.

A perimeter server and all adapters that communicate with the local perimeter server must be configured on the same SSP engine. An engine can have more than one perimeter server but a perimeter server can be used by only one engine. You can configure a perimeter server for one trading partner with large files and low transaction volume, and another perimeter server on the same engine for a different trading partner with smaller files and high transaction volume. By configuring each perimeter server according to the trading partner, you increase SSP performance.

Sample remote perimeter Server Configurations

Use remote perimeter servers with SSP if you want to:

Eliminate an inbound hole in your firewall to allow connections from less secure to more secure areas.

Send data to your customers from the perimeter server as the originating IP address.

Use different network cards for inbound and outbound traffic.

Implement multiple DMZ scenarios. You can use perimeter servers in your outer DMZ with SSP in the internal DMZ.

You have flexible deployment options for using perimeter servers with SSP: from a simple IP break to no inbound holes in the firewall. Following are sample deployment options.

Deployment Option Example—Two remote perimeter Servers on a Computer with Two NIC Cards

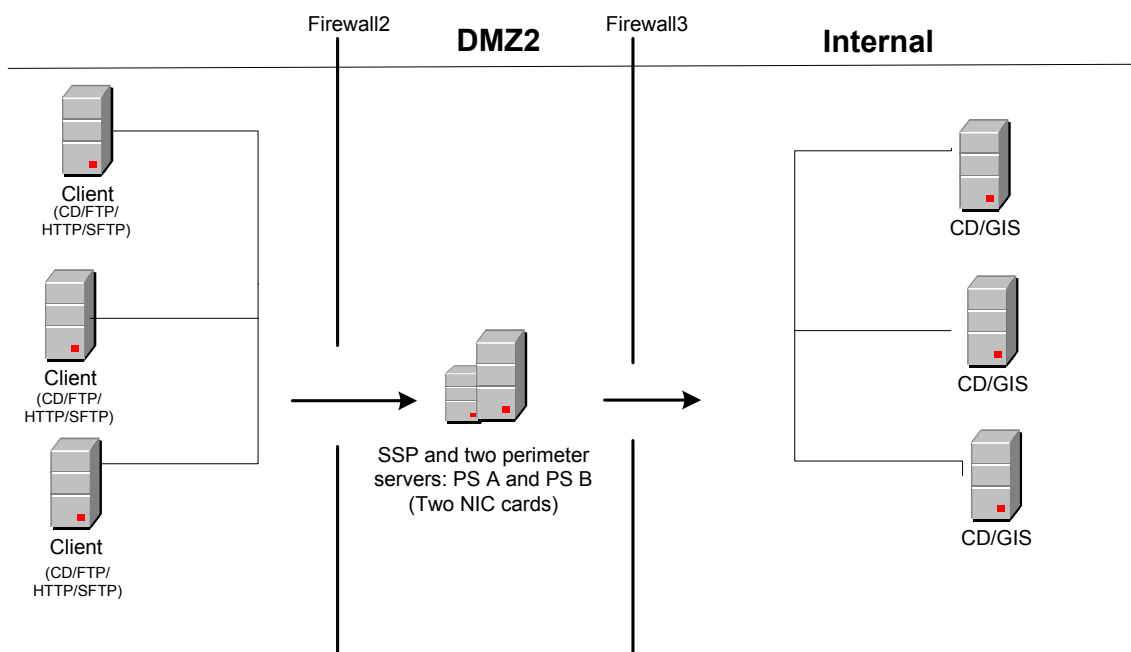
Deployment Option Example—From More Secure to Less Secure

Deployment Option Example—From Less Secure to More Secure

Deployment Option Example —External Authentication Perimeter Server

Deployment Option Example—Two remote perimeter Servers on a Computer with Two NIC Cards

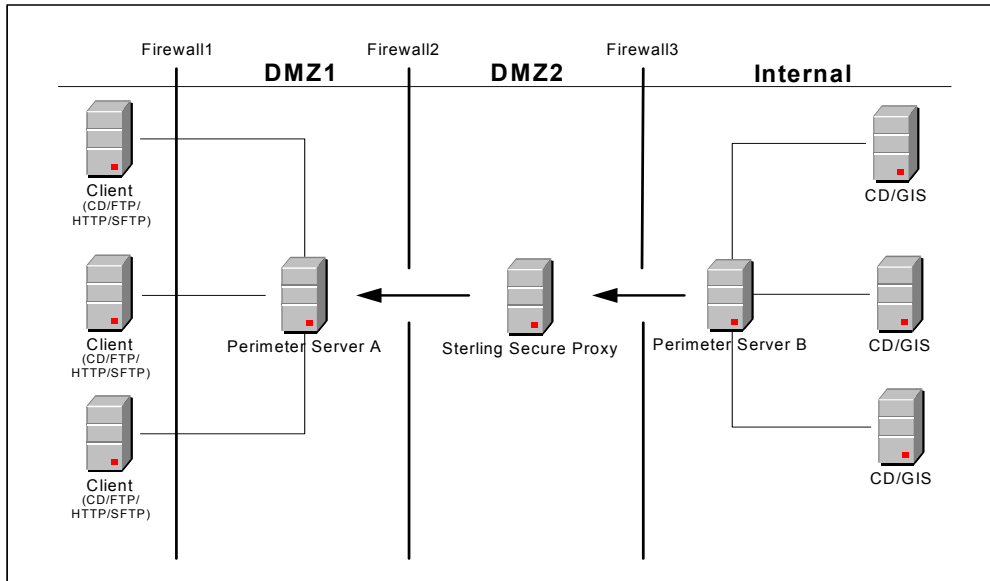
The following sample illustrates a configuration where two remote perimeter servers are installed. One remote perimeter server manages inbound traffic and the other manages outbound traffic.



In this configuration, the firewall is configured to allow connections from trading partners to the remote perimeter server A. PS A then routes traffic to SSP. Outbound traffic is routed from SSP through PS B to the SI or Connect:Direct server.

Deployment Option Example—From More Secure to Less Secure

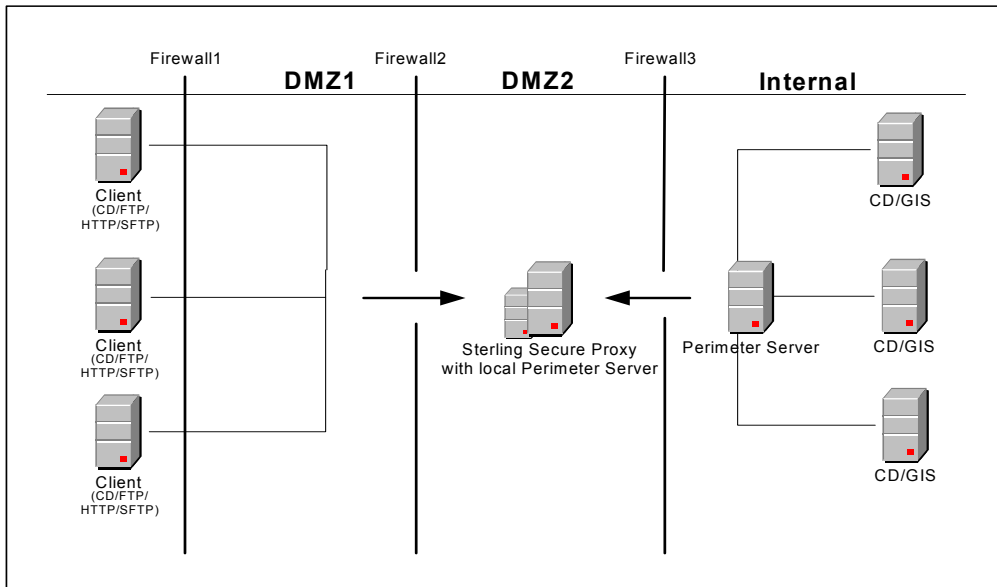
For additional firewall security, you can install a perimeter server in a more secure area than SSP and set up your firewall to allow only connections initiated from a more secure area to a less secure area. The following diagram demonstrates a configuration with two perimeter servers:



In this example, SSP is configured to use two perimeter servers that reside on remote computers: one on an external network (Perimeter Server A), and one in the internal network (Perimeter Server B). The firewalls are configured to allow only connections initiated from inside a more secure area (only an outbound hole in the firewall). When SSP is started, a connection is established from Perimeter Server B to SSP and from SSP to Perimeter Server A. Through these communication lines, SSL/TLS sessions can be established between clients and SSP, and between SSP and Connect:Direct or SI.

Deployment Option Example—From Less Secure to More Secure

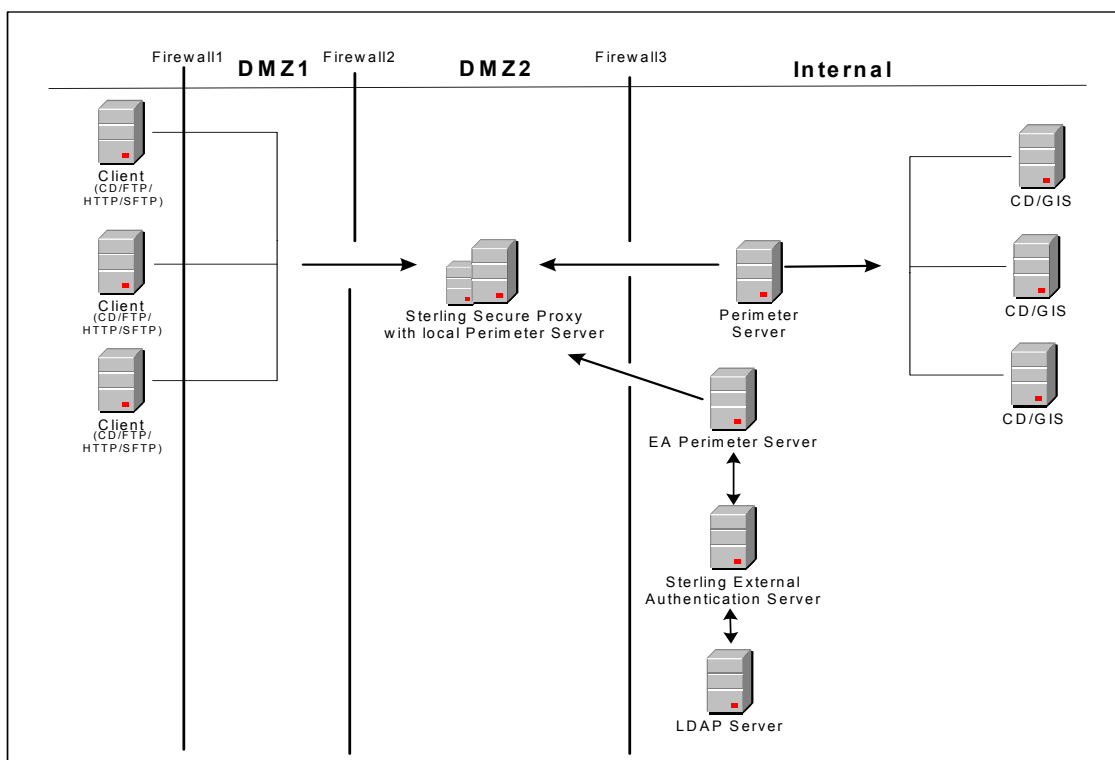
The following sample illustrates a configuration where the remote perimeter server is installed in a more secure location than SSP:



In this configuration, the firewall is configured to allow connections from external trading partners to SSP and from the internal perimeter server to SSP. In this configuration, traffic is moving from a less secure to a more secure location. To restrict unauthorized access, you can limit the perimeter server to perform only those activities required for SSP operations. Refer to *Install a Remote Perimeter Server* on page 73 for more information.

Deployment Option Example —External Authentication Perimeter Server

The following sample illustrates a configuration where the remote perimeter server in the trusted zone is used to connect to EA:



In this example, SSP is configured to use two remote perimeter servers and the local perimeter server. When SSP is started, a connection is established from SSP to the remote perimeter server. Through this communication line, SSL/TLS sessions can be established between clients and SSP. Another remote perimeter server is used to communicate between EA and SSP.

Define a remote perimeter Server for a Less Secure Environment

A common network configuration pattern is for SSP to reside in the innermost, secure network zone and the perimeter server to reside in the DMZ. In this case the connection should be established from SSP to the perimeter server—that is, from the more secure towards the less secure network zone.

This section contains the following procedures:

- Configure a remote perimeter Server in a Less Secure Zone

- Edit a remote perimeter Server in a Less Secure Zone Definition

- Modify the Water Mark Values and Local Host Information of a remote perimeter Server in a Less Secure Zone

Configure a remote perimeter Server in a Less Secure Zone

To configure a perimeter server in a less secure zone:

1. Select Advanced from the menu bar.

2. Select Actions > New Perimeter Server > Less Secure Zone.
3. Specify the following values:
 - ◆ Perimeter Server Name
 - ◆ Perimeter Server Host
 - ◆ Perimeter Server Port
4. Click Save.

Edit a remote perimeter Server in a Less Secure Zone Definition

To edit the definition of a perimeter server in a less secure zone:

1. Select Advanced from the menu bar.
2. Expand the Perimeter Servers tree and expand the Less Secure Zone tree.
3. Click the perimeter server definition you want to edit.
4. Edit the values as needed.
5. Click Save.

Modify the Water Mark Values and Local Host Information of a remote perimeter Server in a Less Secure Zone

To modify the water mark values and local host information of a perimeter server in a less secure zone:

1. Select Advanced from the menu bar.
2. Expand the Perimeter Server tree and expand the Less Secure Zone tree.
3. Select the perimeter server to edit.
4. Click the Advanced tab.
5. Change the following values as needed:
 - ◆ Perimeter Server Outbound Low Water Mark
 - ◆ Perimeter Server Outbound High Water Mark
 - ◆ Perimeter Server Inbound Low Water Mark
 - ◆ Perimeter Server Inbound High Water Mark
 - ◆ Proxy Local Interface
 - ◆ Proxy Local Port
6. From the Perform DNS Resolution list, select the place where DNS resolution occurs.
7. Click Save.

Configure and Edit a remote perimeter Server Definition When Installed in a More Secure Network

In some cases, it is desirable for SSP to communicate with a perimeter server installed in a more secure network zone. In this case establish the network connection from the perimeter server to SSP.

This section contains the following procedures:

Configure a remote perimeter Server in a More Secure Zone

Edit A More Secure Zone remote perimeter Server Definition

Modify Water Mark Values and Local Host Information of a remote perimeter Server Installed in a More Secure Zone

Configure a remote perimeter Server in a More Secure Zone

To configure a perimeter server in a more secure zone:

1. Select Advanced from the menu bar.
2. Select Actions > New Perimeter Server > More Secure Zone.
3. Specify the following values:
 - ◆ Perimeter Server Name
 - ◆ Proxy Local Listen Port
4. Click Save.

Edit A More Secure Zone remote perimeter Server Definition

To edit a more secure zone perimeter server definition:

1. Select Advanced from the menu bar.
2. Expand the Perimeter Servers tree and expand the More Secure Zone tree.
3. Click the perimeter server definition to edit.
4. Edit the values as needed.
5. Click Save.

Modify Water Mark Values and Local Host Information of a remote perimeter Server Installed in a More Secure Zone

To modify water mark values and local host information of a perimeter server in a more secure zone:

1. Select Advanced from the menu bar.
2. Expand the Perimeter Server tree and expand the More Secure Zone tree.
3. Select the perimeter server to edit.
4. Click the Advanced Tab.
5. Change the following values as needed:
 - ◆ Perimeter Server Outbound Low Water Mark
 - ◆ Perimeter Server Outbound High Water Mark
 - ◆ Perimeter Server Inbound Low Water Mark
 - ◆ Perimeter Server Inbound High Water Mark
 - ◆ Proxy Local Interface

6. From the Perform DNS Resolution list, select the place where DNS resolution occurs.
7. Click Save.

Map Perimeter Servers

After you configure perimeter servers, map how they are used by each adapter: inbound perimeter server, outbound perimeter server, or EA perimeter server.

To map perimeter servers:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and select the adapter you want to edit.
3. Click the Advanced tab.
4. Select a perimeter server for each of the following as needed. The default is local.
 - ◆ Inbound Perimeter Server
 - ◆ Outbound Perimeter Server
 - ◆ External Authentication Perimeter Server
5. Click Save.

Repeat this process for each adapter that uses a remote perimeter server.

Note: If you change the perimeter server mapped to an adapter, you must restart the adapter and the perimeter server before the change is enabled.

Modify Perimeter Server Properties

Two property values are defined in the `perimeter.properties` file located in the `install_dir/bin` folder. These properties determine SSL Session caching. Modify the following properties as necessary:

Parameter	Description
<code>SslSessionDatabaseTimeoutSeconds</code>	How long a cached SSL session is valid. The valid range is 30 seconds to 24 hours (60*60*24 = 86400 seconds). Default=1 hour or 3600 seconds.
<code>SslSessionDatabaseSize</code>	Maximum number of sessions to cache. This parameter is used by FTP and HTTP reverse proxy adapters. SSL sessions are not cached for Connect:Direct proxy adapters. Valid range is 1024 to 16384. Default=4096.

Configure SSP for Sterling External Authentication Server (EA)

To provide a more advanced method of securing an inbound or outbound connection to SSP, use Sterling External Authentication Server (EA). EA allows you to authenticate certificate information or user credentials presented by the inbound node or to perform user ID and password mapping for the credentials used to attach to the outbound node.

EA Server Configuration - Worksheet

Before you begin configuring SSP for authentication options using EA, gather the information on this worksheet from the EA administrator. Collect this information for each EA server you will configure.

Configuration Manager Field	Value
EA Server Name	
EA Server Address	
EA Server Port	
Outbound Port Range	
Security Setting	_____ (SSL or TLS)
Trust Store	
CA/Trusted Certificates	
Key Store	
Key/System Certificate	
Cipher Suites	

Configure an EA Server Connection

You can use EA to increase the security of your SSP environment. EA can be used to validate certificates from an inbound node, authenticate inbound users, and provide more secure credentials to the outbound node.

Before you can configure SSP to use EA, you must configure an EA server definition.

To configure an EA server definition:

1. Click Advanced from the menu bar.
2. Select Actions > New External Authentication Server.
3. Specify values for the following fields:
 - ◆ EA Server Name

- ◆ EA Server Address
 - ◆ EA Server Port
 - ◆ Outbound Port Range
4. To enable SSL or TLS for the EA server connection, click the Security tab and enable Use Secure Connection.
 5. Set the following values:
 - ◆ Security Setting
 - ◆ Trust Store
 - ◆ CA/Trusted Certificates
 - ◆ Key Store
 - ◆ Key/System Certificate
 - ◆ Cipher Suites
 6. Click Save.

Specify Alternate EA Servers for Failover Support

You can specify alternate EA servers that SSP connects to if a connection to the primary EA server cannot be made. Up to three alternate EA servers can be defined for each EA server.

You must first configure an EA server connection for each EA server you want to identify for failover support. Then you can identify alternate EA servers to use if an EA server is not available by selecting an EA server definition from the list.

To specify an alternate EA server for failover support:

1. Click Advanced from the menu bar.
1. Expand the External Authentication Servers tree and select the EA server you want to edit.
2. Click the Advanced tab.
3. Select an alternate server from the Alternate EA Server #1 list.
4. Select additional servers as needed from the remaining lists. Connection attempts will be made to the alternate servers in the order in which they are specified.
5. Click Save.

Use a Perimeter Server to Connect to EA

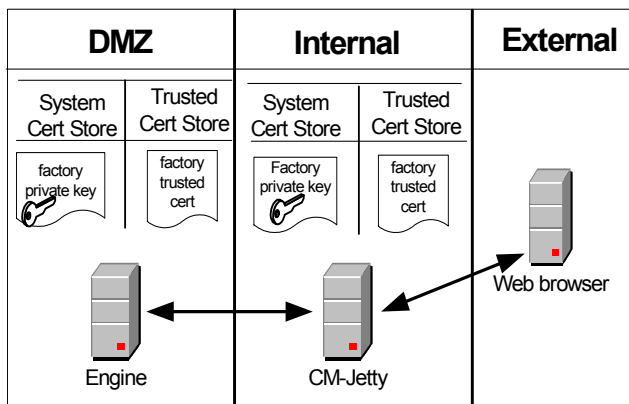
You can configure EA to use a remote perimeter server in the trusted zone to manage connections to and from EA. This configuration enables you to have one outbound opening in your more trusted firewall. For more information on configuring and mapping perimeter servers, refer to *Configure Perimeter Servers to Manage SSP Communications* on page 263.

Manage Certificates Between SSP Components

To maintain security in SSP, the engine and Configuration Manager (CM) communicate using SSL. SSP uses TCP/IP communications links between the web browser and the Jetty web server, the web server and CM, and CM and the engine. The only link that can be unsecure is between the web browser and the Jetty web server.

When you install SSP, a default certificate is installed to allow you to communicate. All components of the SSP system including CM, engine, and the Jetty web server share the same certificate. This self-signed certificate is called the factory certificate and has a ten year expiration.

Before you can begin production, you must import a secure certificate. The default configuration uses a single key to secure the connection between the engine and CM. The certificate distribution looks like this:



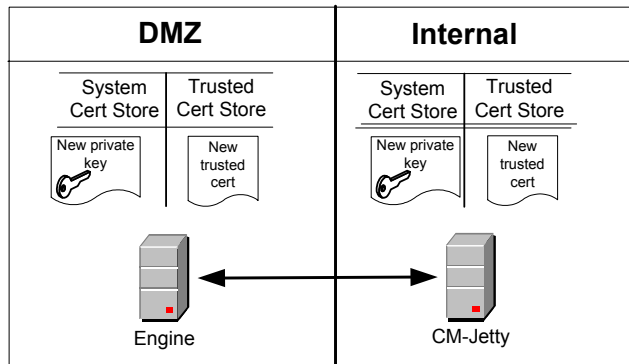
To secure the communication between these components, replace the factory certificates using one of the models in this chapter.

Topics include:

- Use a Common Certificate for the Engine and CM
- Use Different Certificates for the Engine and CM
- Restore Factory Certificates

Use a Common Certificate for the Engine and CM

The simplest way to update the certificate distribution is to replace the factory certificate with a new certificate and use that certificate for both the engine and CM. The certificate distribution looks like this:



Following are the procedures to replace a factory certificate with a common certificate:

Replace the Factory Certificate with a Common Certificate on UNIX or Linux

Replace the Factory Certificate with a Common Certificate on Windows

Replace the Factory Certificate with a Common Certificate on UNIX or Linux

To replace the factory certificate used between the engine and CM on UNIX or Linux:

1. Stop CM.
 - a. At CM, navigate to the *install_dir/bin* directory, where *install_dir* is the installation directory, and type the following command:
2. Type the following command to replace the factory certificate:

```
./stopCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
- c. Type the user name and password for the administrator.

```
./configureCmSsl.sh -u commonCert=<cert file> commonCertAlias=<alias>
```

where:

- ◆ *<cert file>* is the path and file name to the certificate that replaces the factory certificate.
- ◆ *<alias>* is an alias name for the new certificate. This can be any value other than **factory**. If you do not specify an alias, **common** is assigned as the default.

3. Type the following command to create an export file of the certificate store:

```
<export file>
```

where *<export file>* is the path and file for the export file.

4. Copy the file you created in step 3 to the engine.
5. Stop the engine.
 - a. At the engine, navigate to the *install_dir/bin* directory and type the following command:

```
Engine.sh
```

- b. Type the passphrase defined for the engine and press **Enter**.
6. At the engine, navigate to the *install_dir/bin* directory and type the following command to import the certificate store created in step 3:

```
./configureEngineSsl.sh -i file=<export file> engCertAlias=<alias>
```

where:

- ◆ *<export file>* is the path and file for the export file.
 - ◆ *<alias>* is the alias name for the new certificate assigned in step 2
7. Start the engine.
 - a. From *install_dir/bin*, type the following command:

```
./startEngine.sh
```

- b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.
8. Start CM.
 - a. Navigate to the *install_dir/bin* directory and type the following command:

```
./startCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Replace the Factory Certificate with a Common Certificate on Windows

To replace the factory certificate used between the engine and CM on Windows:

1. Stop CM on Windows from Windows services.
2. Using a command line interface on CM, navigate to the *install_dir\bin* directory.

- Type the following command to replace the factory certificate:

```
configureCmSsl -u commonCert=<cert file> commonCertAlias=<alias>
```

where:

- ◆ *<cert file>* is the path and file name to the certificate that replaces the factory certificate.
 - ◆ *<alias>* is an alias name for the new certificate. This can be any value other than **factory**. If you do not specify an alias, **common** is assigned as the default.
- Type the following command to create an export file of the certificate store:

```
configureCmSsl -e file=<export file>
```

where *<export file>* is the path and file for the export file.

- Copy the file you created in step 3 to the engine.
- Stop the engine on Windows from Windows services.
- Using a command line at the engine, navigate to the *install_dir\bin* directory and type the following command to import the certificate store created in step 4.

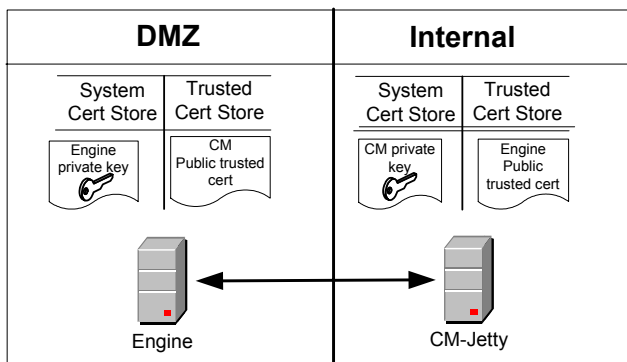
```
configureEngineSsl -i file=<export file> engCertAlias=<alias>
```

where:

- ◆ *<export file>* is the path and file for the export file.
 - ◆ *<alias>* is the alias name for the new certificate assigned in step 4.
- Start the engine on Windows from Windows services.
 - Start CM on Windows from Windows services.

Use Different Certificates for the Engine and CM

You can use different certificates to secure the engine-to-CM connection and to secure the Jetty web server-to-CM connection. This certificate distribution is illustrated below:



Following are the procedures to replace a factory certificate with an engine and a CM certificate:

Replace the Factory Certificate with an Engine Certificate and CM Certificate on UNIX or Linux

Replace the Factory Certificate with an Engine and CM Certificate on Windows

Replace the Factory Certificate with an Engine Certificate and CM Certificate on UNIX or Linux

To replace the factory certificates, with one certificate at the engine and a different certificate at CM on UNIX or Linux:

1. Stop CM.
 - a. Navigate to the CM *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command:

```
./stopCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
 - c. Type the user name and password for the administrator.
2. From *install_dir*/bin, type the following command to replace the factory certificate with a CM certificate:

```
./configureCmSsl.sh -u cmCert=<cert file> cmCertAlias=<alias>
```

where:

- ◆ *<cert file>* is the path and file name to the certificate that replaces the factory certificate for CM.
 - ◆ *<alias>* is the alias name for the new CM certificate. It can be any value other than **factory**. If you do not specify an alias, “cm” is assigned as the default.
3. On the CM computer, type the following command to replace the factory certificate with an engine certificate:

```
./configureCmSsl.sh -u engCert=<cert file> engCertAlias=<alias>
```

where:

- ◆ *<cert file>* is the path and file name to the certificate you want to use to replace the factory certificate for the engine.
 - ◆ *<alias>* is the alias name for the new engine certificate. This can be any value other than **factory**. If you do not specify an alias, “engine” is assigned as the default.
4. Type the following command to create an export file of the certificate store:

```
configureCmSsl -e file=<export file>
```

where *<export file>* is the path and file for the export file.

5. Copy the file you created in step 4 to the engine.
6. Stop the engine.
 - a. On the engine, navigate to the *install_dir*/bin directory and type the following command:

```
./stopEngine.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
7. From the *install_dir*/bin directory, type the following command to import the certificates created in step 2 and step 3.

```
configureEngineSsl -i file=<export file> engCertAlias=<alias>
```

where:

- ◆ *<export file>* is the path and file for the export file.
 - ◆ *<alias>* is an alias name for the engine certificate assigned in step 3. If an *engCertAlias* was omitted in step 3, specify *engine* as the alias.
8. Start the engine.
 - a. Type the following command:

```
./startEngine.sh
```

- b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.
9. Start CM.
 - a. Navigate to the *install_dir*/bin directory and type the following command:

```
./startCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Replace the Factory Certificate with an Engine and CM Certificate on Windows

To replace the factory certificate, with a certificate at the engine and a different certificate at CM on Windows:

1. Stop CM on Windows from Windows services.
2. Using a command line interface at CM, navigate to the *install_dir*\bin directory.

3. Type the following command to replace the factory certificate with a CM certificate:

```
configureCmSsl -u cmCert=<cert file> cmCertAlias=<alias>
```

where:

- ◆ *<cert file>* is the path and file name to the certificate that replaces the factory certificate for CM.
 - ◆ *<alias>* is the alias name for the new CM certificate. It can be any value other than **factory**. If you do not specify an alias, *cm* is assigned as the default.
4. On the CM computer, type the following command to replace the factory certificate with an engine certificate:

```
configureCmSsl -u engCert=<cert file> engCertAlias=<alias>
```

where:

- ◆ *<cert file>* is the path and file name to the certificate you want to use to replace the factory certificate for the engine.
 - ◆ *<alias>* is the alias name for the new engine certificate. This can be any value other than **factory**. If you do not specify an alias, *engine* is assigned as the default.
5. Type the following command to create an export file of the certificate store:

```
configureCmSsl -e file=<export file>
```

where *<export file>* is the path and file for the export file.

6. Copy the file you created in step 5 to the engine.
7. Stop the engine on Windows from Windows services.
8. Type the following command to import the certificates created in step 3 and step 4.

```
configureEngineSsl -i file=<export file> engCertAlias=<alias>
```

where:

- ◆ *<export file>* is the path and file for the export file.
 - ◆ *<alias>* is an alias name for the engine certificate assigned in step 3. If an *engCertAlias* was omitted in step 3, specify *engine* as the alias.
9. Start the engine on Windows from Windows services.
 10. Start CM on Windows from Windows services.

Restore Factory Certificates

Use these procedures to restore the internal certificates used to the factory certificates.

Restore the Factory Certificate on UNIX or Linux

Restore the Factory Certificate on Windows

Restore the Factory Certificate on UNIX or Linux

To restore the certificate distribution to the factory settings on UNIX or Linux:

1. Stop CM.
 - a. Navigate to the *install_dir/bin* directory, where *install_dir* is the installation directory, and type the following command:

```
./stopCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
 - c. Type the user name and password for the administrator.
 2. From the *install_dir/bin* directory, type the following command to restore the factory certificate:

```
./configureCmSsl.sh -r
```

3. Type the following command to export the factory-restored certificate store:

```
./configureCmSsl.sh -e file=<export file>
```

where *<export file>* is the path and file for the export file.

4. Copy the export file to the engine.
5. Stop the engine.
 - a. Navigate to the *install_dir/bin* directory and type the following command:

```
./stopEngine.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
 6. From the *install_dir/bin* directory, type the following command to import the factory-restored certificate store:

```
configureEngineSsl -i file=<export file> engCertAlias=factory
```

where *<export file>* is the path and file for the certificate store.

7. Start the engine.
 - a. Type the following command:

```
./startEngine.sh
```

- b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.
8. Start CM.
 - a. On CM, navigate to the *install_dir/bin* directory and type the following command:

```
./startCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Note: Restoring the configuration to use the factory certificate does not delete the certificates that were previously in use.

Restore the Factory Certificate on Windows

To restore the certificate distribution to the factory settings on Windows:

1. Stop CM on Windows from Windows services.
2. From the *install_dir/bin* directory, type the following command to restore the factory certificate:

```
configureCmSsl -r
```

3. Type the following command to export the factory-restored certificate store:

```
configureCmSsl -e file=
```

where *<export file>* is the path and file for the export file.

4. Copy the export file to the engine.
5. Stop the engine on Windows from Windows services.
6. From the *install_dir/bin* directory, type the following command to import the factory-restored certificate store:

```
configureEngineSsl -i file= engCertAlias=factory
```

where *<export file>* is the path and file for the certificate store.

7. Start the engine on Windows from Windows services.
8. Start CM on Windows from Windows services.

Note: Restoring the configuration to use the factory certificate does not delete the certificates that were previously in use.

Change the Password of the CM Key Store and Trust

The password for the key store and the trust store is set to password at installation.

Change the Password of the CM Key Store and Trust Store on UNIX or Linux

To change the password:

1. Stop CM.
 - a. Navigate to the *install_dir/bin* directory, where *install_dir* is the installation directory, and type the following command:

```
./stopCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
 - c. Type the user name and password for the administrator.
 2. From the *install_dir/bin* directory, type the following command:

```
./configureCmSsl.sh -x
```

3. When prompted, type the existing password and press **Enter**.
 4. Type the new password and press **Enter**.
 5. Start CM.
 - a. From the *install_dir/bin* directory, type the following command:

```
./startCM.sh
```

- b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Change the Password of the CM Key Store and Trust Store on Windows

To change the password:

1. Stop CM on Windows from Windows services.
2. From the *install_dir/bin* directory, type the following command:

```
configureCmSsl -x
```

3. When prompted, type the existing password and press **Enter**.
 4. Type the new password and press **Enter**.
 5. Start CM on Windows from Windows services.

Change the Password of the Engine Key Store and Trust Store

The password for the key store and the trust store is set to password at installation.

Use these procedures to change the password:

Change the Password of the Engine Key Store and Trust Store on UNIX or Linux

Change the Password of the Engine Key Store and Trust Store on Windows

Change the Password of the Engine Key Store and Trust Store on UNIX or Linux

To change the password:

1. Stop the engine.
 - a. Navigate to the *install_dir/bin* directory, and type the following command:

```
./stopEngine.sh
```

- b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
 - c. Type the user ID and password of the administrator.
 2. Using a command line interface on CM, navigate to the *install_dir/bin* directory and type the following command:

```
./configureEngineSsl.sh -x
```

3. When prompted, type the existing password and press **Enter**.
 4. Type the new password and press **Enter**.
 5. Retype the password and press **Enter**.
 6. Start the engine.
 - a. Navigate to the *install_dir/bin* directory on the engine and type the following command:

```
./startEngine.sh
```

- b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.

Change the Password of the Engine Key Store and Trust Store on Windows

To change the password:

1. Stop the engine on Windows from Windows services.
2. Using a command line interface on CM, navigate to the *install_dir/bin* directory and type the following command:

```
configureEngineSsl -x
```

3. When prompted, type the existing password and press **Enter**.
 4. Type the new password and press **Enter**.
 5. Retype the password and press **Enter**.
 6. Start the engine on Windows from Windows services.

Configuration Utilities

Two utilities are used in the previous procedures to configure SSL: `configureCmSsl` and `configureEngineSsl`. Refer to the tables below to identify the functions that can be performed on the engine and CM. You are prompted for a password when one is required.

Use the following functions to configure CM, using the `configureCmSsl` utility:

Parameter	Description
-s	Show current configuration.
-u	Update configuration. Available options include: <ul style="list-style-type: none">◆ <code>commonCert</code>—fully-qualified location of the common certificate to be shared by the SSP components engine, CM, and web server.◆ <code>commonCertAlias</code>—alias for the common certificate and shared by all SSP components. If the certificate file name is omitted, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to <i>common</i>.◆ <code>cmCert</code>—the fully-qualified location of CM and jetty web server certificate.◆ <code>cmCertAlias</code>—alias for the CM/jetty web server certificate. If no file name is provided, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to <i>cm</i>.◆ <code>engCert</code>—the fully-qualified location of the engine certificate.◆ <code>engCertAlias</code>—alias for the engine certificate.◆ <code>webCert</code>—the fully-qualified location of the jetty web server certificate.◆ <code>webCertAlias</code>—alias for the jetty web server certificate. If no file name is provided, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to <i>webserver</i>.◆ <code>cmClientCert</code>—the fully-qualified location of the CM client certificate.◆ <code>cmClientCertAlias</code>—alias for the CM client certificate. If no file name is provided, a certificate with this alias must exist in the key store. This certificate is used by CM to communicate with the engine. If no alias is provided, the value defaults to <i>cmServer</i>.◆ <code>cmServerCert</code>—the fully-qualified location of the CM server certificate.◆ <code>cmServerCertAlias</code>—alias for the CM server certificate. If no file name is provided, a certificate with this alias must exist in the key store. This certificate is used by CM to communicate with the jetty web server. If no alias is provided, the value defaults to <i>cmClient</i>.◆ <code>cmSslProt</code>—the SSL or TLS protocol used for the session between CM and the engine. Valid values are: <i>SSLv2</i>, <i>SSLv3</i>, <i>TLSv1</i>, or <i>SSLv2Hello</i>.◆ <code>cmCiphers</code>—ordered list of cipher suites for communication between CM and the engine. Separate ciphers with a comma, colon, or semicolon.◆ <code>https</code>—identifies if security is enabled between a web browser and the jetty web server. <i>n</i> = disable security, <i>Y</i> = security enabled. <code>https</code> is enabled by default.◆ <code>webHost</code>—the IP bind address for the jetty web server. The default value is <i>localhost</i>. If CM has multiple NIC cards, use the field to specify the IP address of the NIC card to use for the jetty web server.◆ <code>webPort</code>—the listen port for the jetty server. The default value is 8443.

Parameter	Description
	<ul style="list-style-type: none"> ◆ webSslProt—the SSL or TLS protocol for the link between the web browser and the jetty web server. Valid values include SSLV2, SSLV3, TLSv1, or SSLV2Hello. ◆ webCiphers—an ordered list of cipher suites to use on the connection between the web browser and jetty web server. Separate ciphers with a comma, colon, or semicolon. ◆ clientAuth—enables client authentication for web browser clients. n= disabled. y = enabled. This option is set to n by default. If you enable clientAuth, you must add trusted certificates for the web server clients. ◆ trustedCert—fully-qualified location of the trusted certificate for the web client.
-e	Export configuration. The export option is: file—to identify the fully-qualified location of the export file. It can be imported into CM or the engine.
-i	Import configuration. The import option is: file—to identify the fully-qualified location of the export file. It can be imported into CM or the engine.
-d	Delete a certificate. The delete option is: alias—the alias of the certificate to delete. This can be specified multiple times.
-x	Change key store password.
-r	Restore factory settings.
-h	List the usage and parameters of the command.

Use the following functions to configure SSL on the engine, using the configureEngineSsl utility:

Parameter	Description
-s	Show current configuration.
-i	Import configuration. Options include <ul style="list-style-type: none"> ◆ file—the fully-qualified location of the import file. ◆ engCertAlias—the alias for the engine certificate.
-d	Delete a certificate. Options include alias—the alias of the certificate to delete. This can be specified multiple times.
-x	Change key store password.
-h	List the usage and parameters of the command.

Manage SSH Keys for SFTP Transactions

This section describes how to use SSH keys when implementing SFTP communications between SSP and your trading partners and target servers.

Topics include:

- About SSH/SFTP
- SSH Key Implementation Models Using SSP
- Manage Local Host Key Stores and Keys
- Manage Authorized User Key Stores and Keys
- Manage Known Host Key Stores and Keys
- Manage Local User Key Stores and Keys

About SSH/SFTP

SSH/SFTP provides a more secure means than FTP to exchange information with trading partners. During an FTP session, the user name and password are transmitted in clear text. An eavesdropper can easily log this FTP user name and password. Using SSH/SFTP instead of FTP, the entire login session, including transmission of password, is encrypted, making it much more difficult for an outsider to observe and collect passwords. By encrypting all traffic, SSH/SFTP effectively eliminates eavesdropping, connection hijacking, and other network-level attacks.

You can configure SSP to require authentication with a password and public key for SSH/SFTP connections. Authentication for SSH/SFTP connections is performed by the exchange of session keys between the server and the client. This assures that both parties know whom they are exchanging data with.

To implement authentication for SFTP connections, you must create SSH key stores and import SSH keys into them. These key stores and keys can then be selected when you are configuring SSP to support SSH/SFTP connections. Configure the following SSH keys for SFTP communications:

Inbound connections

- ◆ Local Host Key—Private key used by SSP to identify itself to the client
- ◆ Authorized User Key—Public key used by SSP to authenticate the user (optional)

Outbound connections

- ◆ Known Host Key—Public key used by SSP to authenticate the server
- ◆ Local User Key—Private key used by SSP to identify itself to the server during public key user authentication (optional)

Because public key server authentication is mandatory in SSH, you must configure both local host keys and known host keys. Client authentication is performed using a password or public key (or both) in SSH. As a result, authorized user keys and local user keys are required *only* if you plan to

use public key authentication. You can choose different user authentication methods for the inbound and outbound connections.

In Configuration Manager, you must create at least one key within a key store to save the key store definition. You can add as many keys as needed to a key store, and they can be shared between multiple adapters. When you have configured SSH key stores, you can copy them (and the keys within them) to create new key stores.

SSH Key Implementation Models Using SSP

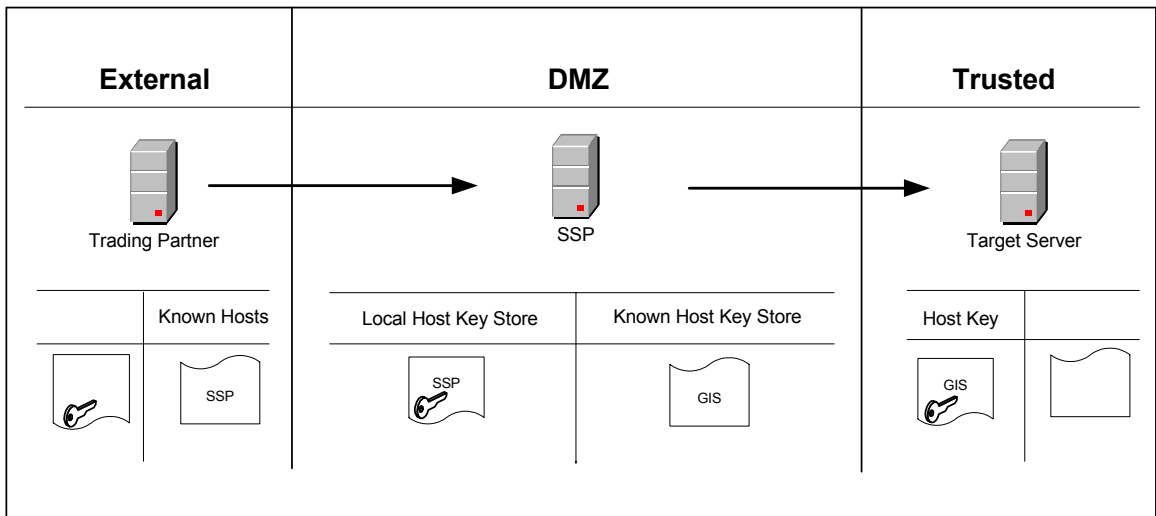
This section presents two models for using SSH keys and shows how to implement the model in SSP.

Use Server Authentication for Inbound and Outbound Connections

Implement Public Key User Authentication for Inbound and Outbound Connections

Use Server Authentication for Inbound and Outbound Connections

In a basic SSH key implementation, you use both local host keys and known host keys for SFTP communications with your trading partner and target server. The key distribution looks like the following:



In this scenario, SSP has the private host key in the Local Host Key Store to support the inbound SFTP connection with the trading partner, and the public host key in the Known Host Key Store to support the outbound SFTP connection with the target server.

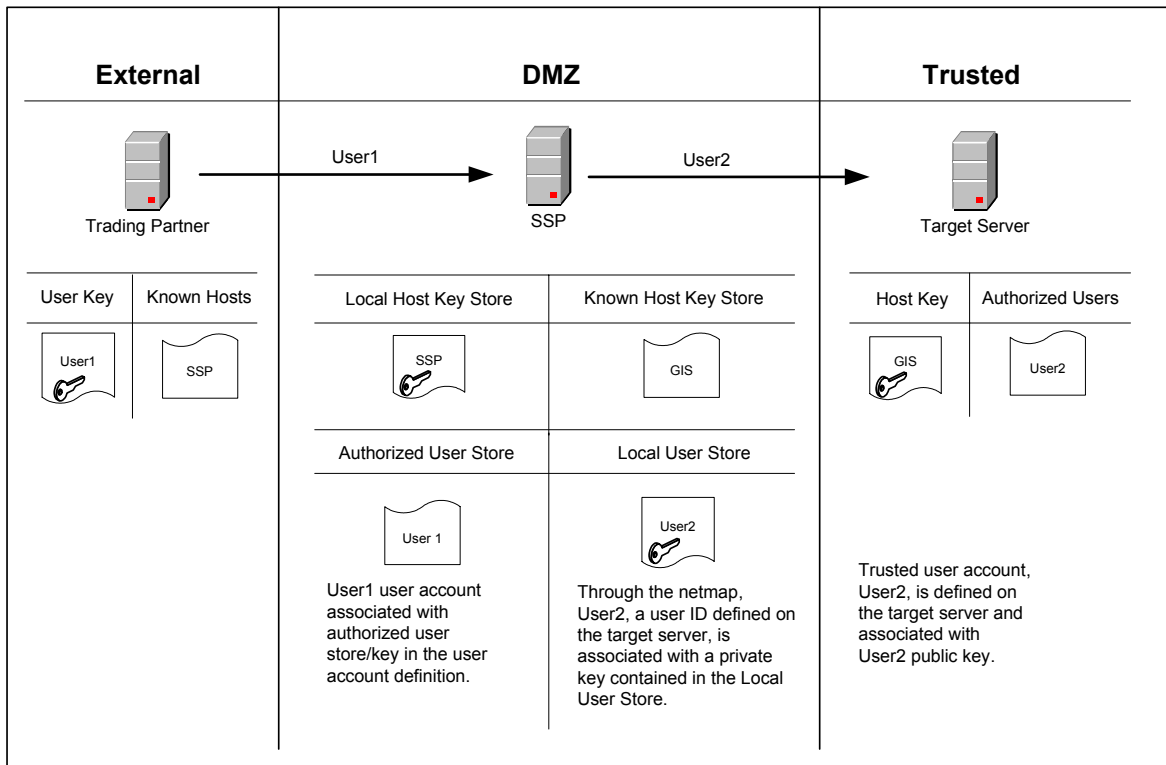
To implement this model:

1. Provide SSP's public local host key to your trading partner.
2. Acquire the target server's public host key.
3. Create a local host key store and import SSP's private key into the local host key store. Refer to *Manage Local Host Key Stores and Keys* on page 292.

4. Create a known host key store and import the target server's public host key into the known host key store. Refer to *Manage Known Host Key Stores and Keys* on page 297.
5. In the SFTP Adapter configuration on the Basic tab, select the local host key store you created and specify the location and name of the local host key you imported into the local host key store.
6. In the outbound node tab of the SFTP server connection definition in the netmap, select the known host key store you created and specify the location and name of the known host key you imported into the known host key store.

Implement Public Key User Authentication for Inbound and Outbound Connections

You can add user authentication to the basic SSH key implementation by using local user keys and authorized user keys for SFTP communications with your trading partner and target server. The key distribution looks like the following:



In this scenario, SSP has the mandatory local host key and known host key, and, for client authentication, it also contains an authorized user key for inbound connections to SSP and a local user key for outbound connections to the target server. In addition, the inbound user ID is replaced with a trusted user account defined on the target server.

To implement this model:

1. Complete the steps in *Use Server Authentication for Inbound and Outbound Connections* on page 290 Use Server Authentication for Inbound and Outbound Connections to configure the mandatory keys required for SSH server authentication.

2. Provide the target server with the public local user key for your internal user ID.
3. Acquire the trading partner's public user key.
4. Create a local user key store and import the target server's private key into the local user key store. Refer to *Manage Local User Key Stores and Keys* on page 299.
5. Create an authorized user key store and import the trading partner's public user key into the authorized user key store. Refer to *Manage Authorized User Key Stores and Keys* on page 294.
6. In the SFTP Policy on the Configuration tab, select:
 - ◆ Key as the Authentication Method on the Advanced tab. For information on how to configure the other authentication methods: Password, Password and Key, Password or Key, refer to SFTP Reverse Proxy Configuration.
 - ◆ Through Local User Store as the User Authentication Mechanism.
 - ◆ Internal User ID - Netmap as the User Mapping method.
7. In the SFTP Netmap outbound node definition Advanced tab specify:
 - ◆ User ID defined on the target server
 - ◆ Local User Key Store and Local User Key for the outbound connection
8. Under credentials in the user account located in the User Stores, on the Advanced tab, select the Authorized User Key Store and select the Authorized User Key you imported into the key store.

Manage Local Host Key Stores and Keys

The local host key store contains the private key used by SSP to identify itself to the client during server authentication in inbound SFTP connections. To use SSH, you must configure a local host key store and import a local host key into the key store. When you are setting up the local host private key, be sure to distribute the matching public key to your trading partners.

Caution: Never distribute the private key to your trading partners.

Create a Local Host Key Store and Import a Key

To create a local host key store and key:

1. Click Credentials from the menu bar.
2. Click Actions > New SSH Key Store > Local Host Key Store.
3. In the Local Host Key Store Configuration window, type a name for the key store in the Local Host Key Store Name field (no spaces allowed).
4. Click New.
5. In the Local Host Key Configuration window, specify the following:
 - ◆ Local Host Key Name
 - ◆ Password

- ◆ Confirm Password
6. Click Browse and select the private key to import into the key store. The key contents display in the Key data field.
 7. Click OK.
 8. Click Save.

You can add as many keys as needed to this key store after it is created. You can now select this key store and key when configuring the SFTP Reverse Proxy Adapter.

Edit a Local Host Key

To edit a local host key:

1. Click Credentials from the menu bar.
1. Expand the SSH Key Stores tree and the Local Host Key Stores tree.
2. Click the key store that contains the key you want to edit.
3. Select the key you want to edit and click Edit.
4. To disable the key, click the Enable Key field.
5. Modify the key definition as necessary.
6. Click OK.
7. Click Save.

Copy a Local Host Key

To copy a local host key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree and the Local Host Key Stores tree.
3. Click the key store that contains the key you want to copy.
4. Select the key you want to copy and click Copy.
5. In the Local Host Key Configuration window, type a name for the new key.
6. Edit the properties as needed.
7. Click OK.
8. Click Save.

Delete a Local Host Key

To delete a local host key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree and the Local Host Key Store tree.
3. Click the key store that contains the key you want to delete.
4. In the Local Host Key Store Configuration window, select the key to delete and click Delete.

5. Click Save.

Copy a Local Host Key Store

After you create a local host key store, you can copy it to create a new local host key store.

To copy a local host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree and the Local Host Key Stores tree.
3. Click the key store to copy and select Actions > Copy Selected.
4. Type a name for the key store (no spaces allowed).
5. Click Save.

Edit a Local Host Key Store

To edit a local host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local Host Key Stores tree.
4. Click the key store you want to edit.
5. In the Local Host Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete a Local Host Key Store

To delete a local host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local Host Key Stores tree.
4. Click the key store to delete and select Actions > Delete Selected.
5. Click Delete.

Manage Authorized User Key Stores and Keys

The authorized user key store contains the public key used by SSP to authenticate the user in SFTP connections. This key is required only if you plan to use public key authentication for inbound SFTP connections. Obtain the public key from your trading partner before you configure the authorized user key store so you will be ready to import the key when you configure the key store.

Create an Authorized User Key Store and Import a Key

To create an authorized user key store and key:

1. Click Credentials from the menu bar.
2. Click Actions > New SSH Key Store > Authorized User Key Store.
3. In the Authorized User Key Store Configuration window, type a name for the key store (no spaces allowed).
4. Click New.
5. In the Authorized User Key Configuration window, type a name for the user key.
6. Click Browse and select the public key to import into the key store. The key contents display in the Key Data field.
7. Click OK.
8. Click Save.

You can now select this key store and key on the User Store Advanced tab of Configuration Manager when you are configuring users in the user store.

Edit an Authorized User Key

To edit an authorized user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store that contains the key you want to edit.
5. In the Authorized User Key Store Configuration window, select the key you want to edit and click Edit.
6. Click Browse to select a different key. The key contents display in the Key Data field.
7. Click OK.
8. Click Save.

Copy an Authorized User Key

To copy an authorized user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Store tree.
4. Click the user key store that contains the key you want to copy.
5. In the Authorized User Key Stores Configuration window, select the key you want to copy and click Copy.
6. In the Authorized User Key Configuration window, type a name for the new key.
7. Edit the properties as needed.

8. Click OK.
9. Click Save.

Delete an Authorized User Key

To delete an authorized user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store that contains the key you want to delete.
5. In the Authorized User Key Store Configuration window, select the key you want to delete and click Delete.
6. Click Save.

Copy an Authorized User Key Store

After you have created an authorized user key store, you can copy it to create new authorized user key stores. To copy an authorized user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store you want to copy and select Actions > Copy Selected.

A copy of the key store is displayed in the Authorized User Key Store Configuration window.

5. Type a name for the key store (no spaces allowed).
6. Click Save.

Edit an Authorized User Key Store

To edit an authorized user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store you want to edit.
5. In the Authorized User Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete an Authorized User Key Store

To delete an authorized user key store:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store you want to delete and select Actions > Delete Selected.
5. Click Delete.

Manage Known Host Key Stores and Keys

The known host key store contains the public key used by SSP to authenticate the server in outbound SFTP connections. To use SSH, you must configure a known host key store and import a known host key into the key store. Obtain the public key from your target server before you configure the known host key store so you will be ready to import the key when you configure the key store.

Create a Known Host Key Store and Import a Key

To create a known host key store and key:

1. Click Credentials from the menu bar.
2. Click Actions > New SSH Key Store > Known Host Key Store.
3. In the Known Host Key Store Configuration window, type a name for the key store (no spaces allowed).
4. Click New.
5. In the Known Host Key Configuration window, type a name for the known host key.
6. Click Browse and select the public key to import into the key store. The key contents display in the Key Data field.
7. Click OK.
8. Click Save.

You can now select this key store and key on the Outbound Node, Basic tab when configuring the SFTP Reverse Proxy Netmap.

Edit a Known Host Key

To edit a known host key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Known Host Key Stores tree.
4. Click the key store that contains the key you want to edit.
5. Select the key you want to edit and click Edit.
6. In the Known Host Key Configuration window, enable or disable the key.
7. Click Browse to select a different key. The key contents display in the Key data field.
8. Click OK.

9. Click Save.

Copy a Known Host Key

To copy a known host key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Known Host Key Stores tree.
4. Click the key store that contains the key you want to copy.
5. In the Known Host Key Store Configuration window, select the key you want to copy and click Copy.
6. In the Known Host Key Configuration window, type a name for the new key.
7. Edit the properties as needed.
8. Click OK.
9. Click Save.

Delete a Known Host Key

To delete a known host key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Known Host Key Stores tree.
4. Click the key store that contains the key you want to delete.
5. In the Known Host Key Store Configuration window, select the key you want to delete and click Delete.
6. Click Save.

Copy a Known Host Key Store

After you have created a known host key store, you can copy it to create a new known host key store.

To copy a known host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Known Host Key Stores tree.
4. Click the key store you want to copy and select Actions > Copy selected.

A copy of the key store displays in the Known Host Key Store Configuration window.

5. Type a name for the key store (no spaces allowed).
6. Click Save.

Edit a Known Host Key Store

To edit an known host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Known Host Key Stores tree.
4. Click the key store you want to edit.
5. In the Known Host Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete a Known Host Key Store

To delete a known host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Known Host Key Stores tree.
4. Click the key store you want to delete and select Actions > Delete Selected.
5. Click Delete.

Manage Local User Key Stores and Keys

The local user key store contains the private key used by SSP to identify itself to the server during public key user authentication in outbound SFTP connections. This key is required *only* if you plan to use public key authentication in outbound SFTP connections. When you are setting up the local user key, be sure to distribute the matching public key to your target server.

Caution: Never distribute the private key to your trading partners.

This section contains the following procedures:

- Create a Local User Key Store and Import a Key
- Edit a Local User Key
- Copy a Local User Key
- Delete a Local User Key
- Copy a Local User Key Store
- Edit a Local User Key Store
- Delete a Local User Key Store

Create a Local User Key Store and Import a Key

To create a local user key store:

1. Click Credentials from the menu bar.
2. Click Actions > New SSH Key Store > Local User Key Store.
3. In the Local User Key Store Configuration window, type a name for the key store (no spaces allowed).
4. Click New.
5. In the Local User Key Configuration window, specify the following:
 - ◆ Local User Key Name
 - ◆ Password
 - ◆ Confirm Password
6. Click Browse and select the private key to import into the key store. The key contents display in the Key data field.
7. Click OK.
8. Click Save.

You can now select this key store and key on the SFTP Netmap, Outbound Node, Advanced tab of Configuration Manager when you are configuring an SFTP Reverse Proxy netmap.

Edit a Local User Key

To edit a local user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Click the key store that contains the key you want to edit.
5. Select the key you want to edit and click Edit.
6. In the Local User Key Configuration window, enable or disable the key.
7. Click Browse to select a different key. The key contents display in the Key data field.
8. Click OK.
9. Click Save.

Copy a Local User Key

To copy a local user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores option.
3. Expand the Local User Key Store option.
4. Click the key store that contains the key you want to copy.
5. In the Local User Key Store Configuration window, select the key you want to copy and click Copy.

6. Type a name for the key store (no spaces allowed).
7. Click OK.
8. Click Save.

Delete a Local User Key

To delete a local user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Select the key store that contains the key you want to delete.
5. In the Local User Key Store Configuration window, select the key you want to delete and click Delete.
6. Click Save.

Copy a Local User Key Store

After you have created a local user key store, you can copy it to create a new local user key store.

To copy a local user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Click the key store you want to copy and select Actions > Copy Selected. A copy of the key store is displayed in the Local User Key Store Configuration window.
5. Type a name for the key store (no spaces allowed).
6. Click Save.

Edit a Local User Key Store

To edit a local user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Click the key store you want to edit.
5. In the Local User Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete a Local User Key Store

To delete a local user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Click the key store you want to delete and select Actions > Delete Selected.
5. Click Delete.

Store System Certificates on a Hardware Security Module (HSM)

A Hardware Security Module (HSM) is a hardware-based security device that generates, stores, and protects cryptographic keys. SSP uses keys and certificates stored in its store or on an HSM. SSP maintains information in its store about all keys and certificates.

To access keys in an HSM device, a reference to the keys and the passphrase protecting the key must be added to SSP. This reference is secure and cannot be used by an intruder to access the certificate information. You can configure keys on the HSM at CM, using command line scripts described in this chapter.

For more security, create the keys on the HSM device and store the HSM private keys on the device. To import externally-created keys into the HSM, first import the external keys into the HSM and then destroy the files containing the external private key.

HSMs implement the Java JCE API. This interface accesses the keys in the device. The JCE implementations for Safenet and Thales have the following differences:

Safenet uses slots, logical entities defined through the Safenet administration utility. Designate a slot for SSP and assign a user PIN. Configure SSP and identify the slot to use. Only one slot can be used by SSP.

Safenet uses a single keystore for all keys in a slot. The user PIN protects all the keys in the slot. Each key within a slot must have a unique alias.

Thales uses a security world that contains one or more HSM modules. The modules can reside on the same or different machines. The keys in the security world are protected by an operator smart card. Create an operator smart card set for SSP, identify “1 of N” for the cards, and assign a passphrase to each card. Before SSP can start, insert the operator smart card protecting the SSP keys into the card reader.

Thales supports multiple keystores. Each keystore can contain multiple keys, but SSP only stores one key per keystore. With Thales, multiple keys can have the same alias. For example, on SI, all keys on an Thales HSM have the alias Key. Each keystore has a unique instance ID defined as a 40-character hexadecimal string. The combination of the instance ID and the key alias makes each key unique.

Enable and Disable the HSM Environment

Use the `setupHSM` command to enable or disable the HSM environment. Run this command on the engine. If you are using a `netHSM` module and CM has access to the `netHSM`, you can also run the command on CM. Running the command on CM allows you to configure the HSM keys without requiring a running engine. However, you must stop CM.

Stop the engine or CM before you run this command. Additionally, you must have permission to write files to the SSP installation directory. If you reinstall the HSM support software, run the `setupHSM -enable` command again to make sure that any updated jar files and libraries are copied to the installation directory.

Enable the HSM Environment

Use the `setupHSM -enable` command to copy files from the HSM hardware to SSP, copy the HSM security providers in the right order, update the `security.properties` file with the appropriate Certicom TLS security string for the HSM you are using, and add any environment variables to the startup scripts.

To setup the HSM environment for Windows, type the following command:

```
setupHSM -enable [parameters]
```

To setup the HSM environment for UNIX or Linux, type the following command:

```
setupHSM.sh -enable [parameters]
```

Following is a description of the enable parameters:

Parameter	Description
hsm	HSM type. Required if you are enabling the HSM. Valid values = nCipher Eracom.
slot	Slot number assigned to SSP. The optional parameter is valid for Safenet only. Default=0.
path	Path to the root directory of the HSM runtime support software. Required. If the path contains embedded spaces, enclose the whole parameter in double-quotes. For example, "path=C:\Program Files\Safenet". On UNIX, the value is normally /opt/nfast for Thales and /opt/Safenet for Safenet. On Windows, the value is normally C:\nfast for Thales and C:\Program Files\Safenet for Safenet.
netserver	Host name or IP address of the netHSM server. Optional. Valid for Safenet on UNIX. It is ignored on Windows.
systempass	Engine system passphrase or CM passphrase, depending upon where the command is run. Optional. If you do not configure this parameter, the user is prompted for the passphrase.

Following is a sample script to setup an Safenet HSM on UNIX:

```
setupHSM.sh -enable hsm=eracom slot=1 path=/opt/Eracom
```

Following is a sample script to setup an Thales HSM on UNIX:

```
setupHSM.sh -enable hsm=nCipher path=/opt/nfast
```

Disable the HSM Environment

Use `setupHSM -disable` to delete the HSM provider files, remove the HSM security providers, restore the default Certicom TLS security provider definitions, and remove the HSM environment variables from the startup scripts.

To disable the HSM environment, type the following command:

```
setupHSM -disable systempass
```

Following is a description of the HSM setup disable parameter:

Parameter	Description
systempass	Engine system passphrase or CM passphrase, depending upon where the command is run. Optional. If you do not define this parameter, you are prompted for the passphrase.

Manage Key Certificates

Use the `manageKeyCerts` command to manage key certificates in the SSP system certificate store and on the HSM. Use this command to perform the following tasks:

- Create Self-Signed Certificates
- Import a Certificate
- Export a Certificate
- Obtain a Certificate from the HSM Device
- Store a Certificate on the HSM Device
- Copy a Certificate
- Move a Certificate from One SSP System Certificate Store To Another Store
- Rename a Certificate on the SSP System Certificate Store
- Delete a Certificate
- List Key Certificates on the SSP System Certificate Store
- List Key Certificates on the HSM Device
- Load References to Keys on the HSM into the SSP System Certificate Store
- Update the HSM Password for HSM Key Certificates Stored in the SSP System Store

Create Self-Signed Certificates

Use the `manageKeyCerts -create` command to create a self-signed key certificate. Stop CM before you run this command.

Consider the following before you use this command:

If the engine parameter is defined, a certificate is created on the HSM configured for that engine. If a `netHSM` is used and multiple engines access the `netHSM`, any of the engines can be specified to create the certificate on the HSM.

If the engine uses a PCI module and it cannot be accessed by other engines, group the key certificates for that engine in a separate system certificate store. Those key certificates cannot be shared with other engines.

If the engine parameter is not defined, and HSM support is enabled on CM, the key certificate is created on the HSM configured for CM. Make sure that engines that use this key certificate can access the HSM enabled for CM.

If the engine parameter is not defined and HSM support is not enabled on CM, the key certificate is created on the SSP system certificate keystore.

To create a self-signed key certificate, type the following command:

```
manageKeyCerts -create [parameters]
```

Following are the parameters used to create a key certificate:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
alias	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to certificate name.
keySize	Key size of the file to create. Valid values = 1024 2048 4096. Default=1024.
CN	Certificate common name. Required. If the name contains spaces, enclose the command and string in double quotes, for example "CN=my name".
email	E-mail address. Optional.
O	Organization. Optional. If the value contains spaces, enclose the command in double quotes, for example, "O=my org".
OU	Organization unit. Optional. Repeat this parameter to specify more than one organization unit. If the value contains spaces, enclose the command in double quotes, for example, "OU=my unit".
L	Location (city). Optional. If the value contains spaces, enclose the command in double quotes, for example, "L=my location".
ST	State. Optional. If the value contains spaces, enclose the command in double quotes, for example, "ST=my state".
C	Two letter country code. Optional.
daysValid	How many days the key certificate is valid. Optional. Default=365.
serial	Serial number for the key certificate. Optional. Default=1.

Parameter	Description
certSignBit	Whether to set the certificate signing bit on in the key usage flags. Valid values = n y false true. Default=n.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = false true. Default=false.
systempass	Passphrase for CM.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Safenet, the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key in the keystore. Optional. Prompts if not defined. For Safenet, this parameter can be anything and will be ignored. For Thales, this must be the same value as the keystore password.

Import a Certificate

Use the `manageKeyCerts -import` command to import a certificate into the SSP system certificate store and the HSM. Stop CM before you run this command.

Consider the following before you use it:

If you define the `engine` parameter, the certificate is imported to the HSM configured for that engine. If a `netHSM` is used and multiple engines access the `netHSM`, any of the engines can be specified to handle the request. Configure HSM support on the engine.

If the engine uses a PCI module and that module cannot be accessed by other engines, group the key certificates for that engine in a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.

If you do not define the `engine` parameter, and HSM support is enabled on CM, the key certificate is imported on the HSM configured for CM. Make sure that engines that use this key certificate can access the HSM enabled for CM.

If you do not define the `engine` parameter and HSM support is not enabled on CM, the key certificate is imported to the SSP system certificate store only.

To import a key certificate into the SSP system certificate store, type the following command:

```
manageKeyCerts -import [parameters]
```

Following is a description of the import parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfiltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
alias	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to certificate name.
file	Fully-qualified path of the key certificate file to import. Required. The file must be PEM or PKCS12. The script looks for BEGIN/END PEM markers in the file. If they are not found, the file is assumed to be PKCS12 format.
replace	Whether to replace a system certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = n y false true. Default=n.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
pemkeypass	Password for the PEM private key, if the file is PEM. Optional. Prompts if not defined.
pkcs12storepass	Password of the PKCS12 file, if the file is not PEM. Optional. Prompts if not defined.
pkcs12keypass	Passphrase for the key in the PKCS12 file, if the import file is not PEM. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Safenet, the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined. For Safenet, this parameter is not used. For Thales, define this parameter using the same value as the keystore password.

Export a Certificate

Use the `manageKeyCerts -export` command to export a certificate from the system store or the HSM. CM can be running when you run this command.

Consider the following before you use this command:

If you specify the engine parameter and the certificate is stored on the HSM, the certificate is exported from the HSM configured at the engine. You must enable the HSM on the engine. If a netHSM is used and multiple engines can access it, any of the engines can be specified to export the certificate.

If you do not specify the engine parameter and the key certificate is stored in an HSM, the certificate is exported from the HSM configured for CM. You must enable the HSM on CM to export a certificate from it.

If the certificate is not stored on an HSM, the engine parameter is ignored and the certificate is exported from the SSP system certificate store.

For key certificates stored on the HSM, only the public certificate in PEM format will be exported. The private key cannot be exported.

To export a key certificate from the SSP system certificate store, type the following command:

```
manageKeyCerts -export [parameters]
```

Following is a description of the export parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
format	Format for the key certificate file. This parameter is required for non-HSM key certificates. Forced to pem for the HSM key certificates. Valid values = pem pkcs12.
file	Fully-qualified path of the file where the key certificate file will be stored. Required.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
pkcs12storepass	Password of the PKCS12 file, if the format is PKCS12 and the key certificate is not stored on an HSM. Optional. Prompts if not defined.
pkcs12keypass	Passphrase for the key in the PKCS12 file, if the format is PKCS12 and the key certificate is not stored on an HSM. Optional. Prompts if not defined.
pemkeypass	Passphrase to encrypt the private key if the format is PEM.

Obtain a Certificate from the HSM Device

Use the manageKeyCerts -getFromHSM command to extract a reference to a key in the HSM and add the entry into the SSP keystore. Stop CM before you run this command.

Consider the following before you use this command:

If you define the engine parameter, certificate information is obtained from the HSM at the engine. You must enable the HSM on the engine.

If you configure netHSM, and multiple engines access the netHSM, any of the engines can be specified in the command.

If you do not specify the engine parameter, the key certificate is obtained from the HSM configured at CM. You must enable the HSM on CM to obtain information from the HSM at CM.

For the Thales HSM, the keystore blob for the key (Key Instance, as displayed by KeySafe) must be provided in the keyStoreData parameter. Obtain this 40-character hexadecimal string by running the -listHSM command.

After a reference to an HSM key certificate is successfully obtained, the HSM key cannot be obtained again under a different SSP system certificate name. This action results in an error.

To obtain a key certificate from the HSM, type the following command:

```
manageKeyCerts -getFromHsm [parameters]
```

Following is a description of the getFromHSM parameters:

Parameter	Description
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfitKeyStore.
engine	Name of the engine with access to the HSM. Optional.
certName	Name of the key certificate on SSP. Required.
alias	Alias for the key certificate on the HSM. Required.
keyStoreData	HSM keystore blob string. Required with the Thales HSM. This is a 40-character hex string, displayed as Key Instance by the Thales KeySafe utility. If not provided and the HSM key certificate already exists in the system certificate store, the current keystore blob is used to pull the key back into the CM store. Use the -listHsm command to get the blobs for key certificates in the HSM. Alternatively, the blob string can be written to a file. Specify that file name in the keyStoreFile parameter.
keyStoreFile	File containing the HSM keystore blob string. If defined, this parameter overrides the keyStoreData parameter.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = n y false true. Default=n.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Safenet, use the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined. For Safenet, this parameter is not used. For Thales, define this parameter using the same value as the keystore password.

Store a Certificate on the HSM Device

If you have an existing certificate in the SSP certificate store, use the `manageKeyCerts -storeOnHsm` command to store the key certificate in the HSM. Stop CM before you use this command.

Consider the following before you use this command:

If you define the engine parameter, the certificate is stored at the HSM for the engine. You must enable HSM on the engine.

If you configure a netHSM and multiple engines access the netHSM, any of the engines can be specified to run the request.

If the engine uses a PCI module and the module cannot be accessed by other engines, group the key certificates for the engine into a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.

If you do not specify the engine parameter, and HSM support is enabled on CM, the key certificate is stored on the HSM configured at CM.

If the key certificate is already stored in an HSM, the command fails.

After a key certificate is stored in an HSM, the key certificate record at CM is updated with a reference to the key in the HSM. If it has a PEM private key, the private key is deleted from the certificate store.

To store a key certificate on the HSM, type the following command:

```
manageKeyCerts -storeOnHsm [parameters]
```

Refer to the following table for a description of the `storeOnHsm` parameters:

Parameter	Description
<code>certName</code>	Name of the key certificate on SSP. Required.
<code>certStore</code>	Name of the system certificate store where the key certificate is stored. This field is optional. Default= <code>dfltKeyStore</code> .
<code>engine</code>	Name of the engine with access to the HSM. Optional.
<code>alias</code>	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to certificate name.
<code>systempass</code>	CM system passphrase.
<code>adminid</code>	Administrator ID. Optional. Prompts if not defined.
<code>adminpass</code>	Administrator password. Optional. Prompts if not defined.
<code>keystorepass</code>	Keystore password. Optional. Prompts if not defined. For Safenet, the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.

Parameter	Description
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined. For Safenet, this parameter is not used. For Thales, define this parameter using the same value as the keystore password.

Copy a Certificate

Use the `manageKeyCerts -copy` command to copy an existing key certificate and assign it a new name. Stop CM before you run this command.

Consider the following before you use this command:

If you specify the engine parameter and the key certificate is stored in an HSM, the engine makes a copy of the key certificate on the HSM, using the new alias provided. HSM support must be enabled at the engine to run this command.

If you configure netHSM and multiple engines access it, specify any of the engines to run the request.

If you do not specify the engine parameter and the key certificate is stored in an HSM, the command makes a copy of the certificate on the HSM configured at CM, using the new alias provided. You must configure the HSM at CM to use this command.

If the key certificate is not stored in an HSM, the engine and new alias parameters are ignored.

To copy a key certificate, type the following command:

```
manageKeyCerts -copy [parameters]
```

Following is a description of the copy parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate is stored. This field is optional. Default=dfltKeyStore.
newName	Name for the copy of the key certificate. Required. Default=certName.
newAlias	Alias for the copy of the key certificate on the HSM. This parameter is required if the key certificate is stored on the HSM.
replace	Whether to replace a key certificate if a certificate with the new name already exists in the SSP system certificate store. Optional. Valid values = n y false true. Default=n.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Move a Certificate from One SSP System Certificate Store To Another Store

Use the `manageKeyCerts -move` command to move the key certificate from one SSP certificate store to another store.

Stop CM before you run this command. After you run it, all references to the original system certificate are updated in the netmap definitions with references to the new certificate store and certificate name.

To move the key certificate from one SSP certificate store to another, type the following command:

```
manageKeyCerts -move [parameters]
```

Following is a description of the move parameters:

Parameter	Description
<code>certName</code>	Name of the key certificate on SSP. Required.
<code>certStore</code>	Name of the system certificate store on SSP. This field is optional. Default= <code>dfltKeyStore</code> .
<code>destCertStore</code>	Name of the system certificate store on SSP where the key certificate will be moved. Required. If the store does not exist, it is created.
<code>newName</code>	Name for the key certificate on the destination system certificate store. Optional. Default= <code>certName</code> .
<code>replace</code>	Whether to replace a key certificate if a certificate with the same name already exists in the destination SSP system certificate store. Optional. Valid values = <code>n</code> <code>y</code> <code>false</code> <code>true</code> . Default= <code>n</code> .
<code>systempass</code>	CM system passphrase.
<code>adminid</code>	Administrator ID. Optional. Prompts if not defined.
<code>adminpass</code>	Administrator password. Optional. Prompts if not defined.

Rename a Certificate on the SSP System Certificate Store

Use the `manageKeyCerts -rename` command to rename a key certificate in the SSP certificate store. Stop CM before you run this command.

After you run it, all references to the original system certificate are updated in the netmap definitions with references to the new certificate store and certificate name.

To rename the key certificate on the SSP certificate store, type the following command:

```
manageKeyCerts -rename [parameters]
```

Refer to following table for a description of the rename parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfitKeyStore.
newName	New name for the key certificate. Required.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Delete a Certificate

Use the `manageKeyCerts -delete` command to delete a key certificate from the SSP keystore or from the HSM. Stop CM before you use this command.

Consider the following before you use this command:

If the system certificate is in use, the command fails. The list of netmap nodes using the system certificate is displayed.

If the key certificate is stored in an HSM, specify the `deleteFromHsm` parameter to delete the key certificate from the HSM as well.

If the engine parameter is defined, the key certificate is stored in an HSM, and `deleteFromHSM` is set to yes, the key certificate is deleted from the HSM at the engine. You must configure HSM support at the engine to use this command.

If a netHSM is configured and multiple engines access the netHSM, any of the engines can be specified to run the command.

If the engine parameter is not specified, the key certificate is stored in an HSM, and the `deleteFromHSM` is set to yes, the command deletes the key certificate from the HSM at CM. HSM support must be enabled at CM to use this command.

If the key certificate is not stored in an HSM, the `deleteFromHSM` and engine parameters are ignored.

To delete the key certificate from the SSP certificate store, type the following command:

```
manageKeyCerts -delete [parameters]
```

Following is a description of the delete parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfitKeyStore.

Parameter	Description
deleteFromHsm	Determines whether to delete the key certificate from the HSM. This parameter is required if the key certificate is stored on the HSM. Valid values = y n true false.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

List Key Certificates on the SSP System Certificate Store

Use the `manageKeyCerts -list` command to list key certificates on the SSP certificate store. The command can run while CM is running.

To list the key certificate on the SSP certificate store, type the following command:

```
manageKeyCerts -list [parameters]
```

Following is a description of the list parameters:

Parameter	Description
certStore	Name of the system certificate store on SSP. This field is optional. Default= <code>dfiltKeyStore</code> . To list certificates in all system certificates stores, define <code>certStore=*</code> .
systempass	CM system passphrase. Optional. Prompts if not specified.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

List Key Certificates on the HSM Device

Use the `manageKeyCerts -listHsm` command to list keys on the HSM. This command can be run while CM is running.

Consider the following before you use this command:

For Thales HSMs, all HSM keys that can be loaded with the provided smart card passphrase are listed, if the `keyStoreData` parameter is not defined.

If you define the `engine` parameter, the keys stored on the HSM at the engine are listed. You must configure HSM support at the engine to use this command.

If a `netHSM` is used and multiple engines access it, any of the engines can be specified to run the request.

If the `engine` parameter is not defined, the command lists the keys stored on the HSM at CM. HSM support must be enabled at CM.

To list the key certificate on the HSM device, type the following command:

```
manageKeyCerts -listHsm [parameters]
```

Refer to the following table for a description of the list parameters:

Parameter	Description
engine	Name of the engine with access to the HSM. Optional.
keyStoreData	HSM keystore blob string. Used with the Thales HSM. This is a 40-character hex string, displayed as "Key Instance" by the Thales KeySafe utility. If it is not provided, all keys that can be loaded with the provided smart card passphrase are listed. Alternatively, the blob string can be written to a file. Specify that file name in the keyStoreFile parameter.
keyStoreFile	File containing HSM keystore data. If defined, this parameter overrides the keyStoreData parameter.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Safenet, the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.

Load References to Keys on the HSM into the SSP System Certificate Store

Use the manageKeyCerts -loadHsm command to load references to keys on the HSM device into the SSP system certificate store. Stop CM before you run this command.

Consider the following before you use this command:

This command is the same as the -getFromHSM command invoked for a list of HSM keys in a properties file. It facilitates HSM key migration from SSP 2.0.02 to SSP 3.1.0, and provides an easy way to populate SSP with HSM keys.

If you define the engine parameter, the keys stored on the HSM at the engine are listed. Enable HSM support at the engine to use this command. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.

If you do not specify the engine parameter, keys stored on the HSM at CM are listed. HSM support must be enabled at CM to use this command.

After a reference to an HSM key certificate is imported into SSP, that HSM key cannot be referenced again under a different SSP system certificate name.

To override the certificate store for a certificate, use the certStore=<store name> in the input properties file.

To load references to keys on the HSM device into the system certificate store, type the following:

```
manageKeyCerts -loadHsm [parameters]
```

Refer to the following table for a description of the loadHsm parameters:

Parameter	Description
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfitKeyStore.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Safenet, the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
autoGenName	Whether to auto-generate the name for the key certificate on SSP. Optional. If enabled, and the properties for the key certificate do not specify the certName property, a name is generated using the prefix "hsm_" followed by a hash of the key certificate properties (alias, keystore, type, provider, issuer, subject, serial). Default=n.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = false true. Default=false.

Parameter	Description
file	<p>Path to the file containing information about the HSM key certificates to load. Required.</p> <p>It refers to the output of the SSP 2.0.02 RemoveSystemCert -l script, the -listHSM command, or a text file with lines in the format key=value. To load multiple key certificates, separate the properties for each with a blank line or a line starting with "[".</p> <p>All key certificates on the HSM device are listed. The command searches the property file, to find a key certificate on the HSM that matches the specified property. If a match is found, an entry for the matched HSM key certificate is added to the SSP system certificate store with the name specified in the properties, or with an auto-generated name if the parameter called autoGenName=y.</p> <p>If the file is the output of the SSP 2.0.02 RemoveSystemCert -l script, the lines on the file are mapped to properties as follows:</p> <ul style="list-style-type: none"> ◆ PrivateKeyInfo for ID—alias (for Safenet) ◆ Name—certName ◆ KeyStoreType—type ◆ Issuer—issuer ◆ Subject—subject ◆ Serial—serial <p>If the file is the output of the version 2.0.02 RemoveSystemCert -l script, remove all lines up to, but not including, the first "PrivateKeyInfo for ID" at the top of the file.</p> <p>If the file is the output of the manageKeyCerts -listHSM script, remove all lines up to, but not including, the first "[1]=====", from the top of the file.</p>

Update the HSM Password for HSM Key Certificates Stored in the SSP System Store

Use the manageKeyCerts -updateHsmPass command after you change the password for the HSM, using the HSM administration utilities. Stop CM before you run this command.

Consider the following before you use this command:

This command does not change the HSM keystore password. It is changed through the HSM administration utilities. You must stop and restart the engine after you change a key store password through the HSM administration utilities.

If you define the engine parameter, this command first tries to load the HSM keys with their current passwords. If a key cannot be loaded, it tries to load the HSM keys with the new password. If the key is successfully loaded, the password for the key is updated on SSP. HSM support must be enabled at the engine to use this command. If netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.

To update the keystore password on all system certificates, define the certStore=* parameter.

To update the password of the HSM on the SSP system certificate store, type the following command:

```
manageKeyCerts -updateHsmPass [parameters]
```


Following is a description of the updateHsmPass parameters:

Parameter	Description
certStore	Name of the system certificate store where the key certificates are stored. This field is optional. Default=dfлтKeyStore.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
newKeyStorePass	New HSM keystore password. Optional. For Safenet, the new user PIN for the slot used by SSP. For Thales, the new passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.

ManageCSRs

Use the manageCSRs command on CM to manage Certificate Signing Requests (CSR). CSRs created with this command cannot be viewed through the CM GUI.

First, create a CSR. The script generates a temporary self-signed key certificate in the HSM or an SSP system certificate store, if the HSM is not enabled. Then send the CSR to a Certification Authority (CA).

When the CA returns the CA-signed certificate, run the manage CSRs command again to replace the self-signed key certificate with the CA-signed certificate. The updated CA-signed certificate is added to the SSP system certificate store, and the CSR status is set to complete.

The key certificate can now be used by SSP.

Use the manageCSRs command to perform the following tasks:

- Create a CSR
- Update a CSR
- Delete a CSR
- List CSRs on the CM Store
- Retrieve a CSR to Send to a Certification Authority
- Retrieve the CA-signed Certificate

Create a CSR

Use the manageCSRs -create command to create a CSR for a key certificate at either the HSM or the SSP system certificate store. You can use this command while CM is running.

Consider the following before you use this command:

- If you define the engine parameter, a key certificate is created on the HSM configured for the engine. You must enable HSM support at the engine in order to run this command. If a

netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.

If you do not define the engine parameter and HSM support is not enabled on CM, the system certificate store certificate is created on the SSP system certificate store.

To create a CSR on Windows:

```
manageCSRs -create [parameters]
```

To create a CSR on UNIX or Linux:

```
manageCSRs.sh -create [parameters]
```

Following is a description of the create CSR parameters:

Parameter	Description
csrName	Name for the CSR. Required.
engine	Name of the engine with access to the HSM. Optional.
alias	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to CSR name.
keySize	Key size of the file to create. Valid values = 1024 2048 4096 Default=1024.
CN	Certificate common name. Required. If the name contains spaces, enclose the command and string in double quotes, for example, "CN=my name".
O	Organization. Optional. If the value contains spaces, enclose the command and string in double quotes, for example, "O=my org".
OU	Organization unit. Optional. Repeat this parameter to specify more than one organization unit. If the value contains spaces, enclose the command and string in double quotes, for example, "OU=my unit".
L	Location (city). Optional. If the value contains spaces, enclose the command in double quotes, for example, "L=my location".
ST	State. Optional. If the value contains spaces, enclose the command and string in double quotes, for example, "ST=my state".
C	Two letter country code. Optional.
email	E-mail address. Optional.
file	Fully-qualified path to the file where the CSR will be stored. If this parameter is not defined, the output of the CSR is displayed on the monitor. To obtain the CSR information later, use the -getpkcs10 command. Optional.

Parameter	Description
systempass	CM system passphrase. Optional.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Safenet, the user PIN for the slot used by SSP. For Thales, the passphrase for the operator smart card, used to protect the key. Be sure that the card is in the card reader before you run this command.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined. This value is not used by the Safenet HSM.

Update a CSR

Use the `manageCSRs -update` command to update a pending CSR with the CA-signed certificate. Stop CM before you run this command.

Consider the following when using this command:

If the key certificate is created in an HSM and you specify the engine parameter, the command notifies the engine to update the key certificate on the HSM. Configure HSM support at the engine to use this command.

If a netHSM is used and multiple engines access it, any of the engines can be specified to perform the update.

If the engine uses a PCI module and that module cannot be accessed by other engines, you must group the key certificates for the engine in a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.

If the key certificate was created in an HSM and you do not specify the engine parameter, the command updates the key certificate on the HSM at CM. You must enable HSM support at CM.

If the key certificate was not created in an HSM, it is updated on the SSP system certificate store. The engine parameter is ignored.

To update a pending CSR, type the following command:

```
manageCSRs -update [parameters]
```

Following is a description of the update parameters:

Parameter	Description
csrName	Name for the CSR. Required.
engine	Name of the engine with access to the HSM. Optional.
file	Fully-qualified path of the CA-signed certificate file. Required.

Parameter	Description
certName	Name of the key certificate on SSP. Required.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
newKeyStorePass	New HSM keystore password. Optional. If defined, this value overrides the keystore password used when the CSR was created. This parameter allows you to update a CSR on the HSM after the keystore password for the HSM is changed.

Delete a CSR

Use the `manageCSRs -delete` command to delete a CSR from the CM store. This command can be run while CM is running.

Consider the following when using this command:

If the CSR is pending and its key certificate was generated on an HSM, the temporary key certificate is deleted from the HSM.

If the CSR is complete, this command deletes the CSR, but does not delete the key certificate. To delete the key certificate, use the `manageKeyCerts -delete` command.

To delete a CSR from CM, type the following command:

```
manageCSRs -delete [parameters]
```

Following is a description of the delete parameters:

Parameter	Description
csrName	Name for the CSR. Required.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
newKeyStorePass	New HSM keystore password. Optional. If defined, this value overrides the keystore password used when the CSR was created. This parameter allows you to update a CSR on the HSM after the keystore password for the HSM is changed.

List CSRs on the CM Store

Use the `manageCSRs -list` command to display a list of CSRs on CM. This command can be run while CM is running.

To list the CSRs in the CM store, type the following command:

```
manageCSRs -list [parameters]
```

Refer to the following table for a description of the list parameters:

Parameter	Description
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Retrieve a CSR to Send to a Certification Authority

Use the `manageCSRs -getpkcs10` command to retrieve a CSR to send to a Certificate Authority (CA). This command can be run while CM is running.

To retrieve a CSR from the HSM that is ready to send to a CA, type the following command:

```
manageCSRs -getpkcs10 [parameters]
```

Refer to the following table for a description of the list parameters:

Parameter	Description
csrName	Name for the CSR. Required.
file	Fully-qualified path of the file where the CSR will be stored.
systempass	CM system passphrase. Optional.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Retrieve the CA-signed Certificate

Use the `manageCSRs -getcacert` command to retrieve the CA-signed certificate received from a CA, after the update command has been run. The certificate is returned in PEM format. This command can be run while CM is running.

To retrieve the CA-signed certificate from the HSM, type the following command:

```
manageCSRs -getcacert [parameter]
```

Refer to the following table for a description of the getcacert parameters:

Parameter	Description
csrName	Name for the CSR. Required.
file	Fully-qualified path where the CA-signed certificate will be stored. If not specified, the certificate text is written to the display.
systempass	CM system passphrase. Optional. Prompts if not defined.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Start and Stop Configuration Manager and the Engine

Use the procedures in this section to run the SSP components and stop them.

Start the Engine on UNIX or Linux

When you install the SSP engine, you define a passphrase. It is required at startup. Use one of the following methods to start the SSP engine:

Start the engine automatically, without interaction from the user.

Require the user to type a passphrase at startup. It is masked and not visible as it is typed.

The server starts in the background. All log messages are written to the bin/startEngine.out file.

Start the Engine Using a Stored Passphrase

To start the engine on UNIX or Linux without being prompted for a passphrase.

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

```
./startEngine.sh
```

2. At the prompt, type the passphrase defined during installation and press **Enter**.

A message is displayed indicating the engine is ready for service.

Note: If the engine is running in the background, the message is not displayed. To view the message, go to the startEngine.out file.

Start the Engine And Require a Passphrase

To start the engine and require that a passphrase be typed at startup:

1. Delete the sb.enc file from *install_dir/conf/system*.
2. Navigate to *install_dir/bin* and type the following command:

```
./startEngine.sh
```

3. When prompted for a passphrase, type the passphrase defined at installation.

Stop the Engine from UNIX or Linux

To stop the engine from the command line:

1. Navigate to *install_dir/bin*, and type the following command:

```
./stopEngine.sh
```

2. At the passphrase prompt, type the passphrase defined for the Engine.

A message is displayed indicating the engine is stopped.

Start the Engine on Windows

When you install the engine on Windows, it is installed as a Windows service and configured to start manually. By default, start SSP by starting the Sterling Secure Proxy Engine service from the Services application in Windows. To start SSP automatically when you run Windows, go to Windows services and change the Sterling Secure Proxy Engine V3.3.01 application startup.

Start the Engine as a Console Application on Windows

To start the engine:

1. Click **Start>Run>Browse**.
2. Double-click the following file in the *install_dir*\bin directory:

```
startEngine.bat
```

3. Click **OK**.
4. If prompted, type the passphrase defined for the engine.
A message is displayed indicating the engine is ready for service.

Note: When you run the engine as a Windows service, the passphrase is encrypted and stored.

Start SSP as an Automatic Windows Service

Running SSP as a Windows service is a convenient method of starting SSP. When you set it up, SSP starts automatically when you start Windows. CM and the engine are defined as Windows services at installation but are not set as automatic services. You need to configure them if you want to enable this startup option. After you set up an automatic Windows service, SSP runs continuously in the background until you shut it down, or shut down Windows.

Refer to Microsoft Windows documentation to configure SSP as an automatic Windows service.

Set Up the Engine to Require a Passphrase Prompt at Startup on Windows

When you install the engine, the passphrase is saved in an encrypted file and the program starts as a Windows service without prompting you to type the passphrase. You can change the startup method to run the program in the foreground and require a passphrase at startup.

To change the startup method to require a passphrase at startup, delete the **sb.enc** file from the *install_dir*\conf\system directory, where *install_dir* is the directory where the engine is installed.

Set up the Engine to Start as a Windows Service

To set up the engine to start as a Windows service:

1. Click **Start>Run>Browse**.
2. Double-click the following file in the *SSP_engine_install_dir*\bin directory:


```
enableBootstrap.bat
```

3. Click **OK**.
4. At the prompt, type the engine passphrase and press **Enter**.

Note: When you run the engine as Windows service, the passphrase is encrypted and stored.

Stop the Engine from a Windows Console Application

To stop the engine when it is running as a Windows console application:


1. Click **Start>Run>Browse**.
2. Double-click the following file from the *install_dir\bin* directory:

```
stopEngine.bat
```

3. Click **OK**.
4. At the prompt, type the engine passphrase and press **Enter**.

Stop the Engine from CM

To stop the engine from CM:

1. Click **Monitoring** from the menu bar.
2. Click **Engine Status (All)**. A list of all configured engines is displayed. Engines that are running are indicated with the .
3. Select the engine that you want to stop.
4. Click **Stop Engine**.
5. Type the engine passphrase and click **OK**.

Stop the Engine from Windows Services

To stop the engine from Windows services, go to Windows services and stop the Sterling Secure Proxy Engine V3.3.01 application.

Run CM on UNIX or Linux

To run CM, first start it and then log on. Use one of the following methods to start CM:

Start SSP automatically, using a stored passphrase read from an encrypted file.

Start SSP and require the user to type a passphrase. The passphrase is masked and not visible.

CM starts in the background and writes log messages to the bin/startCM.out file.

Start CM Without Providing a Passphrase at Startup on UNIX or Linux

Use this method to start CM using a stored passphrase. The file called `sb.enc` is created during installation and must exist in the `ssp_install_dir/conf/system` directory.

To start CM automatically without the need to provide a passphrase:

Navigate to the `install_dir/bin` directory, and type the following command:

```
./startCM.sh
```

Start CM and Require a Passphrase at Startup on UNIX or Linux

To start CM and require that a passphrase be provided:

1. Navigate to `install_dir/conf/system` and delete the `sb.enc` file
2. Type the following command:

```
./startCM.sh
```

3. Type the CM passphrase and press **Enter**.

Log On to CM on UNIX or Linux

You log on and access the CM through a web browser.

To sign in to CM from Windows:

1. Open Internet Explorer.
2. Type the sign in information in the following format. Refer to the table for a description:

```
https://hostname or ipaddress:port/SSPDashboard
```

Component	Description
hostname or ipaddress	Name or IP address of the computer where CM is installed.
port	Port defined for the web server at installation. Default= 8443.

3. On the sign in screen, type the user ID and passphrase and click **Sign In**.

Stop CM on UNIX or Linux

If you close the web browser, CM continues to run.

To stop CM on UNIX or Linux:

1. Log out of CM.
2. Navigate to the `install_dir/bin` directory and type the following command:

```
./stopCM.sh
```

3. Type the passphrase for CM.
4. Type the administrator user name and passphrase.

Run CM on Windows

To run CM, first start the application and then log on.

Start CM from Windows

To start CM:

1. Click **Start>Run>Browse**.
2. Browse to *install_dir*\bin, where *install_dir* is the CM installation directory
3. Double-click the following file:

startCM.bat

4. Click **OK**.
5. If prompted, type the passphrase defined for CM.

A message is displayed that CM is ready for service and identifying the URL used to connect to the CM server.

6. Record the URL to connect to the CM server on the Startup Worksheet.

Log on to CM from Windows

After starting CM, log on to the SSP dashboard and access CM through a web browser.

To log on to CM:

1. Open Internet Explorer.
2. Type the logon in the following format. Refer to the table for a description of the components:

https:// : /SSPDashboard

Type the following information for your configuration:

Component	Description
hostname or ipaddress	Name or IP address of the CM host system.
port	The port defined for CM at installation. The default value is 8443.

3. On the logon screen, type the user ID and passphrase.
4. Click **Logon**.

Stop the Engine from Windows

If you close the web browser, the engine continues to run.

To stop the engine on Windows, go to Windows services and stop the Sterling Secure Proxy Engine V3.3.01 application.

Change Log Settings

SSP provides multiple log files including an audit log, secure proxy log, node logs, perimeter server log, SFTP log, and Certicom log.

Audit Log

The audit log contains messages about system operations and events. View the log for information about suspected misuse, and identify the user, application, or remote trading partner responsible for the misuse. The audit log provides proof that SSP functions and events occurred. It identifies the occurrence of malicious attack attempts. It can provide proof to resolve disputes with customers or legal entities, and prevent the payment of penalties for legal or service level agreement violations.

An audit log called `auditlog.xml` is created for both CM and the engine in the `install_dir/logs/audit` directory. An audit log record can be sent to a syslog daemon to be routed elsewhere for processing.

Audit log records are formatted in XML and are written to a file with an `.inc` suffix. Another file with suffix `.xml` contains an XML prolog and epilog information. The two files together make up one version of the audit log.

When an audit log file reaches a predefined size, it is archived and saved as `auditlog1.xml`. If archive files have already been created, each archive file is renamed. For example, when a new archive file is created, a log called `auditlog3.xml` is renamed to `auditlog4.xml` and `auditlog2.xml` is renamed to `auditlog3.xml`. You configure the maximum number of archive files to maintain.

Audit log settings are configured in the `log.properties` file located in the `install_dir/bin` directory.

Audit Log Parameters

You can modify the following parameters for an audit log in the `log.properties` file:

Parameter	Description
<code>audit.log.filename</code>	The location and file name to assign to an audit log. The default value is <code>./logs/audit/auditlog.xml</code> .
<code>audit.log.maxfilesize</code>	The number of files allowed in an audit log. When the <code>maxfilesize</code> is reached, the audit log is closed and a new log is opened. The default audit log file size is 500KB.
<code>audit.log.maxbackupindex</code>	Number of archive files to maintain. If the number identified in this parameter is exceeded, the oldest archive file is deleted. The default value is 100.
<code>audit.log.file.routing</code>	Determines if the audit log is written to a file. <code>y</code> = write the log to a file. <code>y</code> is the default setting. <code>n</code> = do not create an audit log file. Note: If you configure the audit log to write to the syslog daemon, this parameter can be set to <code>n</code> . Otherwise, an audit log is written to a file, regardless of the value of this parameter.

Parameter	Description
audit.log.syslog.routing	Determines if the audit log is written to syslog. y = write the log to syslog. n = do not write the audit log to syslog. n is the default setting. Configure a valid syslogd.port and syslogd.host in order to write to syslog.
audit.log.syslog.facility	Facility number to associate with audit log messages. The default value is 18.

Enable SysLog Support in the Audit Log

To route audit log content to a syslog in a UNIX or Linux environment, configure the following parameters in the log.properties file:

Parameter	Description
syslogd.enable	Enables syslog daemon support. y = enabled. n = disabled. n is the default setting.
syslogd.host	Name or IP address of the syslog host. The default value is the local host.
syslogd.port	UDP port where the syslog host receives log messages. The default is 514.

CM Audit Log Events

Following are the configuration events that are written to the CM audit log:

- A list of all fields when you create a new configuration object.
- Modify fields when you update a configuration object.
- A list of all fields when you delete an object.
- All fields of a configuration pushed to an engine.

Engine Audit Log Events

Following are the configuration events that are written to the engine audit log:

- All fields of an initial engine configuration received from CM.
- Changed fields from an engine configuration update from CM.
- Inbound connections received for all protocols.
- Inbound handshakes completed for the FTP, HTTP, and Connect:Direct protocols.
- Inbound login successes and failures for the FTP, HTTP, and SFTP protocols.
- Outbound connections established for all protocols.
- Outbound handshakes completed for the FTP, HTTP, and Connect:Direct protocols.
- Outbound login successes and failures for the FTP, HTTP, and SFTP protocols.

Secure Proxy Log

Use the secure proxy log to troubleshoot SSP issues. Secure proxy logs are created for the CM and the engine. The file is called `secureproxy.log` at the engine and `cms.log` at CM.

When a secure proxy log file reaches a predefined size, the current log is archived and the file name is changed to `secureproxy.log.1`. If archive files already exist, each archive file is renamed. For example, a log called `secureproxy.log.3` is renamed to `secureproxy.log.4` and a log `secureproxy.log.2` is renamed `secureproxy.log.3`. The maximum number of archive files to maintain is configured. Secure proxy log settings are configured in the `log.properties` file located in the `install_dir/bin` directory.

Secure Proxy Log Parameters

Following are the parameters that can be modified for a secure proxy log in the `log.properties` file:

Parameter	Description
<code>proxy.log.file.routing</code>	Determines if the secure proxy log is written to a file. y = write the log to a file. This value is the default. n=do not create a log file. Note: If you configure the secure proxy log to be written to the syslog daemon, this parameter can be set to n. Otherwise, a debug log is written to a file, regardless of the value of this parameter.
<code>proxy.log.filename</code>	The location and file name to assign to a log. The default value is <code>../logs/secureproxy.log</code> .
<code>proxy.log.maxfilesize</code>	The maximum file size allowed for a secure proxy log. When the maximum file size is reached, the debug log is closed and a new log is opened. The default log file size is 50MB.
<code>proxy.log.maxbackupindex</code>	The number of archive files to maintain. If the number of archive files identified in this parameter is exceeded, the oldest archive file is deleted. The default value is 10.
<code>proxy.log.level</code>	The logging level for the secure proxy log. The default value is INFO. This value can be set using CM.
<code>proxy.log.syslog.routing</code>	Determines if the secure proxy log is written to syslog. y = write the log to syslog. n is the default setting. n = do not write the debug log to syslog. Configure a valid <code>syslogd.port</code> and <code>syslogd.host</code> in order to write to syslog.
<code>proxy.log.syslog.facility</code>	Facility number to associate with secure proxy log messages. Default=17.

Secure Proxy File Output

Following are the fields in a secure proxy log:

Field	Format	Description
Date	dd-mm-yyyy where dd = day, mm = month, and yyyy = year.	The date the message was logged.
Time	hh:mm:ss:mss where hh = hours, mm = minutes, ss = seconds, mss = milliseconds.	The time the message was logged.
Session id	A 72-digit number	A number assigned to the session.
Name of component issuing log msg	"{"+name+"}"	The component that issues the message such as AcceptorThread:Secure
Logging level	ERROR, WARN, INFO, DEBUG	The type of logging that is written to the log.
Msg text	a text string	An explanation of the error message.

Node Logs

You can turn on node level logging to log sessions for a specific node. The node-level logs are named `secureproxy-<netmapName>.<nodeName>.log` where *netmapName* is the name of the netmap and *nodeName* is the name of the node for which activity is being logged.

When the sessions for a node end, the node-level log file for the session is closed. A new session appends to the end of the node log file. Both inbound and outbound nodes log both sides of the connection. Enabling logging on one of the nodes captures end-to-end session events.

Certicom Logs

Use the Certicom log to troubleshoot communications issues when using SSL or TLS. The file is called `certicom.log`.

Following are the parameters that can be modified for a Certicom log in the `log.properties` file:

Parameter	Description
<code>certicom.log.file.routing</code>	Determines if the certicom log is written to a file. y = write the log to a file. y is the default setting. n = do not create a log file. Note: If you configure the log to be written to the syslog daemon, this parameter can be set to n. Otherwise, a log is written to a file, regardless of the value of this parameter.
<code>certicom.log.filename</code>	The location and file name to assign to a log. Default= <code>./logs/certicom.log</code> .

Parameter	Description
certicom.log.maxfilesize	Maximum file size allowed for a certicom log. When the maximum is reached, the log is closed and a new log is opened. Default=100MB.
certicom.log.maxbackupindex	The number of archive files to maintain. If the number of archive files identified in this parameter is exceeded, the oldest archive file is deleted. The default value is 1.
certicom.log.level	Logging level for the certicom log. The default value is ERROR.
certocm.log.syslog.routing	Determines if the certicom log is written to syslog. y = write the log to syslog. n is the default setting. n = do not write the debug log to syslog. Configure a valid syslogd.port and .host in order to write to syslog.
certicom.log.syslog.facility	Facility number to associate with Certicom log messages. Default=17.

Perimeter Server Log

Perimeter server log information is written to a log file called perimeter.log. The default maximum size for the perimeter log is 100 MB.

When a log file reaches a predefined size, the current log is renamed and a new log is created. For example, an older log called perimeter.log1 is renamed to perimeter.log2 and the log perimeter.log2 becomes perimeter.log3.

Perimeter server log parameters are defined in the log.properties file. You can change one or more of the following parameters:

Parameter	Description
perimeter.log.file.routing	Determines if the perimeter log is written to a file. y = write the log to a file. y is the default setting. n = do not create a perimeter log file. Note: If you configure the perimeter log to be written to the syslog daemon, this parameter can be set to n. Otherwise, a perimeter log is written to a file, regardless of the value of this parameter.
perimeter.log.filename	The location and file name to assign to a perimeter server log. Default=../logs/perimeter.log.
perimeter.log.maxfilesize	The maximum size allowed in a perimeter server log. When the maxfilesize is reached, the perimeter server log is closed and a new log is opened. Default=100MB.
perimeter.log.maxbackupindex	The number of archive files to maintain. If the number of archive files identified in this parameter is exceeded, the oldest archive file is deleted. Default=1.
perimeter.log.level	The logging level for the perimeter log. The default value is ERROR. This value can be set using CM.

Parameter	Description
perimeter.log.syslog.routing	Determines if the perimeter log is written to syslog. y = write the log to syslog. n = do not write the log to syslog. n is the default setting. You must configure a valid syslogd.port and syslogd.host in order to write to syslog.
perimeter.log.syslog.facility	Facility number to associate with the log messages. Default=17.

SFTP Logs

If you configure SSP for an SFTP environment, two additional logs are maintained: a Maverick log and an SFTP adapter log.

Maverick Log

The Maverick toolkit is used to manage communications in an SFTP environment. All of the protocol messages generated by the Maverick toolkit are written to a log file called maverick.log. If you have problems in an SFTP environment, view this log to help troubleshoot the issue. File routing and syslog routing for a Maverick log are controlled by the proxy.log.file.routing and proxy.log.syslog.routing settings.

The default size of the maverick.log file is 100MB. The maverick log is set up to maintain one archive file so that when the maverick.log files reaches 100MB, a new file is created, and the archive file is renamed to maverick.log.1.

Following are the properties for the maverick log that you can change in the log.properties file:

Field	Description
maverick.log.filename	The location and file name to assign to a maverick server log. The default is ../logs/maverick.log.
maverick.log.maxfilesize	The maximum size of a maverick log file before archiving it and creating a new file. Default=100MB.
maverick.log.maxbackupindex	The number of backup files to maintain. The default value is 1.
maverick.log.level	The logging level to write to the maverick log file. Available options include: NONE, ERROR, WARN, INFO, and DEBUG. Default=INFO.

SFTP Adapter Log

A log is maintained for SFTP adapter activity. The file is called sftp.adapter-*<adapterName>*.log where *adapterName* is the name of the adapter as configured in SSP.

The SFTP adapter log is set up to maintain 10 archive files. When the log files reaches 50MB, a new file is created and the archive file is renamed to sftp.adapterAdapterA.log.1. If older versions exist, they will be renamed first. For example, an older log called sftp.adapter<*adapterName*>.log

is renamed to `sftp.adapter<adapterName>.log1` and the `sftp.adapter<adapterName>.log 2` is renamed `sftp.adapter<adapterName>.log 3`. The maximum number of versions to keep is configured in the `log.properties` file.

Following are the properties for the SFTP log that you can change in the `log.properties` file:

Field	Description
<code>sftp.log.enable</code>	Identifies if SFTP adapter messages are written to a separate log. Valid values are true false The default value is false. If this parameter is set to true, the adapter log information is written to the log file.
<code>sftp.log.filename</code>	Location and file name to assign to an SFTP adapter log. Default= <code>../logs/sftp.adapter-adaptername.log</code> where <code>adaptername</code> is the name assigned to the adapter in SSP.
<code>sftp.log.maxfilesize</code>	The maximum size of an SFTP log file before archiving it and creating a new file. Default=50MB.
<code>sftp.log.maxbackupindex</code>	The number of backup files to maintain. Default=10.

Start and Stop a Remote Perimeter Server

Use the procedures in this section to start and stop a remote perimeter server.

Start and Stop a Remote Perimeter Server on UNIX or Linux

To start a remote perimeter server on UNIX or Linux:

1. Change to the directory where the perimeter server is installed.
2. Type `startupPSService.sh`.

To stop a remote perimeter server on UNIX or Linux:

1. Change to the directory where the perimeter server is installed.
2. Type `stopPs.sh`.

Start and Stop a Remote Perimeter Server on Windows

You can start a perimeter server from a Windows service or from the command line.

To start a perimeter server from the command line on Windows:

1. Change to the directory where the perimeter server is installed.
2. Type `startPSService.cmd`.

You can stop a perimeter server from a Windows service or from the command line.

To stop a perimeter server from a command line on Windows:

1. Change to the directory where the perimeter server is installed.
2. Type `stopPs.cmd`.

Configure a Single Sign-on Connection to an HTTP Server

Sterling Secure Proxy (SSP) can be used as a proxy with Sterling File Gateway (SFG) and other HTTP applications and supports a single sign-on connection. Single sign-on (SSO) provides access control that allows a user to log in once to SSP, using the HTTP protocol, and then gain access to SFG without logging in again. SSO bypasses normal user authentication in SFG and trusts that SSP has authenticated the user.

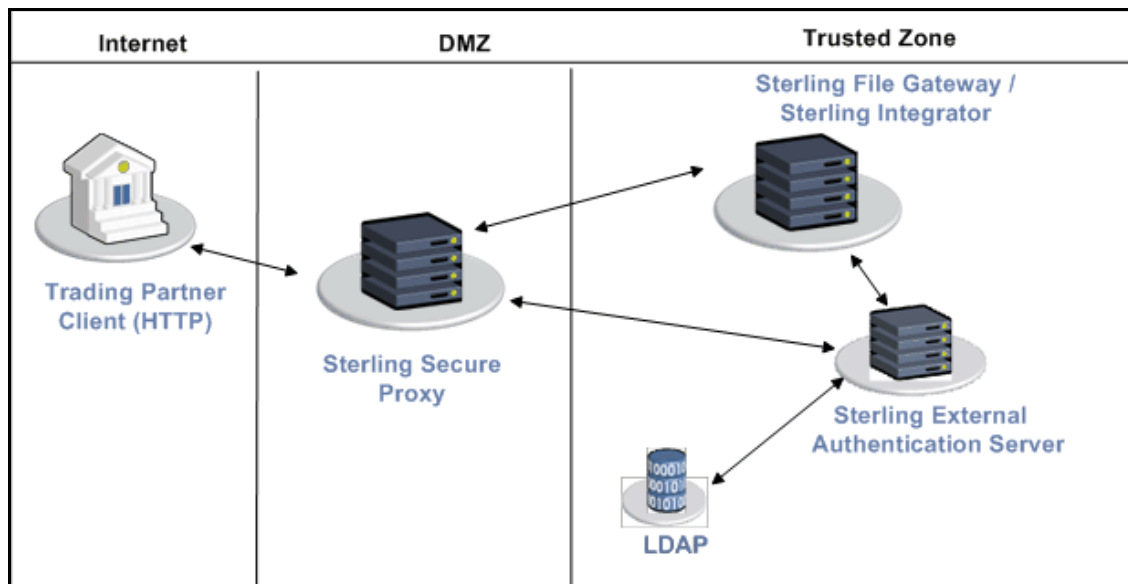
To support single sign-on, configure an SSP Login page and Sterling External Authentication Server (EA) to generate SSO tokens. Configuring SSO allows a trading partner to log on and use the same login session to connect to SSP and SFG. By default, EA uses OpenSAML to create and manage SSO tokens. However, you can customize your environment to use a third-party application to generate tokens.

This topic describes how to configure the HTTP protocol in SSP between the trading partner and SSP and between SSP and SFG to enable authentication through EA. It also describes how to configure EA to issue tokens to authenticate the connection between SSP and SFG, without the need to log in again for this connection.

Flow of Data for Single Sign-On Configuration Between SFG and SSP

After you set up the basic single sign-on configuration, trading partners can communicate in a secure environment that provides authentication. The trading partner first connects to SSP which then connects to SFG on behalf of the trading partner.

Following is an illustration of the flow of data:



Following are the steps that occur during a single sign-on session between a trading partner, SSP, and SFG when EA is used to generate and manage tokens:

1. The trading partner requests a connection to SFG.
2. SSP receives the request and the SSL handshake between SSP and the trading partner begins. If SSL authentication is configured, the proxy submits its certificate to the trading partner. If client authentication is configured, the trading partner then submits its certificate to SSP for authentication. You can optionally configure SSP to enforce client authentication and send the certificate to EA for validation.
3. SSP presents a Login page to the trading partner, who provides his user ID and password. If the HTTP policy is configured to use basic authentication, SSP sends an unauthorized response and the browser displays the browser user ID/password prompt.
4. SSP sends the user ID and password to EA and then validates it against information stored in LDAP.
5. If the credentials are valid, EA creates an OpenSAML v2 token and SSP returns a cookie associated with the token to the trading partner.
6. The trading partner sends an HTTP request to SSP and includes the cookie.
7. SSP checks for the cookie and validates the token using EA.
8. SSP then connects to SFG and performs an SSL handshake. It then sends the HTTP request with the cookie from the trading partner to SFG.
9. SFG then validates the token against EA and begins normal operation.

Configuration Considerations

Before you complete the single sign-on configuration, be aware of the following considerations:

Only the HTTP, Connect:Direct, FTP, and SFTP protocols supports single sign-on connections.

Each single sign-on user you create in SFG must be modified in the Sterling Integrator User Accounts as an External user with the correct Authentication Host. Sterling Integrator uses the specified *Authentication Host* to authenticate the user.

The myFileGateway, FileGateway, and Sterling Integrator dashboard users use application authentication in the HTTP policy.

The Sterling Integrator AS2 and WebDav users use basic authentication in the HTTP policy.

Customize the SSP Login page—When you configure the basic scenario and select Application Authentication in the HTTP policy, you use the default SSP Login page. The default page provides basic information, including user name and password. To customize this page, to include additional information and your logo, complete the procedure, *Customize the Login Page* on page 374.

Organization of Single Sign-On Scenarios

The scenarios describe how to configure single sign-on between SSP and HTTP-enabled trading partners and between SSP and SFG.

Configure the Basic Scenario to Enable a Connection to the myFileGateway Application

Configure the basic scenario to allow you to quickly become operational using single sign-on to connect to myFileGateway, the trading partner interface in SFG. After you complete this scenario, test the connection to ensure that you have correctly configured it. After you determine that it works, add SSL/TLS support. You then have a basic configuration and can begin operation.

Configure Advanced Features

After you configure the basic SSO setup, determine if your environment requires an advanced feature. Following are the advanced features:

Customize the OpenSAML v2 tokens—You use the default token generation definition when you configure the basic single sign-on definition. To customize the token definition, you can modify the named identity provider, the token signing key, or how long a token can be used before it expires. Refer to *Customize the SSO Cookie Attributes* on page 353.

Configure Sterling Integrator or SFG with additional pools—You use additional pools to support more than one EA server. Refer to *Configure Sterling Integrator or SFG to use multiple EA servers* on page 354.

Use a third-party application to configure tokens—The basic scenario uses EA to configure and manage tokens. To use a third-party application to configure tokens, complete additional setup procedures. Refer to *Allow a Third-Party Provider to Create Tokens* on page 356.

Configure a Single Sign-on connection to the FileGateway Application—After you configure the basic SSO setup for myFileGateway, determine if internal company users require the ability to connect to the FileGateway application through SSP. Refer to *Add Single Sign-On Support for FileGateway* on page 358.

Configure a Single Sign-on connection to other applications on Sterling Integrator—This includes the Dashboard and Mailbox interfaces. Refer to *Configure Single Sign-On for Dashboard* on page 362.

Configure a Single Sign-on connection to HTTP services on Sterling Integrator that use Basic Authentication—This includes applications such as WebDav and AS2. Refer to *Add Single Sign-On Support for Basic Authentication Applications on Sterling Integrator* on page 370.

Customize the Logon Portal—You use the default Logon Portal when you configure an HTTP adapter with SSP. To customize the Logon Portal, you can modify the Logon Portal pages and user messages, or you can configure SSP to use an external logon portal. Refer to *Customize the Logon Portal* on page 371.

Configure Optional Features

SSP provides the following optional features and you can configure them as required for your environment. These features are available for SSO and non-SSO configurations. Refer to Protocol configuration information on for instructions on how to configure the following features:

- Modify the HTTP connection requirement between SSP and inbound nodes by defining a specific IP address, a wildcard peer pattern, or an IP/subnet pattern

- Secure the outbound HTTP connection between SSP and SFG using SSL or TLS

- Authenticate an inbound certificate using EA

- Define alternate nodes for failover support

EA provides the ability to configure multifactor authentication. In addition to configuring client authentication in SSP, EA can authenticate the IP address, certificate, password, and/or group access.

Worksheets

Before you complete each procedure, gather the information you need to configure it, on the worksheet provided. For each worksheet:

Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

Accept default values for fields not listed.

Note the Configuration Manager field where you will specify the value.

Field Definitions

Field definitions are provided for all single sign-on functions. Refer to online help for definitions.

Basic Single Sign-On Scenario for myFileGateway

Complete the following tasks to define an HTTP configuration between a trading partner and SSP and between SSP and SFG to support a single sign-on connection for myFileGateway:

Configure SSP to support basic single sign-on

Use the default single sign-on configuration in EA to manages OpenSAML v2 tokens

Prepare SFG to support the single sign-on option

Validate connections between the trading partner, SSP, and SFG

Configure SSP for Basic Single Sign-On

Complete the following procedures to configure SSP for basic single sign-on:

Create an SSO configuration.

Create an SSP policy to support a single sign-on connection to SFG.

Define a netmap to identify inbound and outbound connections.

Define an HTTP adapter.

Create an SSO Configuration

Before you create an SSO configuration, gather the following information:

Configuration Manager Field	Feature	Value
Name	SSO configuration file.	
Default Landing Page	Identify the SFG application to connect to.	/myfilegateway

To define an SSO configuration:

1. Click Advanced from the menu bar.
2. Click Actions>New SSO Configuration.
3. On the Basic tab, type a configuration name in the Name field.
4. Type the fully qualified host name that the trading partner will use to connect to myFileGateway in the Fully Qualified Host Name field.
5. Click on the Advanced tab.
6. Type /myfilegateway in the Default Landing Page field.
7. If this is for an unsecure connection (HTTP, rather than HTTPS), you must uncheck the SSO Cookie Secure Flag.
8. Click Save.

Create an HTTP Policy to Support a Single-Sign On Connection

To create an HTTP policy to support a single sign-on connection to SFG:

1. Select Application Authentication in the User Authentication Type field. The values, *Through External Authentication* and *SSO token from External Authentication*, are selected by default.
 - ♦ If the trading partner uses a non-browser client, select Basic Authentication in the User Authentication Type field. Enable *Through External Authentication* and enable *SSO token from External Authentication*.
2. Type the definition you defined in EA in the External Authentication Profile field.

For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Create an HTTP Netmap to Support a Single Sign-On Connection to myFileGateway

To create an HTTP netmap to support a single sign-on connection to myFileGateway:

1. Configure the inbound node information for your external trading partners. Select the policy defined for SSO in the preceding section.
2. Configure the outbound node information for your SFG server.

Define the HTTP Reverse Proxy Adapter Used for the Single Sign-On Connection

To create an HTTP adapter to support a single sign-on connection to myFileGateway:

1. Specify standardRouting for the Routing Type field.
2. Specify the netmap you created for the single sign-on connection to SFG for the Netmap field.
3. Specify the SSO configuration you created for the single sign-on connection to SFG for the SSO Configuration field.
4. Specify your EA server in the External Authentication Server field.

For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Configure Sterling EA to Support Single Sign-On

To allow an SSO connection between a trading partner and SSP to route traffic to SFG, you configure OpenSAML v2.0 tokens in EA. You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines which options are enabled. Refer to the Sterling External Authentication Server documentation library for instructions on configuring an EA definition.

The EA server generates and manages tokens. A default configuration called SEAS-SAML is enabled when you install EA. If you use the default configuration, EA is the identity provider, token signing keys are automatically generated, and the token expires after 15 minutes. Use the default configuration when you configure basic single sign-on. To customize EA for single sign-on, refer to *Customize Token Definitions Created by EA* on page 357.

Prepare SFG to Support Single Sign-On

Before you enable single sign-on between a trading partner and SFG, when using SSP, you modify the SFG installation. The files required to enable SSO are installed with EA.

Prepare SFG to Support Single Sign-On on UNIX or Linux

To prepare SFG to support SSO on UNIX or Linux:

1. From the EA server, copy the files and subdirectories from the `EA_install_dir/lib/sterling/sfg-sso-plugin` directory to a location that is accessible to the SFG server, where `EA_install_dir` is the location of the EA installation.

Note: If you use FTP to copy the files to the SFG server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the SFG server, move to the `SFG_install_dir/properties` directory, where `SFG_install_dir` is the SFG installation directory.
3. Type the following commands to copy the SSO properties files to the SFG server, where `base_dir` is the location where you copied the files in step 1:

```
cp base_dir/sfg-sso-plugin/properties/security.properties_seas-sso_ext.in .
cp base_dir/sfg-sso-plugin/properties/authentication_policy.properties_seas-auth_ext.in .
cp base_dir/sfg-sso-plugin/properties/servers.properties_seas-sso_ext .
cp base_dir/sfg-sso-plugin/properties/servers.properties_seas-auth_ext .
```

4. Stop SFG if it is running.
5. In the `servers.properties_seas-sso_ext` file, uncomment the following line and replace `<SI_install>` with the actual installation path for SFG:

```
# seas-sso=<SI_install>/properties/seas-sso/1.0/seas-sso.properties
```

6. In the `servers.properties_seas-auth_ext` file, uncomment the following line and replace `<SI_install>` with the actual installation path for SFG:

```
# seas-auth=<SI_install>/properties/seas-auth/1.0/seas-auth.properties
```

7. From the `SFG_install_dir/bin` directory, type the following commands:

```
./install3rdParty.sh seas-ssso 1.0 -j base_dir/sfg-ssso-plugin/seas-ssso.jar
./install3rdParty.sh seas-ssso 1.0 -p base_dir/sfg-ssso-plugin/properties/seas-ssso.properties
./install3rdParty.sh seas-auth 1.0 -p base_dir/sfg-ssso-plugin/properties/seas-auth.properties
```

8. From the `SFG_install_dir/jar/seas-ssso/1.0` directory, create a subdirectory named `private`.
9. Move to the `/private` directory.
10. Type the following command to copy the jar files to the `/private` directory on the SFG server:

```
cp base_dir/sfg-ssso-plugin/private/*.jar .
```

Modify SFG to Support Single Sign-On on UNIX or Linux

Before SFG supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to `customer_overrides.properties` to prevent custom settings from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify SFG to enable single-sign on:

1. In the `install_dir/properties` directory, locate or create the `customer_overrides.properties` file.
2. Open the file in a text editor and add the properties that you want to override.
 - a. Add the following values to configure single sign-on:
 - `security.SSO_FORWARD_URL.MYFILEGATEWAY.LOGOUT=/Signon/logout`
 - `security.SSO_FORWARD_URL.MYFILEGATEWAY.TIMEOUT=/Signon/timeout`
 - `security.SSO_FORWARD_URL.MYFILEGATEWAY.VALIDATION_FAILED=/Signon/validationerror`
 - b. Add the following connection parameters to configure the SFG connection to EA:
 - `seas-ssso.EA_HOST=IP address or host name of EA server`
 - `seas-ssso.EA_PORT=listen port of EA server`
Specify the appropriate secure or clear listen port from the EA server configuration.
 - `seas-ssso.EA_PS_NAME=perimeter server used to connect to EA`
Specify `local` if you do not use a perimeter server to connect to EA.

- `seas-ss0.EA_SECURE_CONNECTION=true` or `false`
true sets connections to EA as secure and *false* sets the connection as clear. If this parameter is true, you must also define the `EA_SYSTEM_CERT` and `EA_TRUSTED_CERT[1]`.
- `seas-ss0.EA_SYSTEM_CERT=name` of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.
- `seas-ss0.EA_TRUSTED_CERT[1]=name` of the trusted certificate used for secure connections to EA. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the `seas-ss0.EA_TRUSTED_CERT(#)` parameter. For example, for the first certificate, configure the parameter, `seas-ss0.EA_TRUSTED_CERT[1]`; for the second certificate, define `seas-ss0.EA_TRUSTED_CERT[2]`, until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart SFG to use the new values.

Prepare SFG to Support Single Sign-On on Windows

To prepare SFG to support SSO on Windows:

1. From the EA server, copy the files from the `EA_install_dir\lib\sterling\sfg-ss0-plugin` directory to a location that is accessible by the SFG server.

Note: If you use FTP to copy the files to the SFG server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the SFG server, move to the *SFG_install_dir*\properties directory.
3. Type the following commands to copy the SSO security.properties files to the SFG server, where *base_dir* is the location where you copied the files in step 1:

```
copy      \sfg-sso-plugin\properties\security.properties_seas-sso_ext.in .
copy      \sfg-sso-plugin\properties\authentication_policy.properties_seas-auth_ext.in .
copy      \sfg-sso-plugin\properties\servers.properties_seas-sso_ext .
copy      \sfg-sso-plugin\properties\servers.properties_seas-auth_ext .
```

4. Stop SFG if it is running.
5. In the server.properties_seas-sso_ext file, uncomment the following line and replace <SI_install> with the actual installation path for SFG:

```
# seas-sso=<SI_install>\properties\seas-sso\1.0\seas-sso.properties
```

6. In the server.properties_seas-auth_ext file, uncomment the following line and replace <SI_install> with the actual installation path for SFG:

```
# seas-auth=<SI_install>\properties\seas-auth\1.0\seas-auth.properties
```

7. From the *SFG_install_dir*\bin directory, type the following commands:

```
install3rdParty.cmd seas-sso 1.0 -j      \sfg-sso-plugin\seas-sso.jar
install3rdParty.cmd seas-sso 1.0 -p      \sfg-sso-plugin\properties\seas-sso.properties
install3rdParty.cmd seas-auth 1.0 -p     \sfg-sso-plugin\properties\seas-auth.properties
```

8. From the *SFG_install_dir*\jar\seas-sso\1.0 directory, create a subdirectory named private.
9. Go to the \private directory.
10. Type the following command to copy the jar files to the SFG server:

```
copy      \sfg-sso-plugin\private\*.jar .
```

Modify SFG to Support Single Sign-On on Windows

Before SFG is configured to support single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to a file called *customer_overrides.properties*. This prevents custom settings from being overwritten when you apply patches. The *customer_overrides.properties* file is not changed during upgrades or patches. If the *customer_overrides.properties* file is not present, you must create it. Refer to the Sterling Integrator *customer_overrides.properties* topic for more information.

To modify SFG to enable single sign-on:

1. In the *install_dir*\properties directory, locate or create the *customer_overrides.properties* file.

2. Open the file in a text editor and add the properties that you want to override.
 - a. Add the following values to configure single sign-on:
 - security.SSO_FORWARD_URL.MYFILEGATEWAY.LOGOUT=
 \Signon\logout
 - security.SSO_FORWARD_URL.MYFILEGATEWAY.TIMEOUT=
 \Signon\timeout
 - security.SSO_FORWARD_URL.MYFILEGATEWAY.
 VALIDATION_FAILED=\Signon\validationerror
 - b. Add the following connection parameters to configure the SFG connection to EA:
 - seas-ss0.EA_HOST=*IP address or host name of EA server*
 - seas-ss0.EA_PORT=*listen port of EA server*
Specify the appropriate secure or clear listen port from the EA server configuration.
 - seas-ss0.EA_PS_NAME=*perimeter server used to connect to EA*
Specify *local* if you do not use a perimeter server to connect to EA.
 - seas-ss0.EA_SECURE_CONNECTION=*true* or *false*
true sets connections to EA as secure and *false* sets the connection as clear.
If this parameter is true, you must also define the EA_SYSTEM_CERT and EA_TRUSTED_CERT[1].
 - seas-ss0.EA_SYSTEM_CERT=*name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.*
 - seas-ss0.EA_TRUSTED_CERT[1]=*name of the trusted certificate used for secure connections to EA. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.*
If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the seas-ss0.EA_TRUSTED_CERT(#) parameter. For example, for the first certificate, configure the parameter, seas-ss0.EA_TRUSTED_CERT[1]; for the second certificate, define seas-ss0.EA_TRUSTED_CERT[2], until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart SFG to use the new values.

Change Sterling Integrator User Accounts for Single Sign-On

After you install and configure the SSO plug-in and restart Sterling Integrator, the Sterling Integrator User Accounts page presents additional choices for Authentication Type. To enable SSO for a user account, select an authentication method of External and then select the appropriate EA server for Authentication Host. The default is SEAS Authentication. Additional choices are available only if you define other EA Connection pools in addition to the default SSO_POOL.

Verify That SFG is Configured for Single Sign-On

Before you configure additional functions, make sure that SFG is ready for use in a single sign-on environment. To verify the configuration, start SFG.

View the authentication.log and security.log to make sure the SFG files are updated. If the update was successful, log files display the success messages.

Authentication.log file displays the following messages:

```
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SI is configured to support
single sign-on
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SSO Property :
SSO_AUTHENTICATION_CLASS.1 = Class name :
com.sterlingcommerce.seas.gis.sso.plugin.SeasSsoProvider
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication - A new SSO
Authentication Policy has been installed.
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication on Page:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager Number of SSO Authentication
Plug-In:1
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO configuration
policy...SSOAuthenticationPolicy isComplete=true isEnabled=true
httpUserIdHeader=SM_USER
ALL 000000000000 GLOBAL_SCOPE SecurityManager initialization complete.
```

Security.log displays the following message:

```
ALL 000000000000 GLOBAL_SCOPE SEAS PLUGIN: Plug-in initialized
```

Verify the SSP Connections

To verify that the engine can receive and initiate communications sessions after configuring the basic single sign-on environment, establish a connection between an HTTP client and the HTTP Reverse Proxy adapter to ensure that the SSP Login page is displayed.

Note: Configuration files must be available on the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

- Establish an HTTP session initiated by a trading partner using an HTTP client
- Initiate an outbound session to an SFG server on behalf of the HTTP client connection

To verify the communications sessions:

1. Make sure the engine is running.
2. From a web browser, initiate an HTTP client request to the SSP server for the myFileGateway URL, `http://ssp_host:port/myfilegateway`.
3. View the SSP Login page used for a single sign-on session.
4. Provide valid logon credentials.
5. View the myFileGateway home page.

If you can view myFileGateway home page, you have confirmed that the connections are working.

Advanced Features of the Single Sign-On Configuration

This section provides instructions on the additional features you can configure by modifying the basic single sign-on scenario. Variations include:

- Customize the SSO Cookie attributes
- Configure Sterling Integrator or SFG to use multiple EA servers
- Allow a third-party provider to create tokens
- Customize token definitions created by EA
- Customize the Login page

Customize the SSO Cookie Attributes

To implement single sign-on, you use single sign-on attributes. When you configure the basic scenario, you use default attributes. You can customize these settings, including the name of the cookie containing the SSO token, HTTP header associated with a user ID, and the attributes associated with a token. Tokens can be generated with EA or with a third-party application. This procedure assumes you are using EA. Refer to *Allow a Third-Party Provider to Create Tokens* on page 356 for instructions on configuring an external application to generate tokens.

Before you customize this page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the single sign-on configuration.	_____
Front End SSO Token Cookie Name (Inbound)	Name to assign the cookies associated with each token. This value must match the definition in SFG.	_____
Back End SSO User Header Name (Outbound)	The HTTP header name containing the user name that is sent to SFG.	_____
Back End SSO Token Cookie Name (Outbound)	Name to assign the cookies associated with each token. This value must match the definition in SFG.	_____

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.

- ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
- 3. To customize the front-end definitions, edit the Front End SSO Token Cookie Name field.
- 4. To customize the back-end definitions, edit the following fields:
 - ◆ Back End SSO User Header Name
 - ◆ Back End SSO Token Cookie Name

Note: The values defined in the back-end fields must match these settings on SFG/Sterling Integrator system:

```
## HTTP header containing the SSO user
security.SSO_USER_HEADER=SM_USER
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-sso.SSO_TOKEN_COOKIE=SSOTOKEN
```

Refer to the SFG documentation for instructions.

5. Click Save.

Configure Sterling Integrator or SFG to use multiple EA servers

If you are implementing single sign-on for HTTP with Basic Authentication, or for protocols other than HTTP, and you need to support additional EA servers, add the following parameters for each additional EA pool configuration to the `customer_overrides.properties` file located in the `install_dir\properties` directory.

Note: The SFG myFileGateway and FileGateway applications always use the default SSO_POOL EA connection to validate SSO tokens, regardless of which Authentication Host is selected for the user. Additional EA connection pools may only be used for HTTP Basic Auth applications, FTP, SFTP, and Connect:Direct.

```
authentication_policy.authentication_n.className=
com.sterlingcommerce.seas.gis.sso.plugin.SeasAuthentication
authentication_policy.authentication_n.display_name =
name to be used on the Sterling Integrator/SFG user administration UI. Use something
different than the default SEAS Authentication, which is used by the default SSO_POOL. This
is the Authentication Host name that is selected when you configure external User Accounts to
use this pool.
authentication_policy.authentication_n.enabled=true
seas-auth.authentication_n.profile = userAuth
seas-auth.authentication_n.ea_pool=unique name for your pool other than the default
SSO_POOL, which shares the EA connection pool with the SFG SSO configuration
```

Note: Change the "n" in the above example to a number greater than 1 to avoid overwriting the default SSO_POOL, which is shared with the FileGateway/myFileGateway SSO configuration. Also, make sure you avoid using a number already in use for LDAP authentication. Define a unique number for each entry.

To use another connection pool instead of the default SSO_POOL, configure the EA connection of the pool with the following parameters, where *pool* is the pool name defined in the preceding section:

seas-auth.pool.EA_HOST=IP address or host name of EA server

seas-auth.pool.EA_PORT=listen port of EA server

seas-auth.pool.EA_PS_NAME=perimeter server used to connect to EA

seas-auth.pool.EA_SECURE_CONNECTION=true or false

true sets connections to EA as secure and *false* sets the connection as clear. If this parameter is true, you must also define the EA_SYSTEM_CERT and EA_TRUSTED_CERT[1].

seas-auth.pool.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure

seas-auth.pool.EA_TRUSTED_CERT[1]=name of the trusted certificate used for secure connections to EA

seas-auth.pool.TIMEOUT=maximum time to wait for making EA connections and receiving responses

seas-auth.pool.TIMEOUT_UNITS=unit of time to use, minutes or seconds, for seas-auth.pool.TIMEOUT parameter

seas-auth.pool.PERSISTENT_EA_CONNECTIONS=whether to keep persistent connections to EA

true sets connections to EA as persistent and *false* sets the connections as not persistent.

seas-auth.pool.MAX_EA_CONNECTIONS=maximum number of EA connections

Note: Additional fields can be added if you wish to override the defaults shown below:

SEAS-SSO Configuration

HTTP cookie containing the SSO token

seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

Maximum time to wait for making EA connections and receiving responses

seas-ss0.SSO_TIMEOUT=30

seas-ss0.SSO_TIMEOUT_UNITS=seconds

Whether to keep persistent connections to EA

seas-ss0.PERSISTENT_EA_CONNECTIONS=true

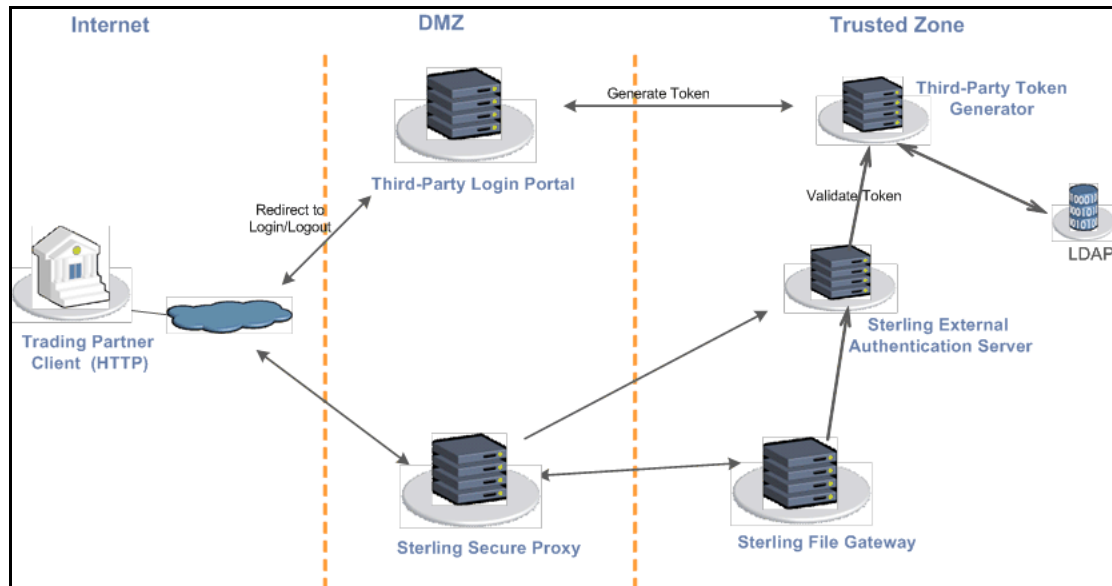
Maximum number of EA connections

seas-ss0.MAX_EA_CONNECTIONS=1

Allow a Third-Party Provider to Create Tokens

You used the default OpenSAML token generation definition when you configured the basic single sign-on definition. The default configuration uses EA to manage tokens. To use a third-party application for token generation, you must modify the SSO Token setup in EA.

The following diagram illustrates the flow using a third-party application for token generation.



Configure EA to Enable a Third-Party Provider to Create Tokens

You can configure EA to validate a token generated by a third-party login application using a custom class. You must first verify that a custom class exists that EA Server can use to verify tokens generated by the third-party application. Refer to the Sterling External Authentication Server documentation library for more information.

Before you configure EA to enable a third-party application to create tokens, gather the following information:

Configuration Manager Field	Feature	Value
Token Manager	The application that creates the tokens.	Custom
Class Name	Name of the Java class that implements the token manager interface.	
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To configure EA and enable a third-party application to generate tokens:

1. Log on to EA.

2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. To configure a token manager other than EA, select Custom from the Token Manager field.
5. Type the class name in the Class name field.
6. To change how long a token can be used before it expires, type a new value in the Token Expiration Period field.
7. Click OK.

Customize SSP to Use a Login Portal of a Third-Party Application

You can configure SSP to redirect connections to a third-party login portal for authentication and SSO token generation. Before doing this, you should verify that a custom class exists that EA Server can use to verify tokens generated by the third-party application. Before you configure SSP to enable the login page of a third-party application, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the single sign-on file.	
External Portal External Application Login URL	External login portal URL where the user is authenticated and a token is generated.	

To configure the SSO Configuration in SSP:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. Click the Logon Portal tab.
4. Select the External portal option. To identify the URL of the application being used to generate tokens, type the URL in the External Application Login URL field.
5. Click Save.

Customize Token Definitions Created by EA

You used the default token definition when you configured the basic single sign-on definition. To customize the token definition, complete the following procedure. You can modify the named

identity provider, the token signing key, or how long a token can be used before it expires. Refer to the Sterling External Authentication Server documentation library for more information.

Before you customize token definitions, gather the following information:

Configuration Manager Field	Feature	Value
Named Identity Provider	Prefix appended to generated tokens to identify the provider. Note: If you change the provider name, any outstanding tokens are invalid.	
Token Signing Key	Alias of the key certificate to sign the token.	
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To customize the token configuration in EA:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. Customize one or more of the following definitions:
 - ◆ Named Identity Provider
 - ◆ Token Signing Key
 - ◆ Token Expiration Period
5. Click OK.

Add Single Sign-On Support for FileGateway

After you configure single sign-on for myFileGateway, determine if internal company users require access to FileGateway through SSP.

Configure Single Sign-On for FileGateway

Complete the following procedures to configure SSP for basic single sign-on for FileGateway:

Create an SSO Configuration for FileGateway on page 359

Create an HTTP Policy to Support SSO for FileGateway on page 359

Define the HTTP Netmap for FileGateway on page 359

Configure HTML Rewrite on page 359

Configure an HTTP Adapter for FileGateway on page 360

Create User Accounts in SFG on page 361

Verify the SSP Connections on page 362

Create an SSO Configuration for FileGateway

To configure SSO for internal users, make a copy of the myFileGateway SSO configuration. Rename the copy to identify the configuration as a FileGateway definition. Change the field called Default Landing Page to connect to /filegateway. Refer to *Create an SSO Configuration* on page 344 for more information.

Note: You can also use the included Welcome Page, which includes links to FileGateway and myFileGateway and can be customized to include other applications.

Create an HTTP Policy to Support SSO for FileGateway

To create an HTTP policy to support a single sign-on connection to FileGateway:

1. Select Application Authentication in the User Authentication Type field. The values, *Through External Authentication* and *SSO token from External Authentication*, are selected by default.
2. Type the definition you defined in EA in the External Authentication Profile field.

For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Define the HTTP Netmap for FileGateway

Make a copy of the myFileGateway HTTP netmap. Rename the copy to identify the configuration as a FileGateway definition. Configure the inbound node definitions for the nodes that need to connect to FileGateway. Configure the outbound node definition to support the connection to FileGateway. Change the values as needed. Refer to *Create an HTTP Netmap to Support a Single Sign-On Connection to myFileGateway* on page 345 for more information.

Configure HTML Rewrite

HTML rewriting allows you to replace the URL links returned by an HTTP server to the SSP server by configuring how the URL links from the server will be mapped to the URL links in SSP. If the HTTP server has web pages with links to other web pages, you must map the URL connections in order for the links to work. You must configure HTML rewrite in order for the FileGateway application to function correctly.

Certain pages on the dashboard use javascript to create URL dynamically using a back-end host name literal. To change this host name to the proxy host name, you add another entry to the HTML rewrite. Use the third entry in the table below as an example.

Note: HTML rewrite may not work for arbitrary javascripts that dynamically create URL at the client side.

To configure this environment, use the following table to help you identify the rewrite values to define in the netmap definition:

Server URL	Proxy URL
http(s)://<fully qualified DNS name for the SFG_host>:<port>	http(s)://<fully qualified DNS name for the proxy_host>:<adapter1_port>
http(s)://<SFG_ipaddress>:<port>	http(s)://<proxy_ipaddress>:<port>
http(s)://<fully qualified DNS name for the SFG_host>	http(s)://<fully qualified DNS name for the proxy_host>
<fully qualified DNS name for the Sterling Integrator host>	<fully qualified DNS name for the proxy host>

To configure HTML rewrite:

1. Click Configuration from the menu bar.
2. Expand the netmap definition you created.
3. On the HTTP Netmap Nodes panel, click the HTML Rewrite tab.
4. Click New.
5. Enable the Support HTML Rewrite field.
6. Type the URL path for the outbound server in the Server URL field.
7. Type the URL path for the proxy in the Proxy URL field.
8. Click Save.
9. Repeat steps 4 through 8 for all HTML Rewrite options you want to configure.
10. To reorder the HTML rewrite definitions:
 - a. Click the radio button beside the URL routing definition to reorder.
 - b. Click Move Up or Move Down until the item is in the correct order.
11. Click Save.

Test the configuration to ensure that single sign-on and HTML rewrite to the SFG server is configured correctly.

Configure an HTTP Adapter for FileGateway

Make a copy of the myFileGateway HTTP adapter. Rename the copy to identify the configuration as a FileGateway configuration. Enable Support HTML Rewrite. Specify the SSO configuration and HTTP netmap configurations you created for FileGateway. Refer to *Define the HTTP Reverse Proxy Adapter Used for the Single Sign-On Connection* on page 345 for more information.

Create User Accounts in SFG

User accounts work with permissions to provide security for your organization. These features make it possible to regulate which users have access to each module in SFG and what functions each user can perform. To create a user account:

1. From SFG, select Tools > B2B Console.
2. From within Gentran Integration Suite, Select Accounts > User Accounts > Create a new Account.
3. Complete the steps in the wizard. Supply the following information about the user:
 - ◆ Authentication type (external)
 - ◆ User ID
 - ◆ Password (For the default user policy, the password must be six characters or more and contain at least two of the following characters. (number, capital letter, !, @, #, \$, %, ^, &, *)
 - ◆ Confirm Password
 - ◆ Policy (Default User Policy)
 - ◆ SSH Authorized User Key
 - ◆ Session Timeout (in minutes)
 - ◆ Accessibility (Dashboard UI)
 - ◆ Dashboard Theme (Default)
4. Select one or more of the following groups to assign the user to, based on their job responsibilities:
 - ◆ File Gateway Integration Architects
 - ◆ File Gateway Operators
 - ◆ File Gateway Route Provisioners
 - ◆ File Gateway System Administrators

Note: Do not assign the FileGateway user to the trading partner group. Otherwise, the user will not be able to login to the File Gateway application.

Note: For full SFG functionality, each of these groups must have at least one user. By default, the following users are created during installation: fg_sysadmin, fg_architect, fg_provisioner, and fg_operator. One user can belong to multiple groups.

Note: To create the equivalent of fg_sysadmin, assign all the File Gateway groups listed above and the Sterling Integrator Admin group to the user.

5. Supply the following information for the user:

- ◆ First Name
 - ◆ Last Name
 - ◆ E-mail
 - ◆ Pager
 - ◆ Preferred Language (English, Japanese)
 - ◆ Manager ID
 - ◆ Identity
6. Review and confirm the user to create the new user account.

Verify the SSP Connections

To verify that the engine can receive and initiate communications sessions after configuring the basic single sign-on environment, establish a connection between an HTTP client and the HTTP reverse proxy adapter to ensure that the SSP Login page is displayed. Refer to *Verify the SSP Connections* on page 352 for more information.

If you can view the FileGateway home page, you have confirmed that the connections are working. You are ready to add SSL or TLS support to the inbound connection. For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Add Single Sign-On Support for Sterling Integrator Dashboard

After you configure single sign-on for myFileGateway, determine if internal company users require access to Sterling Integrator dashboard through SSP.

Configure Single Sign-On for Dashboard

Complete the following procedures to configure SSP for basic single sign-on for dashboard:

Create an SSO Configuration for Dashboard on page 362

Create an HTTP Policy to Support SSO for Dashboard on page 363

Define the HTTP Netmap for Dashboard on page 363

Configure HTML Rewrite on page 363

Configure an HTTP Adapter for Dashboard on page 364

Create User Accounts in Sterling Integrator on page 368

Verify the SSP Connections on page 369

Create an SSO Configuration for Dashboard

To configure SSO for dashboard users, make a copy of the myFileGateway SSO configuration. Rename the copy to identify the configuration as a dashboard definition. Change the field called Default Landing Page to connect to /dashboard/sso.jsp. Refer to *Create an SSO Configuration* on page 344 for more information.

Note: You can also select the Welcome Page as the Default Landing Page when you configure the SSO configuration.

Create an HTTP Policy to Support SSO for Dashboard

To create an HTTP policy to support a single sign-on connection to Sterling Integrator Dashboard:

1. Select Application Authentication in the User Authentication Type field. The values, *Through External Authentication* and *SSO token from External Authentication*, are selected by default.
2. Type the definition you defined in EA in the External Authentication Profile field.

For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Define the HTTP Netmap for Dashboard

Make a copy of the myFileGateway HTTP netmap. Rename the copy to identify the configuration as a dashboard definition. Configure the inbound node definitions for the nodes that need to connect to dashboard. Configure the outbound node definition to support the connection to dashboard. Change the values as needed. Refer to *Create an HTTP Netmap to Support a Single Sign-On Connection to myFileGateway* on page 345 for more information.

Configure HTML Rewrite

HTML rewriting allows you to replace the URL links that refer to the Sterling Integrator server in the HTML pages returned by the HTTP server with the links pointing to SSP. When the user clicks on these links, the HTTP requests come back to SSP. You must configure HTML rewrite in order for the dashboard application to function correctly.

Certain pages on the dashboard use javascript to create URL dynamically using a back-end host name literal. To change this host name to the proxy host name, you add another entry to the HTML rewrite. Use the fourth entry in the table below as an example.

Note: HTML rewrite may not work for arbitrary javascripts that dynamically create URL at the client side.

To configure this environment, use the following table to help you identify the rewrite values to define in the netmap definition:

Server URL	Proxy URL
http(s)://<fully qualified DNS name for the Sterling Integrator_host>:<port>	http(s)://<fully qualified DNS name for the proxy_host>:<adapter1_port>
http(s)://<Sterling Integrator_ipaddress>:<port>	http(s)://<fully qualified DNS name for the proxy_host>:<adapter1_port>

Server URL	Proxy URL
http(s)://<fully qualified DNS name for the Sterling Integrator_host>:	http(s)://<fully qualified DNS name for the proxy_host>:
http(s)://<fully qualified DNS name for the Sterling Integrator_host>	http(s)://<fully qualified DNS name for the proxy_host>
<fully qualified DNS name for the Sterling Integrator host>	<fully qualified DNS name for the proxy host>

Note: Ensure that the Destination Address field in the outbound node has the fully qualified DNS name of the Sterling Integrator host. If not, the URLs in the HTML pages that reference the Sterling Integrator host will not match the host URL entered for the HTML rewrite.

To configure HTML rewrite:

1. Click Configuration from the menu bar.
2. Expand the netmap definition you created.
3. On the HTTP Netmap Nodes panel, click the HTML Rewrite tab.
4. Click New.
5. Enable the Support HTML Rewrite field.
6. Type the URL path for the outbound server in the Server URL field.
7. Type the URL path for the proxy in the Proxy URL field.
8. Click Save.
9. Repeat steps 4 through 8 for all HTML Rewrite options you want to configure.
10. To reorder the HTML rewrite definitions:
 - a. Click the radio button beside the URL routing definition to reorder.
 - b. Click Move Up or Move Down until the item is in the correct order.
11. Click Save.

Configure an HTTP Adapter for Dashboard

Make a copy of the myFileGateway HTTP adapter. Rename the copy to identify the configuration as a dashboard configuration. Enable Support HTML Rewrite. Specify the SSO configuration and HTTP netmap configurations you created for dashboard. Refer to *Define the HTTP Reverse Proxy Adapter Used for the Single Sign-On Connection* on page 345 for more information.

Modify Sterling Integrator to Support Single Sign-On for Dashboard

Before the Sterling Integrator dashboard is configured to support single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to a file called customer_overrides.properties. This prevents custom settings

from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. Do one of the following:
 - ◆ On a Windows system, in the `install_dir\properties` directory, locate or create the `customer_overrides.properties` file.
 - ◆ On a UNIX or Linux system, in the `install_dir/properties` directory, locate or create the `customer_overrides.properties` file.

2. Open the file in a text editor and ensure that all of the following entries are included and are not commented out. Refer to *Configure Sterling EA to Support Single Sign-On* on page 346 for additional information about specific properties parameters.

```
## neo-ui.properties for dashboard SSO
neo-struts-ui.url.ws.sso=http(s)://<ssp fully qualified host name>:<ssp http adapter
listen port>/ws/
neo-struts-ui.url.dash.sso=http(s)://<ssp fully qualified host name>:<ssp http
adapter listen port>/dashboard/

# Enable sso authentication
security.SSO_AUTHENTICATION_ENABLED=true

# Enable sso authentication on each page
security.SSO_PAGE_AUTHENTICATION_ENABLED=true

## HTTP header containing the SSO user
security.SSO_USER_HEADER=SM_USER

# SSO Provider
security.SSO_AUTHENTICATION_CLASS.1=com.sterlingcommerce.seas.gis.sso.plugin.SeasSso
Provider

## External Page for SSO when Logout (Specify the SSO Server external page for each of
the cases)
##      Example: SSO_FORWARD_URL.MAILBOX.LOGOUT=http://sterlingcommerce.com
##      After SSO User logout from Mailbox, instead of display the Mailbox
Login Screen,
##      display Sterling Commerce Web page.
security.SSO_FORWARD_URL.AFT.LOGOUT=/Signon/logout
security.SSO_FORWARD_URL.MYAFT.LOGOUT=/Signon/logout
security.SSO_FORWARD_URL.MAILBOX.LOGOUT=/Signon/logout
security.SSO_FORWARD_URL.WS.LOGOUT=/Signon/logout
security.SSO_FORWARD_URL.DASHBOARD.LOGOUT=/Signon/logout
## Default handling for LOGOUT if don't know source
security.SSO_FORWARD_URL.LOGOUT=/Signon/logout

## External Page for SSO when Timeout (Specify the SSO Server External page for each
of the case)
security.SSO_FORWARD_URL.AFT.GIS_TIMEOUT=/Signon/timeout
security.SSO_FORWARD_URL.MYAFT.GIS_TIMEOUT=/Signon/timeout
security.SSO_FORWARD_URL.MAILBOX.GIS_TIMEOUT=/Signon/timeout
security.SSO_FORWARD_URL.WS.GIS_TIMEOUT=/Signon/timeout
security.SSO_FORWARD_URL.DASHBOARD.GIS_TIMEOUT=/Signon/timeout
## Default handling for TIMEOUT if don't know source
security.SSO_FORWARD_URL.GIS_TIMEOUT=/Signon/timeout

## External Page for SSO on Validation/Authentication failure (SSO User Validation
Failed - At login or Page Validation)
security.SSO_FORWARD_URL.AFT.VALIDATION_FAILED=/Signon/validationerror
security.SSO_FORWARD_URL.MYAFT.VALIDATION_FAILED=/Signon/validationerror
security.SSO_FORWARD_URL.MAILBOX.VALIDATION_FAILED=/Signon/validationerror
security.SSO_FORWARD_URL.WS.VALIDATION_FAILED=/Signon/validationerror
security.SSO_FORWARD_URL.DASHBOARD.VALIDATION_FAILED=/Signon/validationerror
## Default handling for VALIDATION FAILED if don't know source
security.SSO_FORWARD_URL.VALIDATION_FAILED=/Signon/validationerror
```



```

## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-sso.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-sso.SSO_TIMEOUT=30
seas-sso.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-sso.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-sso.MAX_EA_CONNECTIONS=1

## External Authentication Server
seas-sso.EA_HOST=<EA Host name or ip address>
seas-sso.EA_PORT=<EA Server listen port>
seas-sso.EA_PS_NAME=local or <configured remote ps name if a remote ps is used>
seas-sso.EA_SECURE_CONNECTION=true/false

## The following are used only if EA_SECURE_CONNECTION=true
seas-sso.EA_SYSTEM_CERT=<name of keycert used to establish ssl connection with EA>
seas-sso.EA_TRUSTED_CERT[1]=<name of one of the certificates in the chain of trusted
certs used to validate the cert from EA>

### Settings for the SEAS-Authentication plugin
#

seas-auth.authentication_1.profile=userAuth
seas-auth.authentication_1.ea_pool=SSO_POOL

seas-auth.authentication_2.profile=userAuth
seas-auth.authentication_2.ea_pool=pool1

# The special pool name "SSO_POOL" means to share a pool with the SSO plugin,
configured in seas-sso.properties.

#####
# pool1
#

## Maximum time to wait for making EA connections and receiving responses
seas-auth.pool1.TIMEOUT=30
seas-auth.pool1.TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-auth.pool1.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-auth.pool1.MAX_EA_CONNECTIONS=1

## External Authentication Server
seas-auth.pool1.EA_HOST=<EA Host name or ip address>
seas-auth.pool1.EA_PORT=<EA Server listen port>
seas-auth.pool1.EA_PS_NAME=local or <configured remote ps name if a remote ps is
used>
seas-auth.pool1.EA_SECURE_CONNECTION=true/false

```

```

## The following are used only if EA_SECURE_CONNECTION=true
seas-auth.pool1.EA_SYSTEM_CERT=<name of keycert used to establish ssl connection with
EA>
seas-auth.pool1.EA_TRUSTED_CERT[1]=<name of one of the certificates in the chain of
trusted certs used to validate the cert from EA>

#####
#
# EA Server 1 Authentication Configuration
#
#####
authentication_policy.authentication_1.className=com.sterlingcommerce.seas.gis.sso.p
login.SeasAuthentication
authentication_policy.authentication_1.display_name=<display name for the auth
configuration, this will show up in the user account screens>
authentication_policy.authentication_1.enabled=true/false

#####
#
# EA Server 2 Authentication Configuration
#
#####
authentication_policy.authentication_2.className=com.sterlingcommerce.seas.gis.sso.p
login.SeasAuthentication
authentication_policy.authentication_2.display_name=<display name for the auth
configuration, this will show up in the user account screens>
authentication_policy.authentication_2.enabled=true/false

```

Create User Accounts in Sterling Integrator

User accounts work with permissions to provide security for your organization. These features make it possible to regulate which users have access to each module in Sterling Integrator and what functions each user can perform. To create a user account:

1. From within Sterling Integrator, Select Accounts > User Accounts > Create a new Account.
2. Complete the steps in the wizard. Supply the following information about the user:
 - ◆ Authentication type (external)
 - ◆ User ID
 - ◆ Authentication Host—Select a host that corresponds to the seas_auth.authentication-*n* configuration maintained in the customer_overrides.properties file described in the previous section.
 - ◆ Password (For the default user policy, the password must be six characters or more and contain at least two of the following characters. (number, capital letter, !, @, #, \$, %, ^, &, *))
 - ◆ Confirm Password
 - ◆ Policy (Default User Policy)
 - ◆ SSH Authorized User Key
 - ◆ Session Timeout (in minutes)

- ◆ Accessibility (Dashboard UI)—This adds the user to the Dashboard Users group.
 - ◆ Dashboard Theme (Default)
3. If the user must access FileGateway, select one or more of the following groups to assign the user to, based on their job responsibilities:
- ◆ File Gateway Integration Architects
 - ◆ File Gateway Operators
 - ◆ File Gateway Route Provisioners
 - ◆ File Gateway System Administrators

Note: Do not assign the FileGateway user to the trading partner group. Otherwise, the user will not be able to login to the File Gateway application.

Note: For full SFG functionality, each of these groups must have at least one user. By default, the following users are created during installation: fg_sysadmin, fg_architect, fg_provisioner, and fg_operator. One user can belong to multiple groups.

Note: To create the equivalent of fg_sysadmin, assign all the File Gateway groups listed above and the Sterling Integrator Admin group to the user.

4. Supply the following information for the user:
- ◆ First Name
 - ◆ Last Name
 - ◆ E-mail
 - ◆ Pager
 - ◆ Preferred Language (English, Japanese)
 - ◆ Manager ID
 - ◆ Identity
5. Review and confirm the user to create the new user account.

Verify the SSP Connections

To verify that the engine can receive and initiate communications sessions after configuring the basic single sign-on environment, establish a connection between an HTTP client and the HTTP reverse proxy adapter to ensure that the SSP Login page is displayed. Refer to *Verify the SSP Connections* on page 352 for more information.

If you can view the dashboard home page, you have confirmed that the connections are working. If the Default Landing Page in your SSO configuration is the Welcome page, you will see a Welcome page with links to back-end applications, such as myFileGateway and the dashboard. Click on the

dashboard link to see the dashboard home page and confirm that the connections are working. You can change the Default Landing Page to /dashboard/sso.jsp to skip the Welcome page and go right to the dashboard.

You are ready to add SSL or TLS support to the inbound connection. For more information, refer to *HTTP Reverse Proxy Configuration* on page 167.

Add Single Sign-On Support for Basic Authentication Applications on Sterling Integrator

After you configure single sign-on for myFileGateway, determine if users require access to Sterling Integrator applications that use basic authentication, such as AS2 or WebDay.

Configure Single Sign-On for a Basic Authentication Application

Complete the following procedures to configure SSP for basic single sign-on for basic authentication applications on Sterling Integrator.

Create an SSO Configuration for a Basic Authentication Application on page 370

Create an HTTP Policy to Support a Single-Sign On Connection on page 370

Define the HTTP Netmap for a Basic Authentication Application on page 371

Configure an HTTP Adapter for a Basic Authentication Application on page 371

Create Basic Authentication Application User Accounts in Sterling Integrator on page 371

Verify the SSP Connections on page 371

Create an SSO Configuration for a Basic Authentication Application

To configure SSO for basic authentication application users, make a copy of the myFileGateway SSO configuration. Rename the copy to identify the configuration as a definition for the application, such as AS2. Change the field called Default Landing Page to connect to the application, such as /as2. Refer to *Create an SSO Configuration* on page 344 for more information.

Create an HTTP Policy to Support a Single-Sign On Connection

To create an HTTP policy to support a single sign-on connection to a basic authentication application:

1. Select Basic Authentication in the User Authentication Type field. Enable *Through External Authentication* and enable *SSO token from External Authentication*.
2. Type the definition you defined in EA in the External Authentication Profile field.

For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Define the HTTP Netmap for a Basic Authentication Application

Make a copy of the myFileGateway HTTP netmap. Rename the copy to identify the configuration as a definition for the application, such as AS2. Configure the inbound node definitions for the nodes that need to connect to the application. Configure the outbound node definition to support the connection to the application. Change the values as needed. Refer to *Create an HTTP Netmap to Support a Single Sign-On Connection to myFileGateway* on page 345 for more information.

Usually, applications that use basic authentication, such as AS/2 and WebDav, do not require HTML rewrite.

Configure an HTTP Adapter for a Basic Authentication Application

Make a copy of the myFileGateway HTTP adapter. Rename the copy to identify the configuration as a basic authentication application configuration, such as AS/2 or WebDav. Specify the SSO configuration and HTTP netmap configurations you created for the application. Refer to *Define the HTTP Reverse Proxy Adapter Used for the Single Sign-On Connection* on page 345 for more information.

Create Basic Authentication Application User Accounts in Sterling Integrator

User accounts work with permissions to provide security for your organization. These features make it possible to regulate which users have access to each module in Sterling Integrator and what functions each user can perform. For more information, refer to *Create User Accounts in Sterling Integrator* on page 368.

Verify the SSP Connections

To verify that the engine can receive and initiate communications sessions after configuring the basic single sign-on environment, establish a connection between an HTTP client and the HTTP reverse proxy adapter to ensure that the browser user ID/password prompt is displayed. Refer to *Verify the SSP Connections* on page 352 for more information.

For applications that use basic authentication, the Default Landing Page is the back-end application URL. If the Default Landing Page in your SSO configuration is the Welcome page, you will see a Welcome page with links to back-end applications, such as myFileGateway, AS2, and WebDav. Click on the application link to see the application home page and confirm that the connections are working. You can change the Default Landing Page to the URI of the application, such as /as2, to skip the Welcome page and go right to the application.

You are ready to add SSL or TLS support to the inbound connection. For more information about configuring the HTTP policy, refer to *HTTP Reverse Proxy Configuration* on page 167.

Customize the Logon Portal

SSP provides a self-service Logon Portal that allows SFG users to manage and change their passwords. The Logon Portal is separately licensed and includes verification of the new password, password expiration notification, display of password policy, and welcome and logon screens. You can also configure SSP to use an external logon portal.

To support the Logon Portal, configure the HTTP protocol in SSP for SSO.

This topic describes how to configure the SSP Logon Portal for the HTTP protocol.

Common User Tasks Managed by the Logon Portal

You can configure the Logon Portal to skip the Welcome page. You cannot configure the sequence of any other Logon Portal pages presented to the user. The following Logon Portal workflows are described below:

Workflow When the User Initiates a Password Change on page 372

Workflow When a User Password is Expired or Must Change on page 372

Workflow When the User Provides Invalid Logon Credentials on page 372

Workflow When a User Account is Locked on page 373

Workflow When the User Cannot Change Password on page 373

Workflow When the User Initiates a Password Change

In this scenario, the trading partner decides to change his password.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials. If the credentials are valid, a Welcome page is displayed. The user can change his password or continue to the HTTP application. If the user selects Change Password, he can view the password policy or change his password.

To change a password, the user must follow the restrictions defined in the password policy. However, the user will not be locked out of SSP if he does not define a valid new password.

Note: The setting to allow the trading partner to change his password is in the EA group profile. This profile is specified in the HTTP policy configuration.

Workflow When a User Password is Expired or Must Change

In this scenario, the trading partner's password has expired or must be changed.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials. If the credentials are valid, a Change Password page is displayed and a user message is presented indicating that the password is expired or must be changed. The user can change his password or view the password policy. If the user successfully changes his password, a Welcome page is displayed. If the user fails to successfully change his password, a Change Password page is displayed with an error message.

Note: The setting to allow the trading partner to change his password is in the EA group profile. This profile is specified in the HTTP policy configuration.

Workflow When the User Provides Invalid Logon Credentials

In this scenario, the trading partner enters an invalid user ID or password.

When a user connects to SFG, SSP presents a Login page. The user provides invalid user credentials. A Login page is displayed with a customizable error message.

Workflow When a User Account is Locked

In this scenario, the trading partner enters a valid user ID and password, but the account is locked.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials for a locked account. A Login page is displayed with a customizable error message.

Workflow When the User Cannot Change Password

In this scenario, the trading partner enters a valid user ID and password, but the user is not allowed to change the password.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials for an account that is not allowed to change the password. A Login page is displayed with a customizable error message.

Note: The setting to allow the trading partner to change his password is in the EA group profile. This profile is specified in the HTTP policy configuration.

Configuration Considerations

Before you complete the Logon Portal configuration, be aware of the following considerations:

- SSP SSO must be properly configured before you can use the change password functionality of the Logon Portal.

- Configure Active Directory or LDAP to allow the trading partner to change his password.

- Comments are included in the default Logon Portal .html pages to simplify editing. Remove all of these comments after you edit the pages to minimize security risks.

Organization of Logon Portal Customization Scenarios

When you first configure SSP for HTTP, you use the default Login page, Welcome page, Change Password Page, Logout page, and Password Policy Page.

This document provides instructions on how to customize an SSP Logon Portal.

- Customize the Login Page* on page 374

- Customize the Welcome Page* on page 375

- Configure SSP to Skip the Welcome Page* on page 376

- Customize the Change Password Page* on page 377

- Customize the Logout Page* on page 378

- Customize Password Policy Page* on page 380

- Customize User Messages* on page 380

- Configure the Forgot Your User ID or Password Page* on page 380

- Configure SSP to Use External Logon Portal* on page 381

Customize the Login Page

The default Login page is a simple page with no logo information and prompts the user to provide a user ID and password. The default page also contains a link if the user forgets his user ID or password.

You can customize the Login page to define how you want the page to look and what information to include on the page. You customize this page by modifying the labels or replacing the entire page.

Before you modify the Login page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Login Page	Custom page to display for the SSP single sign-on login.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Login page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Login page:
 - ◆ Login Page
 - ◆ Login Directory Name

- ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
 6. If you want to change the text or graphics on the Login page, open the `\install_dir\signon` directory and modify the login .html file as required.

Note: If you modify the login .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the login .html file, create a copy of the `\install_dir\Signon` directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Welcome Page

The default Welcome page is a simple page with no logo information that provides links to the Logout page and the Change Password page. The default page does not include links to back-end applications.

You can customize the Welcome page to define how you want the page to look and what information to include on the page. You customize this page by modifying the labels or replacing the entire page.

Before you modify the Welcome page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Welcome Page	Custom page to display for the SSP single sign-on welcome.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Welcome page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Welcome page:
 - ◆ Welcome Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. If you want to change the text or graphics on the Welcome page, open the `\install_dir\signon` directory and modify the `welcome.html` file as required.

Add the URLs for back-end applications in the `welcome.html` as required.

Note: If you modify the `welcome.html` file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the `welcome.html` file, create a copy of the `\install_dir\Signon` directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Configure SSP to Skip the Welcome Page

You can configure SSP so that a user is directed to the back-end application instead of the Welcome page after logging in. You can configure this behavior by modifying the SSO configuration.

To configure SSP to skip the Welcome page:

1. Click Advanced from the menu bar.
2. Do one of the following:

- ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Advanced tab, type the URL for the back-end application in the Default Application URL field. You can use the relative URL, such as /myfilegateway, or full URL.
 4. Click Save.

Customize the Change Password Page

The default Change Password page is a simple page with no logo information that prompts the user to provide his user ID, existing password, and new password. The default page provides a link to the Password Policy page.

You can customize the Change Password page to define how you want the page to look and what information to include on the page. You customize this page by modifying the labels or replacing the entire page.

Before you modify the Change Password page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Change Password Page	Custom page to display for the SSP single sign-on Change Password page.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Change Password page:

1. Click Advanced from the menu bar.
2. Do one of the following:

- ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
 4. Change one or more of the following fields to customize the Change Password page:
 - ◆ Change Password Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
 5. Click Save.
 6. If you want to change the text or graphics on the Change Password page, open the `\install_dir\signon` directory and modify the change password .html file as required.

Note: If you modify the change password .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the change password .html file, create a copy of the `\install_dir\Signon` directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Logout Page

The default Logout page is a simple page with no logo information. You can configure SSP to use the Login page in place of the Logout page.

You can customize the Logout page to define how you want the page to look and what information to include on the page. You customize this page by modifying the labels or replacing the entire page.

Before you modify the Logout page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	

Configuration Manager Field	Feature	Value
SSP Internal Portal		
◆ Logout Page	Custom page to display for the SSP single sign-on Logout page. If you want to use the Login page as the Logout page, specify the Login page here.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Logout page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Logout page:
 - ◆ Logout Password Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. If you want to change the text or graphics on the Logout page, open the `\install_dir\signon` directory and modify the logout .html file as required.

Note: If you modify the logout .html file, do not modify the following lines:

```
var ssoMsgText="{ssoMsgText}";  
var ssoMsgTitle="{ssoMsgTitle}";  
var ssoMsgType="{ssoMsgType}";  
var ssoMsgOnly="{ssoMsgOnly}";
```

7. If you modify the logout .html file, create a copy of the `\install_dir\Signon` directory.
Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize Password Policy Page

The Password Policy page is a simple page with no logo information that displays the password policy.

SSP obtains the password policy dynamically from Active Directory or Tivoli via EA. If you update the password policy, the Password Policy page displays the new password policy.

Customize User Messages

You can customize user messages that display on the Logon Portal pages. You customize these messages by modifying the `messageBundle.properties` file, located in the `Signon/resources` directory.

By default, messages are associated with specific events. For example, if a logon attempt fails because the user password has expired, the logon page displays a message alerting the user that the password has expired. For security reasons, you might use the same general error message for all logon failures.

To customize user messages:

1. From the `SSP install_dir\Signon\resources` directory, open the `messageBundle.properties` file in a text editor.
2. Modify the message text for all user messages you want to customize.
3. Save the `messageBundle.properties` file.
4. Restart SSP.

Configure the Forgot Your User ID or Password Page

You can configure the Forgot Your User ID or Password page to display a customized user message. You customize this user message by editing the Login page .html file.

To customize the Forgot Your User ID or Password page, open the `\install_dir\signon` directory and modify the `login.html` file as required.

Note: If you modify the login .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

If you modify the logout .html file, create a copy of the `\install_dir\Signon` directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Configure SSP to Use External Logon Portal

You can configure SSP to use an external logon portal.

Before you configure an external logon portal, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
External Portal		
◆ External Application Login URL	URL of external login portal.	

To configure SSP to use the external logon portal:

1. Click **Advanced** from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click **Actions>New SSO Configuration**.
 - b. Type an SSO configuration name in the **Name** field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click **SSO Configurations**.
 - b. Click the configuration to modify.
3. On the **Logon Portal** tab, select **External Portal**.
4. Type the URL of the external login portal in the **External Application Login URL** field.
5. Click **Save**.

Configure a Single Sign-on Connection to a Connect:Direct Server

Sterling Secure Proxy (SSP) can be used as a proxy with Sterling Integrator and Sterling File Gateway (SFG) and supports a single sign-on connection for Connect:Direct connections. Single sign-on (SSO) bypasses the normal user authentication process in Sterling Integrator and instead trusts that SSP has authenticated the user.

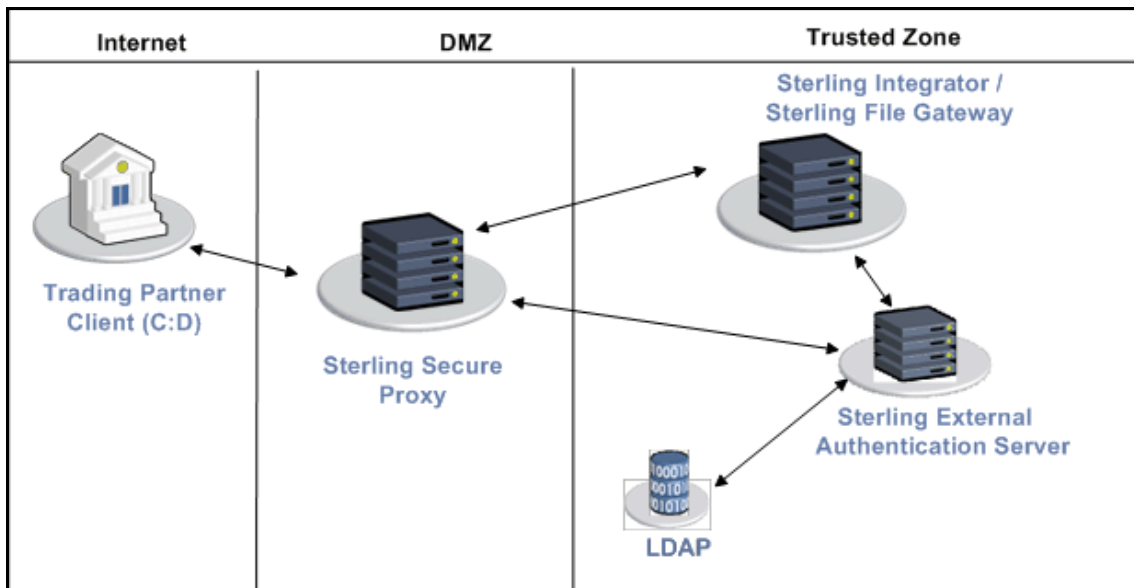
To support single sign-on, configure EA to generate SSO tokens. Configuring SSO allows a trading partner to log on and use the same login session to connect to SSP and Sterling Integrator. By default, EA uses OpenSAML to create and manage SSO tokens. However, you can customize your environment to use a third-party application to generate tokens.

This topic describes how to configure the Connect:Direct protocol in SSP between the trading partner and SSP and between SSP and Sterling Integrator to enable authentication through EA. It describes how to configure EA to issue tokens to authenticate the connection between SSP and Sterling Integrator. It also describes how to configure a self-service Change Password Portal for external trading partners.

Flow of Data for Single Sign-On Configuration Between Sterling Integrator and SSP

After you set up the basic single sign-on configuration, trading partners can communicate in a secure environment that provides authentication. The trading partner first connects to SSP which then connects to Sterling Integrator on behalf of the trading partner.

Following is an illustration of the flow of data:



Following are the steps that occur during a single sign-on session between a trading partner, SSP, and Sterling Integrator when EA is used to generate and manage tokens:

1. The trading partner requests a connection to Sterling Integrator.
2. SSP receives the request, and the SSL handshake between SSP and the trading partner begins. If SSL authentication is configured, the proxy submits its certificate to the trading partner. If client authentication is configured, the trading partner then submits its certificate to SSP for authentication. You can optionally configure SSP to enforce client authentication and send the certificate to EA for validation.
3. SSP sends an authentication request to the trading partner, who provides his user ID and password.
4. SSP sends the user ID and password to EA and then validates it against information stored in LDAP.
5. If the credentials are valid, EA creates an OpenSAML v2 token and returns the token to SSP.
6. SSP connects to Sterling Integrator and performs an SSL handshake. SSP then sends the request with the token from EA to Sterling Integrator.
7. Sterling Integrator validates the token against EA and begins normal operation.

Configuration Considerations

Before you complete the single sign-on configuration, be aware of the following considerations:

Only the HTTP, Connect:Direct, FTP, and SFTP protocols support single sign-on connections.

The SSP Change Password Portal requires an HTTP adapter, which is an optional, licensed component of SSP, and a license for the Change Password Portal. Refer to *Configure Change Password Portal* on page 396 for instructions to configure this feature.

Organization of Single Sign-On Scenarios

The scenarios describe how to configure single sign-on between SSP and trading partners and between SSP and Sterling Integrator.

Configure the Basic Scenario to Enable a Connection to Sterling Integrator

Configure the basic scenario to allow you to quickly become operational using single sign-on to connect to a Connect:Direct Server Adapter in Sterling Integrator. After you complete this scenario, test the connection to ensure that you have correctly configured it. You then have a basic configuration and can begin operation.

Configure Advanced Features

After you configure the basic SSO setup, determine if your environment requires an advanced feature. Following are the advanced features:

Use a third-party application to configure tokens. The basic scenario uses EA to configure and manage tokens. To use a third-party application to configure tokens, you complete additional setup procedures. Refer to *Allow a Third-Party Provider to Create Tokens* on page 393.

Customize the OpenSAML v2 tokens—You use the default token generation definition when you configure the basic single sign-on definition. To customize the token definition, you can modify the named identity provider, the token signing key, or how long a token can be used before it expires. Refer to *Customize Token Definitions Created by EA* on page 394.

Configure Sterling Integrator or SFG with additional pools—You use additional pools to support more than one EA server. Refer to *Configure Sterling Integrator or SFG to use multiple EA servers* on page 395.

Configure Optional Features

SSP provides optional security features and you can configure them as required for your environment.

EA provides the ability to configure multifactor authentication. In addition to configuring client authentication in SSP, EA can also authenticate the IP address, certificate, password, and/or group access. Refer to the Sterling External Authentication Server documentation library for instructions.

SSP provides a Change Password Portal that allows trading partners to manage their own passwords. This feature requires the HTTP adapter, which is an optional, licensed component of SSP. Refer to *Configure Change Password Portal* on page 396 for instructions on how to configure this feature.

Worksheets

Before you complete each procedure, gather the information you need to configure on the worksheet provided. For each worksheet:

- Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

- Accept default values for fields not listed.

- Note the Configuration Manager field where you will specify the value.

Basic Single Sign-On Scenario

Complete the following tasks to define a Connect:Direct configuration between a trading partner and SSP and between SSP and Sterling Integrator to support a single sign-on connection to a Connect:Direct Server Adapter:

- Configure SSP to support basic single sign-on.

- Use the default single sign-on configuration in EA to manage OpenSAML v2 tokens.

- Prepare Sterling Integrator to support the single sign-on option.

- Validate the connections between the trading partner, SSP, and Sterling Integrator.

Configure SSP for Basic Single Sign-On

Complete the following procedures to configure SSP for basic single sign-on:

- Create a Connect:Direct policy to support a single sign-on connection to Sterling Integrator.

- Define a Connect:Direct netmap to identify inbound and outbound connections.

- Define a Connect:Direct adapter.

Create a Connect:Direct Policy to Support a Single-Sign On Connection

To create a Connect:Direct policy to support a single sign-on connection to SFG:

1. Enable Through External Authentication.
2. Type the definition you defined in EA in the External Authentication Profile field.

3. From the Internal User ID field, enable SSO token from External Authentication.

For more information about configuring the Connect:Direct policy, refer to *Connect:Direct Proxy Configuration*.

Create a Connect:Direct Netmap to Support a Single Sign-On Connection to Sterling Integrator

To create a Connect:Direct netmap to support a single sign-on connection to Sterling Integrator, configure the Connect:Direct node information for your external trading partners. The inbound node must use the preceding Connect:Direct policy to support SSO.

For more information about configuring the Connect:Direct netmap, refer to *Connect:Direct Proxy Configuration*.

Define the Connect:Direct Adapter Used for the Single Sign-On Connection

To create an Connect:Direct adapter to support a single sign-on connection to Sterling Integrator:

1. Specify the netmap you created for the single sign-on connection to Sterling Integrator for the Netmap field.
2. Specify the name of the outbound node for the SNODE Netmap Entry field.
3. Specify your EA server in the External Authentication Server field.

For more information about configuring the Connect:Direct adapter, refer to *Connect:Direct Proxy Configuration*.

Configure Sterling EA to Support Single Sign-On

To allow an SSO connection between a trading partner and SSP to route traffic to Sterling Integrator, you configure OpenSAML v2.0 tokens in EA. You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines which options are enabled. Refer to *Sterling External Authentication Server documentation library* for instructions on configuring an EA definition.

The EA server generates and manages tokens. A default configuration called SEAS-SAML is enabled when you install EA. If you use the default configuration, EA is the identity provider, token signing keys are automatically generated, and the token expires after 15 minutes. Use the default configuration when you configure basic single sign-on.

To customize EA for single sign-on, refer to *Customize Token Definitions Created by EA* on page 394.

Prepare Sterling Integrator to Support Single Sign-On

Before you enable single sign-on between a trading partner and Sterling Integrator, when using SSP, you modify the Sterling Integrator installation. The files required to enable SSO are installed with EA.

Prepare Sterling Integrator to Support Single Sign-On on UNIX or Linux

To prepare Sterling Integrator to support SSO on UNIX or Linux:

1. From the EA server, copy the files and subdirectories from the *EA_install_dir/lib/sterling/sfg-ss-plugin* directory to a location that is accessible to the Sterling Integrator server, where *EA_install_dir* is the location of the EA installation.

Note: If you use FTP to copy the files to the Sterling Integrator server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the Sterling Integrator server, move to the *Sterling_Integrator_install_dir/properties* directory, where *Sterling_Integrator_install_dir* is the Sterling Integrator installation directory.
3. Type the following commands to copy the SSO properties files to the Sterling Integrator server, where *base_dir* is the location where you copied the files in step 1:

```
cp base_dir/sfg-ss-plugin/properties/security.properties_seas-ss-ext.in .
cp base_dir/sfg-ss-plugin/properties/authentication_policy.properties_seas-auth_ext.in .
cp base_dir/sfg-ss-plugin/properties/servers.properties_seas-ss-ext .
cp base_dir/sfg-ss-plugin/properties/servers.properties_seas-auth_ext .
```

4. Stop Sterling Integrator if it is running.
5. In the *server.properties_seas-ss-ext* file, uncomment the following line and replace *<SI_install>* with the actual installation path for Sterling Integrator:

```
# seas-ss=<SI_install>/properties/seas-ss/1.0/seas-ss.properties
```

6. In the *server.properties_seas-auth_ext* file, uncomment the following line and replace *<SI_install>* with the actual installation path for Sterling Integrator:

```
# seas-auth=<SI_install>/properties/seas-auth/1.0/seas-auth.properties
```

7. From the *Sterling_Integrator_install_dir/bin* directory, type the following commands:

```
./install3rdParty.sh seas-ss 1.0 -j /sfg-ss-plugin/seas-ss.jar
./install3rdParty.sh seas-ss 1.0 -p /sfg-ss-plugin/properties/seas-ss.properties
./install3rdParty.sh seas-auth 1.0 -p /sfg-ss-plugin/properties/seas-auth.properties
```

8. From the *Sterling_Integrator_install_dir/jar/seas-ss/1.0* directory, create a subdirectory named *private*.
9. Move to the */private* directory.
10. Type the following command to copy the .jar files to the */private* directory on the Sterling Integrator server:

```
cp /sfg-ss-plugin/private/*.jar .
```

Modify Sterling Integrator to Support Single Sign-On on UNIX or Linux

Before Sterling Integrator supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to `customer_overrides.properties` to prevent custom settings from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. In the `install_dir/properties` directory, locate or create the `customer_overrides.properties` file.
2. Open the file in a text editor and add the properties that you want to override.

Add the following parameters to configure the connection to EA:

- ◆ `seas-sso.EA_HOST=IP address or host name of EA server`
- ◆ `seas-sso.EA_PORT=listen port of EA server`

Specify the appropriate secure or clear listen port from the EA server configuration.

- ◆ `seas-sso.EA_PS_NAME=perimeter server used to connect to EA`
- ◆ `seas-sso.EA_SECURE_CONNECTION=enables a secure EA`

true sets connections to EA as secure and *false* sets the connection as clear. If this parameter is true, you must also define the `EA_SYSTEM_CERT` and `EA_TRUSTED_CERT[1]`.

- ◆ `seas-sso.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.`
- ◆ `seas-sso.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.`

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the `seas-sso.EA_TRUSTED_CERT(#)` parameter. For example, for the first certificate, configure the parameter, `seas-sso.EA_TRUSTED_CERT[1]`; for the second certificate, define `seas-sso.EA_TRUSTED_CERT[2]`, until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart Sterling Integrator to use the new values.

Prepare Sterling Integrator to Support Single Sign-On on Windows

To prepare Sterling Integrator to support SSO on Windows:

1. From the EA server, copy the files and subdirectories from the *EA_install_dir\lib\sterling\sfg-ss0-plugin* directory to a location that is accessible by the Sterling Integrator server.

Note: If you use FTP to copy the files to the Sterling Integrator server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the Sterling Integrator server, move to the *Sterling Integrator_install_dir\properties* directory.
3. Type the following command to copy the SSO security.properties file to the Sterling Integrator server, where *base_dir* is the location where you copied the files in step 1:

```
copy      \sfg-ss0-plugin\properties\security.properties_seas-ss0_ext.in .
copy      \sfg-ss0-plugin\properties\authentication_policy.properties_seas-auth_ext.in .
copy      \sfg-ss0-plugin\properties\server.properties_seas-ss0_ext .
copy      \sfg-ss0-plugin\properties\server.properties_seas-auth_ext .
```

4. Stop Sterling Integrator if it is running.
5. In the server.properties_seas-ss0_ext file, uncomment the following line and replace <SI_install> with the actual installation path for Sterling Integrator:

```
# seas-ss0=<SI_install>\properties\seas-ss0\1.0\seas-ss0.properties
```

6. In the `server.properties_seas-auth_ext` file, uncomment the following line and replace `<SI_install>` with the actual installation path for Sterling Integrator:

```
# seas-auth=<SI_install>\properties\seas-auth\1.0\seas-auth.properties
```

7. From the `Sterling_Integrator_install_dir\bin` directory, type the following commands:

```
install3rdParty.cmd seas-sso 1.0 -j          \sfg-sso-plugin\seas-sso.jar
install3rdParty.cmd seas-sso 1.0 -p          \sfg-sso-plugin\properties\seas-sso.properties
install3rdParty.cmd seas-auth 1.0 -p         \sfg-sso-plugin\properties\seas-auth.properties
```

8. From the `Sterling_Integrator_install_dir\jar\seas-sso\1.0` directory, create a subdirectory named `private`.
9. Go to the `\private` directory.
10. Type the following command to copy the `.jar` files to the Sterling Integrator server:

```
copy          \sfg-sso-plugin\private\*.jar .
```

Modify Sterling Integrator to Support Single Sign-On on Windows

Before Sterling Integrator supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to `customer_overrides.properties` to prevent custom settings from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. In the `install_dir\properties` directory, locate or create the `customer_overrides.properties` file.
2. Open the file in a text editor and add the properties that you want to override.

Add the following parameters to configure the connection to EA:

- ◆ `seas-sso.EA_HOST=IP address or host name of EA server`
- ◆ `seas-sso.EA_PORT=listen port of EA server`

Specify the appropriate secure or clear listen port from the EA server configuration.

- ◆ `seas-sso.EA_PS_NAME=perimeter server used to connect to EA`
- ◆ `seas-sso.EA_SECURE_CONNECTION=enables a secure EA`

`true` sets connections to EA as secure and `false` sets the connection as clear.

If this parameter is true, you must also define the `EA_SYSTEM_CERT` and `EA_TRUSTED_CERT[1]`.

- ◆ `seas-sso.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.`

- ◆ seas-ss0.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the seas-ss0.EA_TRUSTED_CERT(#) parameter. For example, for the first certificate, configure the parameter, seas-ss0.EA_TRUSTED_CERT[1]; for the second certificate, define seas-ss0.EA_TRUSTED_CERT[2], until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart Sterling Integrator to use the new values.

Change Sterling Integrator User Accounts for Single Sign-On

After you install and configure the SSO plug-in and restart Sterling Integrator, the Sterling Integrator User Accounts page presents additional choices for Authentication Type. To enable SSO for a user account, select an authentication method of External and then select the appropriate EA server for Authentication Host. The default is SEAS Authentication. Additional choices are available only if you define other EA Connection pools in addition to the default SSO_POOL.

Verify That Sterling Integrator is Configured for Single Sign-On

Before you configure additional functions, make sure that Sterling Integrator is ready for use in a single sign-on environment. To verify the configuration, start Sterling Integrator.

View the authentication.log and security.log to make sure the Sterling Integrator files are updated. If the update was successful, log files display the success messages.

Authentication.log file displays the following messages:

```
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SI is configured to support
single sign-on
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SSO Property :
SSO_AUTHENTICATION_CLASS.1 = Class name :
com.sterlingcommerce.seas.gis.sso.plugin.SeasSsoProvider
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication - A new SSO
Authentication Policy has been installed.
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication on Page:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager Number of SSO Authentication
Plug-In:1
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO configuration
policy...SSOAuthenticationPolicy isComplete=true isEnabled=true
httpUserIdHeader=SM_USER
ALL 000000000000 GLOBAL_SCOPE SecurityManager initialization complete.
```

Security.log displays the following message:

```
ALL 000000000000 GLOBAL_SCOPE SEAS-SSO: Plug-in initialized
```

Verify the SSP Connections

After you configure the basic single sign-on environment, to verify that the engine can receive and initiate communications sessions, establish a connection between a Connect:Direct server and the Connect:Direct Proxy adapter.

Note: Configuration files must be available on the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

- Establish a Connect:Direct session initiated by a trading partner using a Connect:Direct server
- Initiate an outbound session to an Sterling Integrator Connect:Direct Server Adapter on behalf of the Connect:Direct server connection

To verify the communications sessions:

1. Make sure the engine is running.
2. Submit a Connect:Direct Process where the SNODE is the Connect:Direct Proxy adapter. The SNODE ID and password must match the user ID and password stored in LDAP and the user ID must be defined in Sterling Integrator as an external user with the correct Authentication Host.
3. View the Connect:Direct server statistics to verify the Connect:Direct session.

Advanced Features of the Single Sign-On Configuration

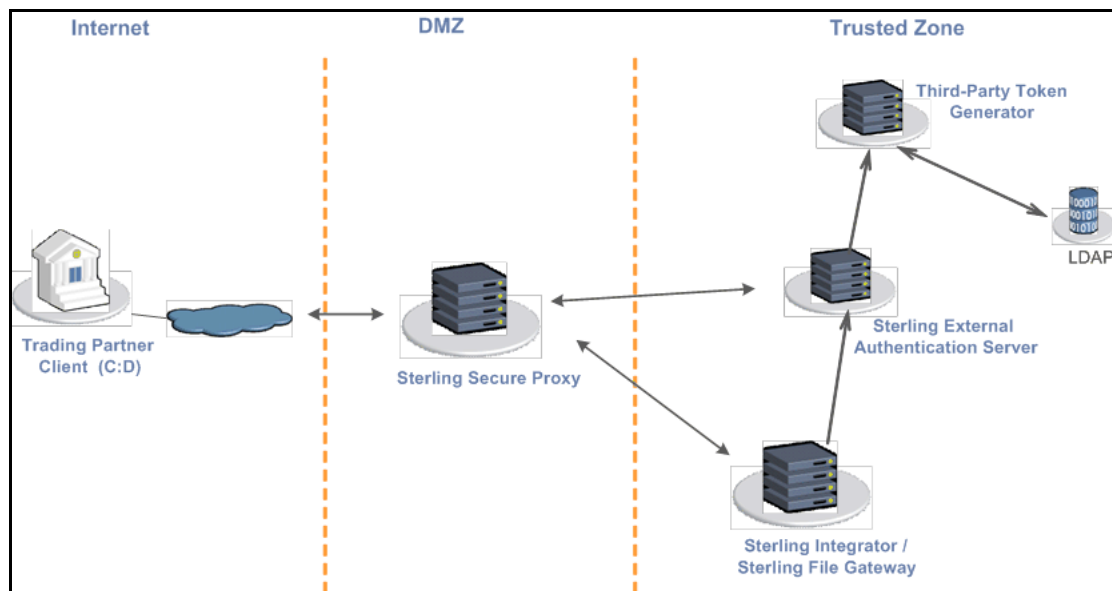
This section provides instructions on the additional features you can configure by modifying the basic single sign-on scenario. Variations include:

- Allow a third-party provider to create tokens
- Customize token definitions created by EA
- Configure Sterling Integrator or SFG to use multiple EA servers
- Configure Change Password Portal

Allow a Third-Party Provider to Create Tokens

You used the default OpenSAML token generation definition when you configured the basic single sign-on definition. The default configuration uses EA to manage tokens. To use a third-party application for token generation, you must modify the SSO Token setup in EA.

The following diagram illustrates the flow using a third-party application for token generation.



Configure EA to Enable a Third-Party Provider to Create Tokens

Before you configure EA to enable a third-party application to create tokens, gather the following information:

Configuration Manager Field	Feature	Value
Token Manager	The application that creates the tokens.	Custom
Class Name	Name of the Java class that implements the Token Manager interface.	

Configuration Manager Field	Feature	Value
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To configure EA and enable a third-party application to generate tokens:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. To configure a token manager other than EA, select Custom from the Token Manager field.
5. Type the class name in the Class name field.
6. To change how long a token can be used before it expires, type a new value in the Token Expiration Period field.
7. Click OK.

Customize Token Definitions Created by EA

You used the default token definition when you configured the basic single sign-on definition. To customize the token definition, complete the following procedure. You can modify the named identity provider, the token signing key, or how long a token can be used before it expires.

Before you customize token definitions, gather the following information:

Configuration Manager Field	Feature	Value
Named Identity Provider	The prefix appended to generated tokens to identify the provider. Note: If you change the identity provider name, any outstanding tokens are invalid.	
Token Signing Key	Alias of the key certificate used to sign the token.	
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To customize the token configuration in EA:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. Customize one or more of the following definitions:
 - ◆ Named Identity Provider
 - ◆ Token Signing Key

- ◆ Token Expiration Period

5. Click OK.

Configure Sterling Integrator or SFG to use multiple EA servers

If you are implementing single sign-on for HTTP with Basic Authentication, or for protocols other than HTTP, and you need to support additional EA servers, add the following parameters for each additional EA pool configuration to the `customer_overrides.properties` file located in the `install_dir\properties` directory.

Note: The SFG myFileGateway and FileGateway applications always use the default SSO_POOL EA connection to validate SSO tokens, regardless of which Authentication Host is selected for the user. Additional EA connection pools may only be used for HTTP Basic Auth applications, FTP, SFTP, and Connect:Direct.

```
authentication_policy.authentication_n.className=
com.sterlingcommerce.seas.gis.sso.plugin.SeasAuthentication
authentication_policy.authentication_n.display_name =
name to be used on the Sterling Integrator/SFG user administration UI. Use something
different than the default SEAS Authentication, which is used by the default SSO_POOL. This
is the Authentication Host name that is selected when you configure external User Accounts to
use this pool.
authentication_policy.authentication_n.enabled=true
seas-auth.authentication_n.profile = userAuth
seas-auth.authentication_n.ea_pool=unique name for your pool other than the default
SSO_POOL, which shares the EA connection pool with the SFG SSO configuration
```

Note: Change the "n" in the above example to a number greater than 1 to avoid overwriting the default SSO_POOL, which is shared with the FileGateway/myFileGateway SSO configuration. Also, make sure you avoid using a number already in use for LDAP authentication. Define a unique number for each entry.

To use another connection pool instead of the default SSO_POOL, configure the EA connection of the pool with the following parameters, where *pool* is the pool name defined in the preceding section:

```
seas-auth.pool.EA_HOST=IP address or host name of EA server
seas-auth.pool.EA_PORT=listen port of EA server
seas-auth.pool.EA_PS_NAME=perimeter server used to connect to EA
seas-auth.pool.EA_SECURE_CONNECTION=enables a secure EA
true sets connections to EA as secure and false sets the connection as clear. If this parameter is
true, you must also define the EA_SYSTEM_CERT and EA_TRUSTED_CERT[1].
seas-auth.pool.EA_SYSTEM_CERT=name of the system certificate in the system certificate
store, if the connection is secure
```

`seas-auth.pool.EA_TRUSTED_CERT[1]`=name of the trusted certificate used by EA for secure connections

`seas-auth.pool.TIMEOUT`=maximum time to wait for making EA connections and receiving responses

`seas-auth.pool.TIMEOUT_UNITS`=unit of time to use, minutes or seconds, for `seas-auth.pool.TIMEOUT` parameter

`seas-auth.pool.PERSISTENT_EA_CONNECTIONS`=whether to keep persistent connections to EA

true sets connections to EA as persistent and *false* sets the connections as not persistent.

`seas-auth.pool.MAX_EA_CONNECTIONS`=maximum number of EA connections

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ssو.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ssو.SSO_TIMEOUT=30
seas-ssو.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ssو.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ssو.MAX_EA_CONNECTIONS=1
```

Configure Change Password Portal

SSP provides a customizable self-service Change Password Portal that allows FTP, SFTP, and Connect:Direct users to manage and change their passwords from a web browser. The Change Password Portal is an optional, licensed component of SSP. Password management capabilities of the Change Password Portal include verification of the new password, password expiration notification, password will expire notification, display of password policy, and customizable screens.

To support the Change Password Portal, configure SSP SSO for the HTTP protocol.

This topic describes how to configure and customize the SSP Change Password Portal.

Change Password Portal Page Flow

The workflow of the Change Password Portal is not configurable. In the following scenario, the trading partner decides to change his password.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials. If the credentials are valid, a Change Password page is displayed. The user can view the password policy or change his password. If the user successfully changes his password, SSP presents a successful password change message. If the user fails to successfully change his password, a Change Password page is displayed with an error message.

To change a password, the user must follow the restrictions defined in the password policy. However, the user will not be locked out of SSP if he does not define a valid new password.

Note: The setting to allow the trading partner to change his password is in the EA group profile. This profile is specified in the protocol policy configuration.

Configuration Considerations

Before you complete the Change Password Portal configuration, be aware of the following considerations:

SSP SSO must be properly configured before you can use the change password functionality of the Change Password Portal.

Configure Active Directory or LDAP to allow the trading partner to change his password.

Comments are included in the default Change Password Portal .html pages to simplify editing. Remove all of these comments after you edit the pages to minimize security risks.

Configuration Overview

Complete the following procedures to configure an SSP Change Password Portal for the Connect:Direct protocol:

Create an SSO Configuration for Change Password Portal on page 397

Define a Policy for Change Password Portal on page 398

Define a Netmap for Change Password Portal on page 398

Define an HTTP adapter for Change Password Portal on page 398

Create an SSO Configuration for Change Password Portal

Before you create an SSO configuration, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSO configuration file	
Fully Qualified Host name the Trading Partner connects to	Fully qualified host name of the Connect:Direct server	
Login Directory Name	Unique directory that contains Change Password Portal .html files	

To define an SSO configuration:

1. Click Advanced from the menu bar.
2. Click Actions>New SSO Configuration.
3. On the Basic tab, type a configuration name in the Name field.
4. Type the host name in the Fully Qualified Host name the Trading Partner connects to field.
5. On the Advanced tab, select Change Password Page in the Default Landing Page field.
6. Uncheck SSO Cookie Secure Flag only if the inbound connection is unsecure (HTTP, not HTTPS).
7. On the Logon Portal tab, define unique Login Directory Name, Login Page and Change Password Page if necessary or accept the defaults to use those included with SSP.
8. Click Save.

Define a Policy for Change Password Portal

Create a new HTTP policy or use an existing one.

Define a Netmap for Change Password Portal

Create a new HTTP netmap or use an existing one.

Define an HTTP adapter for Change Password Portal

Create a new HTTP adapter to connect to the Change Password Portal.

Note: The HTTP adapter is an optional component of SSP.

Before you create an HTTP adapter configuration, gather the following information:

Configuration Manager Field	Feature	Value
Adapter Name	Name to assign to the HTTP adapter configuration file	
Listen Port	Port for the Change Password Portal	
Netmap	Netmap you create for Change Password Portal	
SSO Configuration	SSO configuration you create for Change Password Portal	

To define a Change Password Portal HTTP adapter:

1. Click Configuration from the menu bar.
2. Click Actions>New Adapter>HTTP Reverse Proxy.
3. Type a configuration name in the Adapter Name field.

4. In the Listen Port field, enter a value for the listening port of the HTTP adapter.
This is the Change Password Portal listening port.
5. Select the Netmap for the Change Password Portal from the Netmap list.
6. Select noRouting from the Routing Type list.
This turns the HTTP adapter into a Change Password Portal and prevents the adapter from being used to connect to any outbound node.
7. Select the SSO configuration from the SSO Configuration list.
8. Complete the HTTP adapter configuration.

Organization of Change Password Portal Customization Scenarios

When you first configure SSP for Connect:Direct, you use the default Login page, Change Password page, Logout page, and Password Policy page.

This document provides instructions on how to customize SSP Change Password Portal pages.

Customize the Login Page on page 399

Customize the Change Password Page on page 401

Customize the Logout Page on page 402

Customize Password Policy Page on page 404

Customize User Messages on page 404

Configure the Forgot Your User ID or Password Page on page 404

Configure SSP to Use External Logon Portal on page 404

Customize the Login Page

The default Login page is a simple page with no logo information and prompts the user to provide a user ID and password. This default page contains a link that can be used if the user forgets his user ID or password.

You can customize a Login page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Login page, gather the following information:

Configuration Manager Field	Feature	Value
Name		Name to assign to the SSP configuration
SSP Internal Portal		
◆ Login Page		Custom page to display for the SSP single sign-on Login page.
◆ Login Directory Name		Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.

Configuration Manager Field	Feature	Value
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Login page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Change Password Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Login page:
 - ◆ Login Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. To change the text or graphics on the Login page, open the `\install_dir\signon` directory and modify the `login.html` file as required.

Note: If you modify the `login.html` file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the `login.html` file, create a copy of the `\install_dir\Signon` directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Change Password Page

The default Change Password page is a simple page with no logo information that prompts the user to provide his user ID, existing password, and new password. The default page provides a link to the Password Policy page.

You can customize a Change Password page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Change Password page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Change Password Page	Custom page to display for the SSP single sign-on Change Password page.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Change Password page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Change Password Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Change Password page:
 - ◆ Change Password Page
 - ◆ Login Directory Name

- ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
 6. If you want to change the text or graphics on the Change Password page, open the `\install_dir\signon` directory and modify the change password .html file as required.

Note: If you modify the change password .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the change password .html file, create a copy of the `\install_dir\Signon` directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Logout Page

The default Logout page is a simple page with no logo information. You can configure SSP to use the Login page in place of the Logout page.

You can customize a Logout page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Logout page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Logout Page	Custom page to display for the SSP single sign-on Logout page. If you want to use the Login page as the Logout page, specify the Login page here.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	

Configuration Manager Field	Feature	Value
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Logout page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Logout page:
 - ◆ Logout Password Page
Set the value to the login .html file name if you want to use the Login page in place of the Logout page.
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. If you want to change the text or graphics on the Logout page, open the \install_dir\signon directory and modify the logout .html file as required.

Note: If you modify the logout .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the logout .html file, create a copy of the \install_dir\Signon directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize Password Policy Page

The Password Policy page is a simple page with no logo information that displays the Password Policy.

SSP obtains the password policy dynamically from Active Directory or Tivoli via EA. If you update the password policy, the Password Policy page displays the new password policy.

Customize User Messages

You can customize user messages that display on the Change Password Portal pages. Customize these messages by modifying the `messageBundle.properties` file, located in the `Signon/resources` directory.

By default, messages are associated with specific events. For example, if a logon attempt fails because the user password has expired, the logon page displays a message alerting the user that the password has expired. For security reasons, you might use the same general error message for all logon failures.

To customize user messages:

1. From the SSP `install_dir\Signon\resources` directory, open the `messageBundle.properties` file in a text editor.
2. Modify the message text for all user messages you want to customize.
3. Save the `messageBundle.properties` file.
4. Restart SSP.

Configure the Forgot Your User ID or Password Page

You can configure the Forgot Your User ID or Password page to display a customized user message. Customize this message by editing the Login page `.html` file.

To customize the Forgot Your User ID or Password page, open the `\install_dir\signon` directory and modify the `login.html` file as required.

Note: If you modify the `login.html` file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

If you modify the `login.html` file, create a copy of the `\install_dir\Signon` directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Configure SSP to Use External Logon Portal

You can configure SSP to use an external logon portal.

Before you configure an external logon portal, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
External Portal		
◆ External Application Login URL	URL of external logon portal.	

To configure SSP to use the external logon portal:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select External Portal.
4. Type the URL of the external login portal in the External Application Login URL field.
5. Click Save.

Configure a Single Sign-on Connection to an FTP Server

Sterling Secure Proxy (SSP) can be used as a proxy with Sterling Integrator and Sterling File Gateway (SFG) and supports a single sign-on connection for FTP connections. Single sign-on (SSO) bypasses the normal user authentication process in Sterling Integrator and instead trusts that SSP has authenticated the user.

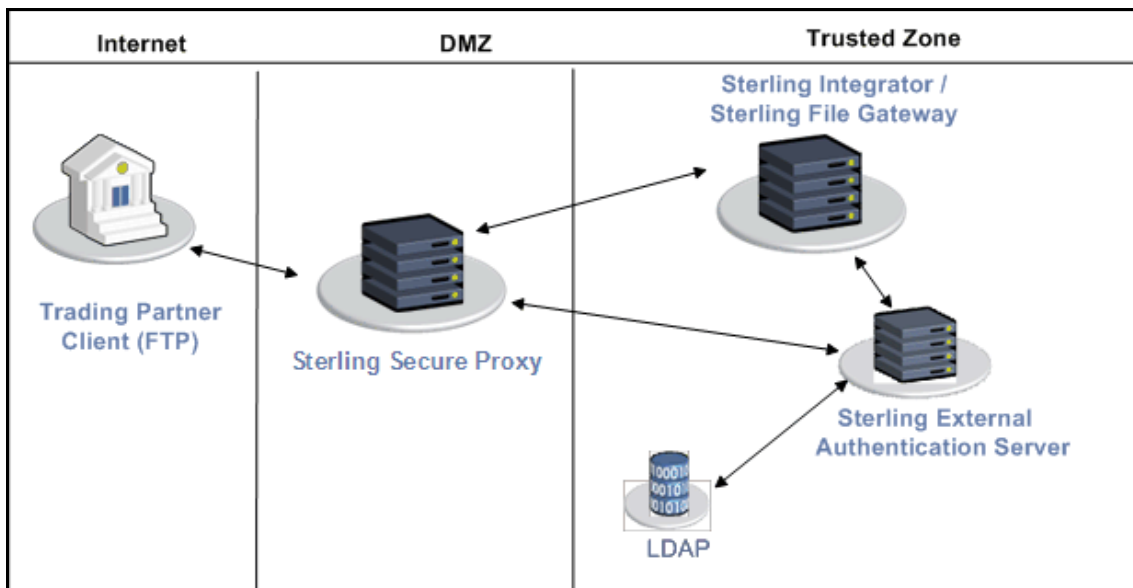
To support single sign-on, configure EA to generate SSO tokens. Configuring SSO allows a trading partner to log on and use the same login session to connect to SSP and Sterling Integrator. By default, EA uses OpenSAML to create and manage SSO tokens. However, you can customize your environment to use a third-party application to generate tokens.

This topic describes how to configure the FTP protocol in SSP between the trading partner and SSP and between SSP and Sterling Integrator to enable authentication through EA. It describes how to configure EA to issue tokens to authenticate the connection between SSP and Sterling Integrator. It also describes how to configure a self-service Change Password Portal for external trading partners.

Flow of Data for Single Sign-On Configuration Between Sterling Integrator and SSP

After you set up the basic single sign-on configuration, trading partners can communicate in a secure environment that provides authentication. The trading partner first connects to SSP which then connects to Sterling Integrator on behalf of the trading partner.

Following is an illustration of the flow of data:



Following are the steps that occur during a single sign-on session between a trading partner, SSP, and Sterling Integrator when EA is used to generate and manage tokens:

1. The trading partner requests a connection to Sterling Integrator.

2. SSP receives the request, and the SSL handshake between SSP and the trading partner begins. If SSL authentication is configured, the proxy submits its certificate to the trading partner. If client authentication is configured, the trading partner then submits its certificate to SSP for authentication. You can optionally configure SSP to enforce client authentication and send the certificate to EA for validation.
3. SSP sends an authentication request to the trading partner, who provides his user ID and password.
4. SSP sends the user ID and password to EA and then validates it against information stored in LDAP.
5. If the credentials are valid, EA creates an OpenSAML v2 token and returns the token to SSP.
6. SSP connects to Sterling Integrator and performs an SSL handshake. SSP then sends the request with the token from EA to Sterling Integrator.
7. Sterling Integrator validates the token against EA and begins normal operation.

Configuration Considerations

Before you complete the single sign-on configuration, be aware of the following considerations:

Only the HTTP, Connect:Direct, FTP, and SFTP protocols support single sign-on connections.

The SSP Change Password Portal requires an HTTP adapter, which is an optional, licensed component of SSP, and a license for the Change Password Portal. Refer to *Configure Change Password Portal* on page 421 for instructions on how to configure this feature.

Organization of Single Sign-On Scenarios

The scenarios describe how to configure single sign-on between SSP and trading partners and between SSP and Sterling Integrator.

Configure the Basic Scenario to Enable a Connection to Sterling Integrator

Configure the basic scenario to allow you to quickly become operational using single sign-on to connect to an FTP Server Adapter in Sterling Integrator. After you complete this scenario, test the connection to ensure that you have correctly configured it. You then have a basic configuration and can begin operation.

Configure Advanced Features

After you configure the basic SSO setup, determine if your environment requires an advanced feature. Following are the advanced features:

Use a third-party application to configure tokens. The basic scenario uses EA to configure and manage tokens. To use a third-party application to configure tokens, you complete additional setup procedures. Refer to *Allow a Third-Party Provider to Create Tokens* on page 417.

Customize the OpenSAML v2 tokens—You use the default token generation definition when you configure the basic single sign-on definition. To customize the token definition, you can modify the named identity provider, the token signing key, or how long a token can be used before it expires. Refer to *Customize Token Definitions Created by EA* on page 419.

Configure Sterling Integrator or SFG with additional pools—You use additional pools to support more than one EA server. Refer to *Configure Sterling Integrator or SFG to use multiple EA servers* on page 419.

Configure Optional Features

SSP provides optional security features and you can configure them as required for your environment. Refer to FTP Reverse Proxy Configuration for instructions on how to configure those features.

EA provides the ability to configure multifactor authentication. In addition to configuring client authentication in SSP, EA can also authenticate the IP address, certificate, password, and/or group access. Refer to the Sterling External Authentication Server documentation library.

SSP provides a password portal that allows trading partners to manage their own passwords. This feature requires the HTTP adapter, which is an optional component of SSP, and an optional license for the Change Password Portal. Refer to *Configure Change Password Portal* on page 421 for instructions on how to configure this feature.

Worksheets

Before you complete each procedure, gather the information you need to configure on the worksheet provided. For each worksheet:

- Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

- Accept default values for fields not listed.

- Note the Configuration Manager field where you will specify the value.

Basic Single Sign-On Scenario

Complete the following tasks to define an FTP configuration between a trading partner and SSP and between SSP and Sterling Integrator to support a single sign-on connection to an FTP Server Adapter:

- Configure SSP to support basic single sign-on.

- Use the default single sign-on configuration in EA to manage OpenSAML v2 tokens.

- Prepare Sterling Integrator to support the single sign-on option.

- Validate the connections between the trading partner, SSP, and Sterling Integrator.

Configure SSP for Basic Single Sign-On

Complete the following procedures to configure SSP for basic single sign-on:

- Create an FTP policy to support a single sign-on connection to Sterling Integrator.

- Define an FTP netmap to identify inbound and outbound connections.

- Define an FTP Reverse Proxy adapter.

Create an FTP Policy to Support a Single-Sign On Connection For FTP

To create an FTP policy to support a single sign-on connection to SFG:

1. From the Internal User ID field, enable SSO token from External Authentication.

2. Enable Through External Authentication.
3. Type the definition you defined in EA in the External Authentication Profile field.

For more information about configuring the FTP policy, refer to FTP Reverse Proxy Configuration.

Create an FTP Netmap to Support a Single Sign-On Connection to Sterling Integrator

To create an FTP netmap to support a single sign-on connection to Sterling Integrator:

1. Configure the inbound node information for your external trading partners. The inbound node must use the preceding FTP policy to support SSO.
2. Configure the outbound node information for your FTP server.

For more information about configuring the FTP netmap, FTP Reverse Proxy Configuration.

Define the FTP Adapter Used for the Single Sign-On Connection

To create an FTP adapter to support a single sign-on connection to Sterling Integrator:

1. Specify the netmap you created for the single sign-on connection to Sterling Integrator for the Netmap field.
2. Specify the name of the outbound node for the Standard Routing Node field.
3. Specify your EA server in the External Authentication Server field.

Configure Sterling EA to Support Single Sign-On

To allow an SSO connection between a trading partner and SSP to route traffic to Sterling Integrator, you configure OpenSAML v2.0 tokens in EA. You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines which options are enabled. Refer to the Sterling External Authentication Server documentation library for instructions.

The EA server generates and manages tokens. A default configuration called SEAS-SAML is enabled when you install EA. If you use the default configuration, EA is the identity provider, token signing keys are automatically generated, and the token expires after 15 minutes. Use the default configuration when you configure basic single sign-on.

To customize EA for single sign-on, refer to *Customize Token Definitions Created by EA* on page 419.

Prepare Sterling Integrator to Support Single Sign-On

Before you enable single sign-on between a trading partner and Sterling Integrator, when using SSP, you modify the Sterling Integrator installation. The files required to enable SSO are installed with EA.

Prepare Sterling Integrator to Support Single Sign-On on UNIX or Linux

To prepare Sterling Integrator to support SSO on UNIX or Linux:

1. From the EA server, copy the files and subdirectories from the *EA_install_dir/lib/sterling/sfg-ss-plugin* directory to a location that is accessible to the Sterling Integrator server, where *EA_install_dir* is the location of the EA installation.

Note: If you use FTP to copy the files to the Sterling Integrator server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the Sterling Integrator server, move to the *Sterling_Integrator_install_dir/properties* directory, where *Sterling_Integrator_install_dir* is the Sterling Integrator installation directory.
3. Type the following commands to copy the SSO properties files to the Sterling Integrator server, where *base_dir* is the location where you copied the files in step 1:

```
cp base_dir/sfg-ss-plugin/properties/security.properties_seas-ss-ext.in .
cp base_dir/sfg-ss-plugin/properties/authentication_policy.properties_seas-auth_ext.in .
cp base_dir/sfg-ss-plugin/properties/servers.properties_seas-ss-ext .
cp base_dir/sfg-ss-plugin/properties/servers.properties_seas-auth_ext .
```

4. Stop Sterling Integrator if it is running.
5. In the *server.properties_seas-ss-ext* file, uncomment the following line and replace *<SI_install>* with the actual installation path for Sterling Integrator:

```
# seas-ss=<SI_install>/properties/seas-ss/1.0/seas-ss.properties
```

6. In the *server.properties_seas-auth_ext* file, uncomment the following line and replace *<SI_install>* with the actual installation path for Sterling Integrator:

```
# seas-auth=<SI_install>/properties/seas-auth/1.0/seas-auth.properties
```

7. From the *Sterling_Integrator_install_dir/bin* directory, type the following commands:

```
./install3rdParty.sh seas-ss 1.0 -j /sfg-ss-plugin/seas-ss.jar
./install3rdParty.sh seas-ss 1.0 -p /sfg-ss-plugin/properties/seas-ss.properties
./install3rdParty.sh seas-auth 1.0 -p /sfg-ss-plugin/properties/seas-auth.properties
```

8. From the *Sterling_Integrator_install_dir/jar/seas-ss/1.0* directory, create a subdirectory named *private*.
9. Move to the */private* directory.
10. Type the following command to copy the jar files to the */private* directory on the Sterling Integrator server:

```
cp /sfg-ss-plugin/private/*.jar .
```

Modify Sterling Integrator to Support Single Sign-On on UNIX or Linux

Before Sterling Integrator supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to `customer_overrides.properties` to prevent custom settings from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. In the `install_dir/properties` directory, locate or create the `customer_overrides.properties` file.
2. Open the file in a text editor and add the properties that you want to override.

Add the following parameters to configure the connection to EA:

- ◆ `seas-sso.EA_HOST=IP address or host name of EA server`
- ◆ `seas-sso.EA_PORT=listen port of EA server`

Specify the appropriate secure or clear listen port from the EA server configuration.

- ◆ `seas-sso.EA_PS_NAME=perimeter server used to connect to EA`
- ◆ `seas-sso.EA_SECURE_CONNECTION=enables a secure EA`

`true` sets connections to EA as secure and `false` sets the connection as clear. If this parameter is true, you must also define the `EA_SYSTEM_CERT` and `EA_TRUSTED_CERT[1]`.

- ◆ `seas-sso.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.`
- ◆ `seas-sso.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.`

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the `seas-sso.EA_TRUSTED_CERT(#)` parameter. For example, for the first certificate, configure the parameter, `seas-sso.EA_TRUSTED_CERT[1]`; for the second certificate, define `seas-sso.EA_TRUSTED_CERT[2]`, until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart Sterling Integrator to use the new values.

Prepare Sterling Integrator to Support Single Sign-On on Windows

To prepare Sterling Integrator to support SSO on Windows:

1. From the EA server, copy the files and subdirectories from the *EA_install_dir*\lib\sterling\sfg-ss0-plugin directory to a location that is accessible to by the Sterling Integrator server.

Note: If you use FTP to copy the files to the Sterling Integrator server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the Sterling Integrator server, move to the *Sterling_Integrator_install_dir*\properties directory.
3. Type the following command to copy the SSO security.properties file to the Sterling Integrator server, where *base_dir* is the location where you copied the files in step 1:

```
copy      \sfg-ss0-plugin\properties\security.properties_seas-ss0_ext.in .
copy      \sfg-ss0-plugin\properties\authentication_policy.properties_seas-auth_ext.in .
copy      \sfg-ss0-plugin\properties\server.properties_seas-ss0_ext .
copy      \sfg-ss0-plugin\properties\server.properties_seas-auth_ext .
```

4. Stop Sterling Integrator if it is running.
5. In the server.properties_seas-ss0_ext file, uncomment the following line and replace <SI_install> with the actual installation path for Sterling Integrator:

```
# seas-ss0=<SI_install>\properties\seas-ss0\1.0\seas-ss0.properties
```

6. In the `server.properties_seas-auth_ext` file, uncomment the following line and replace `<SI_install>` with the actual installation path for Sterling Integrator:

```
# seas-auth=<SI_install>\properties\seas-auth\1.0\seas-auth.properties
```

7. From the `Sterling_Integrator_install_dir\bin` directory, type the following commands:

```
install3rdParty.cmd seas-sso 1.0 -j          \sfg-sso-plugin\seas-sso.jar
install3rdParty.cmd seas-sso 1.0 -p          \sfg-sso-plugin\properties\seas-sso.properties
install3rdParty.cmd seas-auth 1.0 -p        \sfg-sso-plugin\properties\seas-auth.properties
```

8. From the `Sterling_Integrator_install_dir\jar\seas-sso\1.0` directory, create a subdirectory named `private`.
9. Go to the `\private` directory.
10. Type the following command to copy the jar files to the Sterling Integrator server:

```
copy          \sfg-sso-plugin\private\*.jar .
```

Modify Sterling Integrator to Support Single Sign-On on Windows

Before Sterling Integrator supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to `customer_overrides.properties` to prevent custom settings from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. In the `install_dir\properties` directory, locate or create the `customer_overrides.properties` file.
2. Open the file in a text editor and add the properties that you want to override.

Add the following parameters to configure the connection to EA:

- ◆ `seas-sso.EA_HOST=IP address or host name of EA server`
- ◆ `seas-sso.EA_PORT=listen port of EA server`

Specify the appropriate secure or clear listen port from the EA server configuration.

- ◆ `seas-sso.EA_PS_NAME=perimeter server used to connect to EA`
- ◆ `seas-sso.EA_SECURE_CONNECTION=enables a secure EA`

`true` sets connections to EA as secure and `false` sets the connection as clear.

If this parameter is true, you must also define the `EA_SYSTEM_CERT` and `EA_TRUSTED_CERT[1]`.

- ◆ `seas-sso.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.`

- ◆ seas-ss0.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the seas-ss0.EA_TRUSTED_CERT(#) parameter. For example, for the first certificate, configure the parameter, seas-ss0.EA_TRUSTED_CERT[1]; for the second certificate, define seas-ss0.EA_TRUSTED_CERT[2], until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart Sterling Integrator to use the new values.

Change Sterling Integrator User Accounts for Single Sign-On

After you install and configure the SSO plug-in and restart Sterling Integrator, the Sterling Integrator User Accounts page presents additional choices for Authentication Type. To enable SSO for a user account, select an authentication method of External and then select the appropriate EA server for Authentication Host. The default is SEAS Authentication. Additional choices are available only if you define other EA Connection pools in addition to the default SSO_POOL.

Verify That Sterling Integrator is Configured for Single Sign-On

Before you configure additional functions, make sure that Sterling Integrator is ready for use in a single sign-on environment. To verify the configuration, start Sterling Integrator.

View the authentication.log and security.log to make sure the Sterling Integrator files are updated. If the update was successful, log files display the success messages.

Authentication.log file displays the following messages:

```
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SI is configured to support
single sign-on
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SSO Property :
SSO_AUTHENTICATION_CLASS.1 = Class name :
com.sterlingcommerce.seas.gis.sso.plugin.SeasSsoProvider
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication - A new SSO
Authentication Policy has been installed.
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication on Page:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager Number of SSO Authentication
Plug-In:1
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO configuration
policy...SSOAuthenticationPolicy isComplete=true isEnabled=true
httpUserIdHeader=SM_USER
ALL 000000000000 GLOBAL_SCOPE SecurityManager initialization complete.
```

Security.log displays the following message:

```
ALL 000000000000 GLOBAL_SCOPE SEAS-SSO: Plug-in initialized
```

Verify the SSP Connections

After you configure the basic single sign-on environment, to verify that the engine can receive and initiate communication sessions, you have to establish a connection between an FTP client and the engine, initiate a session from the engine to the Sterling Integrator server in the trusted zone, and review the SSP audit log for the results.

Note: Configuration files must be available on the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

Establish an FTP session initiated by a trading partner using an FTP client

Initiate an outbound session to an Sterling Integrator FTP Server Adapter on behalf of the FTP client connection

To verify the communications sessions:

1. Make sure the engine is running.
2. Initiate an FTP client session to the Sterling Integrator server in your trusted zone.
3. View the Inbound Node Log and the Outbound Node Log.
4. Confirm that the data transfer was successful, as illustrated in the sample log below:

Sample Inbound Node Log

```

11 Sep 2009 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134 SSP104I
Session: 1 - Session Proceeding after Node match: Any
11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.246.42 DPORT=10054 SUID=admin
DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]

```

Sample Outbound Node Log

```

11 Sep 2009 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134 SSP104I
Session: 1 - Session Proceeding after Node match: Any
11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.246.42 DPORT=10054 SUID=admin
DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]

```

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Advanced Features of the Single Sign-On Configuration

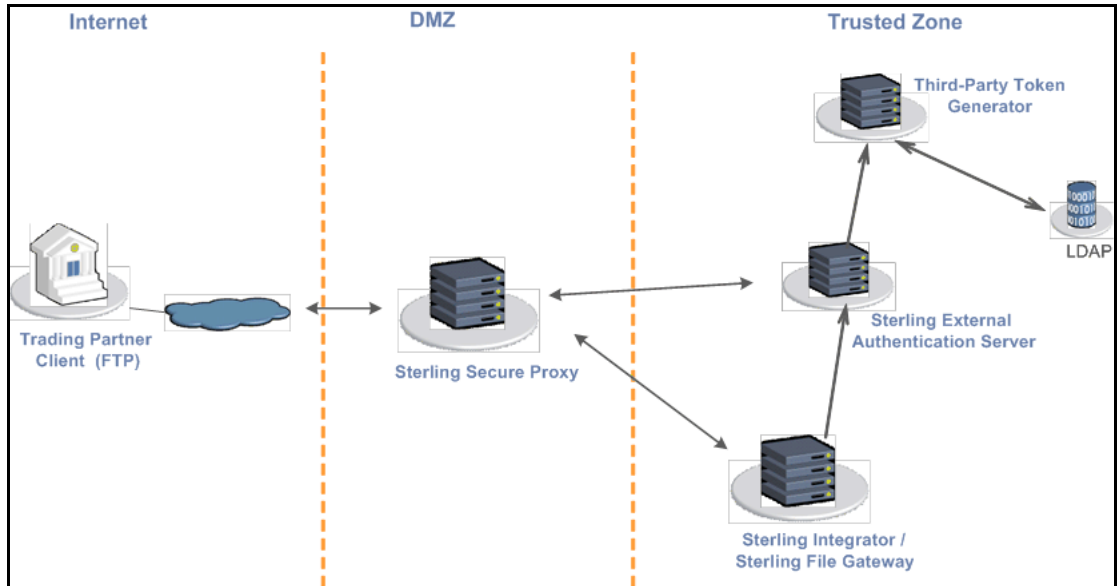
This section provides instructions on the additional features you can configure by modifying the basic single sign-on scenario. Variations include:

- Allow a third-party provider to create tokens
- Customize token definitions created by EA
- Configure Sterling Integrator or SFG to use multiple EA servers
- Configure Change Password Portal

Allow a Third-Party Provider to Create Tokens

You used the default OpenSAML token generation definition when you configured the basic single sign-on definition. The default configuration uses EA to manage tokens. To use a third-party application for token generation, you must modify the SSO Token setup in EA.

The following diagram illustrates the flow using a third-party application for token generation.



Configure EA to Enable a Third-Party Provider to Create Tokens

Before you configure EA to enable a third-party application to create tokens, gather the following information:

Configuration Manager Field	Feature	Value
Token Manager	The application that creates the tokens.	Custom
Class Name	Name of the Java class that implements the Token Manager interface.	
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To configure EA and enable a third-party application to generate tokens:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. To configure a token manager other than EA, select Custom from the Token Manager field.
5. Type the class name in the Class name field.
6. To change how long a token can be used before it expires, type a new value in the Token Expiration Period field.
7. Click OK.

Customize Token Definitions Created by EA

You used the default token definition when you configured the basic single sign-on definition. To customize the token definition, complete the following procedure. You can modify the named identity provider, the token signing key, or how long a token can be used before it expires.

Before you customize token definitions, gather the following information:

Configuration Manager Field	Feature	Value
Named Identity Provider	The prefix appended to generated tokens to identify the provider. Note: If you change the identity provider name, any outstanding tokens are invalid.	
Token Signing Key	Alias of the key certificate used to sign the token.	
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To customize the token configuration in EA:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. Customize one or more of the following definitions:
 - ◆ Named Identity Provider
 - ◆ Token Signing Key
 - ◆ Token Expiration Period
5. Click OK.

Configure Sterling Integrator or SFG to use multiple EA servers

If you are implementing single sign-on for HTTP with Basic Authentication, or for protocols other than HTTP, and you need to support additional EA servers, add the following parameters for each additional EA pool configuration to the `customer_overrides.properties` file located in the `install_dir\properties` directory.

Note: The SFG myFileGateway and FileGateway applications always use the default SSO_POOL EA connection to validate SSO tokens, regardless of which Authentication Host is selected for the user. Additional EA connection pools may only be used for HTTP Basic Auth applications, FTP, SFTP, and Connect:Direct.

authentication_policy.authentication_n.className=
com.sterlingcommerce.seas.gis.sso.plugin.SeasAuthentication

authentication_policy.authentication_n.display_name =
name to be used on the Sterling Integrator/SFG user administration UI. Use something
different than the default SEAS Authentication, which is used by the default SSO_POOL. This
is the Authentication Host name that is selected when you configure external User Accounts to
use this pool.

authentication_policy.authentication_n.enabled=true

seas-auth.authentication_n.profile = userAuth

seas-auth.authentication_n.ea_pool=unique name for your pool other than the default
SSO_POOL, which shares the EA connection pool with the SFG SSO configuration

Note: Change the "n" in the above example to a number greater than 1 to avoid overwriting the default SSO_POOL, which is shared with the FileGateway/myFileGateway SSO configuration. Also, make sure you avoid using a number already in use for LDAP authentication. Define a unique number for each entry.

To use another connection pool instead of the default SSO_POOL, configure the EA connection of the pool with the following parameters, where *pool* is the pool name defined in the preceding section:

seas-auth.*pool*.EA_HOST=*IP address or host name of EA server*

seas-auth.*pool*.EA_PORT=*listen port of EA server*

seas-auth.*pool*.EA_PS_NAME=*perimeter server used to connect to EA*

seas-auth.*pool*.EA_SECURE_CONNECTION=enables a secure EA

true sets connections to EA as secure and *false* sets the connection as clear. If this parameter is true, you must also define the EA_SYSTEM_CERT and EA_TRUSTED_CERT[1].

seas-auth.*pool*.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure

seas-auth.*pool*.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections

seas-auth.*pool*.TIMEOUT=maximum time to wait for making EA connections and receiving responses

seas-auth.*pool*.TIMEOUT_UNITS=unit of time to use, minutes or seconds, for

seas-auth.*pool*.TIMEOUT parameter

seas-auth.*pool*.PERSISTENT_EA_CONNECTIONS=whether to keep persistent connections to EA

true sets connections to EA as persistent and *false* sets the connections as not persistent.

seas-auth.*pool*.MAX_EA_CONNECTIONS=maximum number of EA connections

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

Configure Change Password Portal

SSP provides a customizable self-service Change Password Portal that allows FTP, SFTP, and Connect:Direct users to manage and change their passwords from a web browser. Password management capabilities of the Change Password Portal include verification of the new password, password expiration notification, password will expire notification, display of password policy, and customizable screens. You can also configure SSP to use an external logon portal.

To support the Change Password Portal, configure SSP SSO for the HTTP protocol.

This topic describes how to configure and customize the SSP Change Password Portal.

Change Password Portal Page Flow

The workflow of the Change Password Portal is not configurable. In the following scenario, the trading partner decides to change his password.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials. If the credentials are valid, a Change Password page is displayed. The user can view the password policy or change his password. If the user successfully changes his password, SSP presents a successful password change message. If the user fails to successfully change his password, a Change Password page is displayed with an error message.

To change a password, the user must follow the restrictions defined in the password policy. However, the user will not be locked out of SSP if he does not define a valid new password.

Note: The setting to allow the trading partner to change his password is in the EA group profile. This profile is specified in the protocol policy configuration.

Configuration Considerations

Before you complete the Change Password Portal configuration, be aware of the following considerations:

SSP SSO must be properly configured before you can use the change password functionality of the Change Password Portal.

Configure Active Directory or LDAP to allow the trading partner to change his password.

Comments are included in the default Change Password Portal .html pages to simplify editing. Remove all of these comments after you edit the pages to minimize security risks.

Configuration Overview

Complete the following procedures to configure an SSP Change Password Portal for the FTP protocol:

Create an SSO Configuration for Change Password Portal on page 422

Define a Policy for Change Password Portal on page 423

Define a Netmap for Change Password Portal on page 423

Define an HTTP adapter for Change Password Portal on page 423

Create an SSO Configuration for Change Password Portal

Before you create an SSO configuration, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSO configuration file	
Fully Qualified Host name the Trading Partner connects to	Fully qualified host name of the FTP server	
Login Directory Name	Unique directory that contains Change Password Portal .html files	

To define an SSO configuration:

1. Click Advanced from the menu bar.
2. Click Actions>New SSO Configuration.
3. On the Basic tab, type a configuration name in the Name field.
4. Type the host name in the Fully Qualified Host name the Trading Partner connects to field.
5. On the Advanced tab, select Change Password Page in the Default Landing Page field.
6. Uncheck SSO Cookie Secure Flag only if the inbound connection is unsecure (HTTP, not HTTPS).

7. On the Logon Portal tab, define unique Login Directory Name, Login Page and Change Password Page if necessary or accept the defaults to use those included with SSP.
8. Click Save.

Define a Policy for Change Password Portal

Create a new HTTP policy or use an existing one.

Define a Netmap for Change Password Portal

Create a new HTTP netmap or use an existing one.

Define an HTTP adapter for Change Password Portal

Create a new HTTP adapter to connect to the Change Password Portal.

Note: The HTTP adapter is an optional component of SSP.

Before you create an HTTP adapter configuration, gather the following information:

CM Field	Feature	Value
Adapter Name	Name to assign to the HTTP adapter configuration file	
Listen Port	Port for the Change Password Portal	
Netmap	Netmap you create for Change Password Portal	
SSO Configuration	SSO configuration you create for Change Password Portal	

To define a Logon Portal HTTP adapter:

1. Click Configuration from the menu bar.
2. Click Actions>New Adapter>HTTP Reverse Proxy.
3. Type a configuration name in the Adapter Name field.
4. In the Listen Port field, enter a value for the listening port of the HTTP adapter.
This is the Change Password Portal listening port.
5. Select the Netmap for the Change Password Portal from the Netmap list.
6. Select noRouting from the Routing Type list.
This turns the HTTP adapter into a Change Password Portal and prevents the adapter from being used to connect to any outbound node.
7. Select the SSO configuration from the SSO Configuration list.
8. Complete the HTTP adapter configuration.

Organization of Change Password Portal Customization Scenarios

When you first configure SSP for FTP, you use the default Login page, Change Password page, Logout page, and Password Policy page.

This document provides instructions on how to customize SSP Change Password Portal pages.

Customize the Login Page on page 424

Customize the Change Password Page on page 425

Customize the Logout Page on page 427

Customize Password Policy Page on page 428

Customize User Messages on page 428

Configure the Forgot Your User ID or Password Page on page 429

Configure SSP to Use External Logon Portal on page 429

Customize the Login Page

The default Login page is a simple page with no logo information and prompts the user to provide a user ID and password. This default page contains a link that can be used if the user forgets his user ID or password.

You can customize a Login page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Login page, gather the following information:

CM Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Login Page	Custom page to display for the SSP single sign-on Login page.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default=text/html.	

To customize the SSP Login page:

1. Click Advanced from the menu bar.
2. Do one of the following:

- ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
 4. Change one or more of the following fields to customize the Login page:
 - ◆ Login Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
 5. Click Save.
 6. To change the text or graphics on the Login page, open the \install_dir\signon directory and modify the login .html file.

Note: If you modify the login .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the login .html file, create a copy of the \install_dir\Signon directory.
Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Change Password Page

The default Change Password page is a simple page with no logo information that prompts the user to provide his user ID, existing password, and new password. The default page provides a link to the Password Policy page.

You can customize a Change Password page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Change Password page, gather the following information:

CM Field	Feature	Value
Name	Name to assign to the SSP configuration	

CM Field	Feature	Value
SSP Internal Portal		
◆ Change Password Page	Custom page to display for the SSP single sign-on Change Password page.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Change Password page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Change Password page:
 - ◆ Change Password Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. If you want to change the text or graphics on the Change Password page, open the \install_dir\signon directory and modify the change password .html file as required.

Note: If you modify the change password .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the change password file, create a copy of the `\install_dir\Signon` directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Logout Page

The default Logout page is a simple page with no logo information. You can configure SSP to use the Login page in place of the Logout page.

You can customize a Logout page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Logout page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Logout Page	Custom page to display for the SSP single sign-on Logout page. If you want to use the Login page as the Logout page, specify the Login page here.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Logout page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.

3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Logout page:
 - ◆ Logout Password Page
Set the value to the login .html file name if you want to use the Login page in place of the Logout page.
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. If you want to change the text or graphics on the Logout page, open the `\install_dir\signon` directory and modify the `logout.html` file as required.

Note: If you modify the `logout.html` file, do not modify the following lines:

```
var ssoMsgText="{ssoMsgText}";  
var ssoMsgTitle="{ssoMsgTitle}";  
var ssoMsgType="{ssoMsgType}";  
var ssoMsgOnly="{ssoMsgOnly}";
```

7. If you modify the `logout.html` file, create a copy of the `\install_dir\Signon` directory.
Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize Password Policy Page

The Password Policy page is a simple page with no logo information that displays the password policy.

SSP obtains the password policy dynamically from Active Directory or Tivoli via EA. If you update the password policy, the Password Policy page displays the new password policy.

Customize User Messages

You can customize user messages that display on the Change Password Portal pages. Customize these messages by modifying the `messageBundle.properties` file, located in the `Signon/resources` directory.

By default, messages are associated with specific events. For example, if a logon attempt fails because the user password has expired, the logon page displays a message alerting the user that the password has expired. For security reasons, you might use the same general error message for all logon failures.

To customize user messages:

1. From the SSP `install_dir\Signon\resources` directory, open the `messageBundle.properties` file in a text editor.
2. Modify the message text for all user messages you want to customize.

3. Save the messageBundle.properties file.
4. Restart SSP.

Configure the Forgot Your User ID or Password Page

You can configure the Forgot Your User ID or Password page to display a customized user message. Customize this message by editing the Login page .html file.

To customize the Forgot Your User ID or Password page, open the \install_dir\signon directory and modify the login .html file as required.

Note: If you modify the login .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

If you modify the login .html file, create a copy of the \install_dir\Signon directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Configure SSP to Use External Logon Portal

You can configure SSP to use an external logon portal.

Before you configure an external logon portal, gather the following information:

CM Field	Feature	Value
Name	Name to assign to the SSP configuration	
External Portal		
◆ External Application Login URL	URL of external logon portal.	

To configure SSP to use the external logon portal:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select External Portal.
4. Type the URL of the external logon portal in the External Application Login URL field.

5. Click Save.

Configure a Single Sign-on Connection to an SFTP Server

Sterling Secure Proxy (SSP) can be used as a proxy with Sterling Integrator and Sterling File Gateway (SFG) and supports a single sign-on connection for SFTP connections. Single sign-on (SSO) bypasses the normal user authentication process in Sterling Integrator and instead trusts that SSP has authenticated the user.

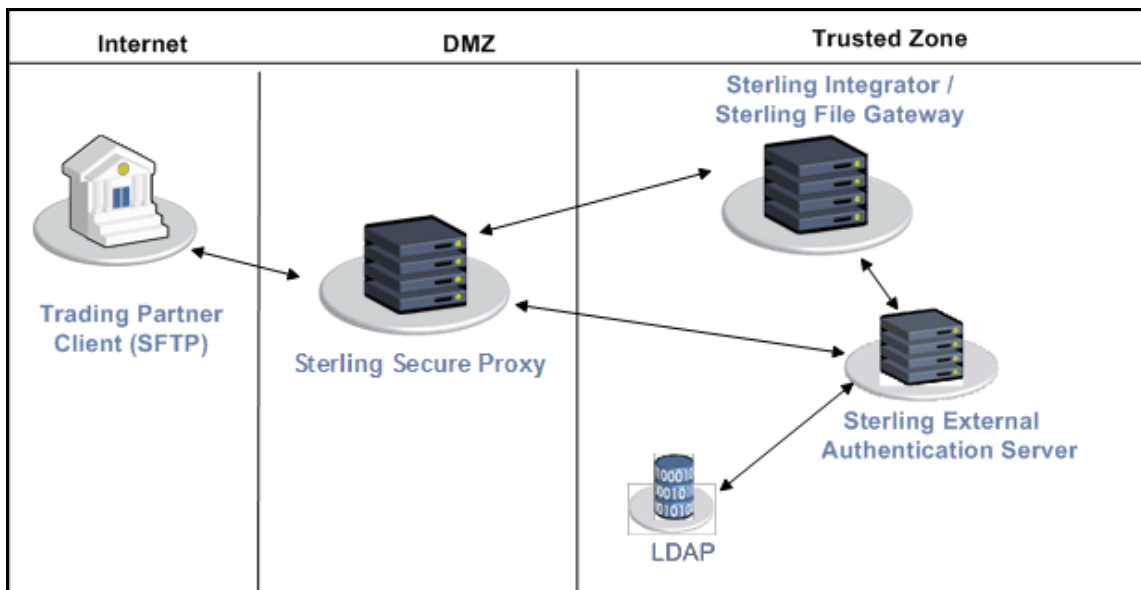
To support single sign-on, configure Sterling External Authentication Server (EA) to generate SSO tokens. Configuring SSO allows a trading partner to log on and use the same login session to connect to SSP and Sterling Integrator. By default, EA uses OpenSAML to create and manage SSO tokens. However, you can customize your environment to use a third-party application to generate tokens.

This topic describes how to configure the SFTP protocol in SSP between the trading partner and SSP and between SSP and Sterling Integrator to enable authentication through EA. It describes how to configure EA to issue tokens to authenticate the connection between SSP and Sterling Integrator. It also describes how to configure a self-service Change Password Portal for external trading partners.

Flow of Data for Single Sign-On Configuration Between Sterling Integrator and SSP

After you set up the basic single sign-on configuration, trading partners can communicate in a secure environment that provides authentication. The trading partner first connects to SSP which then connects to Sterling Integrator on behalf of the trading partner.

Following is an illustration of the flow of data:



Following are the steps that occur during a single sign-on session between a trading partner, SSP, and Sterling Integrator when EA is used to generate and manage tokens:

1. The trading partner requests a connection to Sterling Integrator.
2. SSP receives the request, and the SSH handshake between SSP and the trading partner begins. The proxy submits its key to the trading partner.
3. SSP sends an authentication request to the trading partner, who provides his user ID and password. If public key authentication is configured, the trading partner submits his key to SSP for authentication. You can optionally configure SSP to enforce key authentication and also send the key to EA for validation.
4. SSP sends either the user ID and password, key, or all three to EA, and then validates this against information stored in LDAP.
5. If the credentials are valid, EA creates an OpenSAML v2 token and returns the token to SSP.
6. SSP connects to Sterling Integrator and performs an SSH handshake. SSP then sends the request with the token from EA to Sterling Integrator.
7. Sterling Integrator validates the token against EA and begins normal operation.

Configuration Considerations

Before you complete the single sign-on configuration, be aware of the following considerations:

Only the HTTP, Connect:Direct, FTP, and SFTP protocols support single sign-on connections.

The SSP Change Password Portal requires an HTTP adapter, which is an optional, licensed component of SSP, and a license for the Change Password Portal. Refer to *Configure Change Password Portal* on page 444 for instructions on how to configure this feature.

Organization of Single Sign-On Scenarios

The scenarios describe how to configure single sign-on between SSP and trading partners and between SSP and Sterling Integrator.

Configure the Basic Scenario to Enable a Connection to Sterling Integrator

Configure the basic scenario to allow you to quickly become operational using single sign-on to connect to an SFTP Server Adapter in Sterling Integrator. After you complete this scenario, test the connection to ensure that you have correctly configured it. You then have a basic configuration and can begin operation.

Configure Advanced Features

After you configure the basic SSO setup, determine if your environment requires an advanced feature. Following are the advanced features:

Use a third-party application to configure tokens. The basic scenario uses EA to configure and manage tokens. To use a third-party application to configure tokens, you complete additional setup procedures. Refer to *Allow a Third-Party Provider to Create Tokens* on page 441.

Customize the OpenSAML v2 tokens—You use the default token generation definition when you configure the basic single sign-on definition. To customize the token definition, you can modify the named identity provider, the token signing key, or how long a token can be used before it expires. Refer to *Customize Token Definitions Created by EA* on page 442.

Configure Sterling Integrator or SFG with additional pools—You use additional pools to support more than one EA server. Refer to *Configure Sterling Integrator or SFG to use multiple EA servers* on page 443.

Configure Optional Features

SSP provides optional security features and you can configure them as required for your environment.

EA provides the ability to configure multifactor authentication. In addition to configuring key authentication in SSP, EA can also authenticate the IP address, key, password, and/or group access. Refer to the Sterling External Authentication Server documentation library for instructions.

SSP provides a password portal that allows trading partners to manage their own passwords. This feature requires the HTTP adapter, which is an optional, licensed component of SSP, and an optional license for the Change Password Portal. Refer to *Configure Change Password Portal* on page 444 for instructions on how to configure this feature.

Worksheets

Before you complete each procedure, gather the information you need to configure on the worksheet provided. For each worksheet:

- Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

- Accept default values for fields not listed.

- Note the Configuration Manager field where you will specify the value.

Basic Single Sign-On Scenario

Complete the following tasks to define an SFTP configuration between a trading partner and SSP and between SSP and Sterling Integrator to support a single sign-on connection to an SFTP Server Adapter:

- Configure SSP to support basic single sign-on.

- Use the default single sign-on configuration in EA to manage OpenSAML v2 tokens.

- Prepare Sterling Integrator to support the single sign-on option.

- Validate the connections between the trading partner, SSP, and Sterling Integrator.

Configure SSP for Basic Single Sign-On

Complete the following procedures to configure SSP for basic single sign-on:

- Create an SFTP policy to support a single sign-on connection to Sterling Integrator.

- Define an SFTP netmap to identify inbound and outbound connections.

- Define an SFTP Reverse Proxy adapter.

Create an SFTP Policy to Support a Single-Sign On Connection For SFTP

To create an SFTP policy to support a single sign-on connection to SFG:

1. Enable Through External Authentication.
2. Type the definition you defined in EA in the External Authentication Profile field.

3. From the Internal User ID field, enable SSO token from External Authentication.

Create an SFTP Netmap to Support a Single Sign-On Connection to Sterling Integrator

To create an SFTP netmap to support a single sign-on connection to Sterling Integrator:

1. Configure the inbound node information for your external trading partners. The inbound node must use the preceding SFTP policy to support SSO.
2. Configure the outbound node information for your SFTP server.

Define the SFTP Adapter Used for the Single Sign-On Connection

To create an SFTP adapter to support a single sign-on connection to Sterling Integrator:

1. Specify the netmap you created for the single sign-on connection to Sterling Integrator for the Netmap field.
2. Specify the name of the outbound node for the Standard Routing Node field.
3. Specify your EA server in the External Authentication Server field.

Configure Sterling EA to Support Single Sign-On

To allow an SSO connection between a trading partner and SSP to route traffic to Sterling Integrator, you configure OpenSAML v2.0 tokens in EA. You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines which options are enabled. Refer to the Sterling External Authentication Server documentation library for instructions.

The EA server generates and manages tokens. A default configuration called SEAS-SAML is enabled when you install EA. If you use the default configuration, EA is the identity provider, token signing keys are automatically generated, and the token expires after 15 minutes. Use the default configuration when you configure basic single sign-on.

To customize EA for single sign-on, refer to *Customize Token Definitions Created by EA* on page 442.

Prepare Sterling Integrator to Support Single Sign-On

Before you enable single sign-on between a trading partner and Sterling Integrator, when using SSP, you modify the Sterling Integrator installation. The files required to enable SSO are installed with EA.

Prepare Sterling Integrator to Support Single Sign-On on UNIX or Linux

To prepare Sterling Integrator to support SSO on UNIX or Linux:

1. From the EA server, copy the files and subdirectories from the `EA_install_dir/lib/sterling/sfg-sso-plugin` directory to a location that is accessible to the Sterling Integrator server, where `EA_install_dir` is the location of the EA installation.

Note: If you use FTP to copy the files to the Sterling Integrator server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the Sterling Integrator server, move to the *Sterling_Integrator_install_dir*/properties directory, where *Sterling_Integrator_install_dir* is the Sterling Integrator installation directory.
3. Type the following commands to copy the SSO properties files to the Sterling Integrator server, where *base_dir* is the location where you copied the files in step 1:

```
cp base_dir/sfg-ssso-plugin/properties/security.properties_seas-ssso_ext.in .
cp base_dir/sfg-ssso-plugin/properties/authentication_policy.properties_seas-auth_ext.in .
cp base_dir/sfg-ssso-plugin/properties/servers.properties_seas-ssso_ext .
cp base_dir/sfg-ssso-plugin/properties/servers.properties_seas-auth_ext .
```

4. Stop Sterling Integrator if it is running.
5. In the server.properties_seas-ssso_ext file, uncomment the following line and replace <SI_install> with the actual installation path for Sterling Integrator:

```
# seas-ssso=<SI_install>/properties/seas-ssso/1.0/seas-ssso.properties
```

6. In the server.properties_seas-auth_ext file, uncomment the following line and replace <SI_install> with the actual installation path for Sterling Integrator:

```
# seas-auth=<SI_install>/properties/seas-auth/1.0/seas-auth.properties
```

7. From the *Sterling_Integrator_install_dir*/bin directory, type the following commands:

```
./install3rdParty.sh seas-ssso 1.0 -j /sfg-ssso-plugin/seas-ssso.jar
./install3rdParty.sh seas-ssso 1.0 -p /sfg-ssso-plugin/properties/seas-ssso.properties
./install3rdParty.sh seas-auth 1.0 -p /sfg-ssso-plugin/properties/seas-auth.properties
```

8. From the *Sterling_Integrator_install_dir*/jar/seas-ssso/1.0 directory, create a subdirectory named private.
9. Move to the /private directory.
10. Type the following command to copy the jar files to the /private directory on the Sterling Integrator server:

```
cp /sfg-ssso-plugin/private/*.jar .
```

Modify Sterling Integrator to Support Single Sign-On on UNIX or Linux

Before Sterling Integrator supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to customer_overrides.properties to prevent custom settings from being overwritten when you apply patches. The customer_overrides.properties file is not changed during upgrades or patches. If the customer_overrides.properties file is not present, you must create it. Refer to the Sterling Integrator customer_overrides.properties topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. In the *install_dir/properties* directory, locate or create the *customer_overrides.properties* file.
2. Open the file in a text editor and add the properties that you want to override.

Add the following parameters to configure the connection to EA:

- ◆ *seas-ss0.EA_HOST=IP address or host name of EA server*
- ◆ *seas-ss0.EA_PORT=listen port of EA server*

Specify the appropriate secure or clear listen port from the EA server configuration.

- ◆ *seas-ss0.EA_PS_NAME=perimeter server used to connect to EA*
- ◆ *seas-ss0.EA_SECURE_CONNECTION=enables a secure EA*

true sets connections to EA as secure and *false* sets the connection as clear. If this parameter is true, you must also define the *EA_SYSTEM_CERT* and *EA_TRUSTED_CERT[1]*.

- ◆ *seas-ss0.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.*
- ◆ *seas-ss0.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.*

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the *seas-ss0.EA_TRUSTED_CERT(#)* parameter. For example, for the first certificate, configure the parameter, *seas-ss0.EA_TRUSTED_CERT[1]*; for the second certificate, define *seas-ss0.EA_TRUSTED_CERT[2]*, until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart Sterling Integrator to use the new values.

Prepare Sterling Integrator to Support Single Sign-On on Windows

To prepare Sterling Integrator to support SSO on Windows:

1. From the EA server, copy the files and subdirectories from the `EA_install_dir\lib\sterling\sfg-sso-plugin` directory to a location that is accessible to by the Sterling Integrator server.

Note: If you use FTP to copy the files to the Sterling Integrator server, be sure to transfer the .jar files in binary mode (TYPE I).

2. On the Sterling Integrator server, move to the `Sterling_Integrator_install_dir\properties` directory.
3. Type the following command to copy the SSO security.properties file to the Sterling Integrator server, where `base_dir` is the location where you copied the files in step 1:

```
copy      \sfg-sso-plugin\properties\security.properties_seas-sso_ext.in .
copy      \sfg-sso-plugin\properties\authentication_policy.properties_seas-auth_ext.in .
copy      \sfg-sso-plugin\properties\server.properties_seas-sso_ext .
copy      \sfg-sso-plugin\properties\server.properties_seas-auth_ext .
```

4. Stop Sterling Integrator if it is running.
5. In the `server.properties_seas-sso_ext` file, uncomment the following line and replace `<SI_install>` with the actual installation path for Sterling Integrator:

```
# seas-sso=<SI_install>\properties\seas-sso\1.0\seas-sso.properties
```

6. In the `server.properties_seas-auth_ext` file, uncomment the following line and replace `<SI_install>` with the actual installation path for Sterling Integrator:

```
# seas-auth=<SI_install>\properties\seas-auth\1.0\seas-auth.properties
```

7. From the `Sterling_Integrator_install_dir\bin` directory, type the following commands:

```
install3rdParty.cmd seas-sso 1.0 -j      \sfg-sso-plugin\seas-sso.jar
install3rdParty.cmd seas-sso 1.0 -p      \sfg-sso-plugin\properties\seas-sso.properties
install3rdParty.cmd seas-auth 1.0 -p     \sfg-sso-plugin\properties\seas-auth.properties
```

8. From the `Sterling_Integrator_install_dir\jar\seas-sso\1.0` directory, create a subdirectory named `private`.
9. Go to the `\private` directory.
10. Type the following command to copy the jar files to the Sterling Integrator server:

```
copy      \sfg-sso-plugin\private\*.jar .
```

Modify Sterling Integrator to Support Single Sign-On on Windows

Before Sterling Integrator supports single sign-on from an SSP environment, you must modify properties. Do not make changes directly to the properties files. Instead, make changes to `customer_overrides.properties` to prevent custom settings from being overwritten when you apply patches. The `customer_overrides.properties` file is not changed during upgrades or patches. If the `customer_overrides.properties` file is not present, you must create it. Refer to the Sterling Integrator `customer_overrides.properties` topic for more information.

To modify Sterling Integrator to enable single sign-on:

1. In the `install_dir\properties` directory, locate or create the `customer_overrides.properties` file.
2. Open the file in a text editor and add the properties that you want to override.

Add the following parameters to configure the connection to EA:

- ◆ `seas-ss0.EA_HOST=IP address or host name of EA server`
- ◆ `seas-ss0.EA_PORT=listen port of EA server`

Specify the appropriate secure or clear listen port from the EA server configuration.

- ◆ `seas-ss0.EA_PS_NAME=perimeter server used to connect to EA`
- ◆ `seas-ss0.EA_SECURE_CONNECTION=enables a secure EA`

`true` sets connections to EA as secure and `false` sets the connection as clear.

If this parameter is true, you must also define the `EA_SYSTEM_CERT` and `EA_TRUSTED_CERT[1]`.

- ◆ `seas-ss0.EA_SYSTEM_CERT=name of the system certificate in the system certificate store, if the connection is secure. Look up the system certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>System.`
- ◆ `seas-ss0.EA_TRUSTED_CERT[1]=name of the trusted certificate used by EA for secure connections. Look up the trusted certificate names in Sterling Integrator by navigating to Trading Partner>Digital Certificates>Trusted.`

If you use chained certificates and each certificate of the chain is checked in individually, you must define each of the certificates in the chain in EA. For each certificate, define a separate value, using the `seas-ss0.EA_TRUSTED_CERT(#)` parameter. For example, for the first certificate, configure the parameter, `seas-ss0.EA_TRUSTED_CERT[1]`; for the second certificate, define `seas-ss0.EA_TRUSTED_CERT[2]`, until all certificates in the chain are defined in EA. The order you configure the certificates in EA does not have to match the definitions in Sterling Integrator.

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

3. Save and close the file.
4. Stop and restart Sterling Integrator to use the new values.

Change Sterling Integrator User Accounts for Single Sign-On

After you install and configure the SSO plug-in and restart Sterling Integrator, the Sterling Integrator User Accounts page presents additional choices for Authentication Type. To enable SSO for a user account, select an authentication method of External and then select the appropriate EA server for Authentication Host. The default is SEAS Authentication. Additional choices are available only if you define other EA Connection pools in addition to the default SSO_POOL.

Verify That Sterling Integrator is Configured for Single Sign-On

Before you configure additional functions, make sure that Sterling Integrator is ready for use in a single sign-on environment. To verify the configuration, start Sterling Integrator.

View the authentication.log and security.log to make sure the Sterling Integrator files are updated. If the update was successful, log files display the success messages.

Authentication.log file displays the following messages:

```
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SI is configured to support
single sign-on
ALL 000000000000 GLOBAL_SCOPE SSOAuthenticationPolicy SSO Property :
SSO_AUTHENTICATION_CLASS.1 = Class name :
com.sterlingcommerce.seas.gis.sso.plugin.SeasSsoProvider
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication - A new SSO
Authentication Policy has been installed.
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO Authentication on Page:Enabled
ALL 000000000000 GLOBAL_SCOPE SecurityManager Number of SSO Authentication
Plug-In:1
ALL 000000000000 GLOBAL_SCOPE SecurityManager SSO configuration
policy...SSOAuthenticationPolicy isComplete=true isEnabled=true
httpUserIdHeader=SM_USER
ALL 000000000000 GLOBAL_SCOPE SecurityManager initialization complete.
```

Security.log displays the following message:

```
ALL 000000000000 GLOBAL_SCOPE SEAS-SSO: Plug-in initialized
```

Verify the SSP Connections

After you configure the basic single sign-on environment, to verify that the engine can receive and initiate communication sessions, you have to establish a connection between an SFTP client and the engine, initiate a session from the engine to the Sterling Integrator server in the trusted zone, and review the SSP audit log for the results.

Note: Configuration files must be available on the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

- Establish an SFTP session initiated by a trading partner using an SFTP client

- Initiate an outbound session to an Sterling Integrator SFTP Server Adapter on behalf of the SFTP client connection

To verify the communications sessions:

1. Make sure the engine is running.
2. Initiate an SFTP client session to the Sterling Integrator server in your trusted zone.
3. View the log file at the client to ensure that the connection from the inbound node to SSP was successful.
4. View the log file of the engine to ensure the connection to SSP was successful.

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Advanced Features of the Single Sign-On Configuration

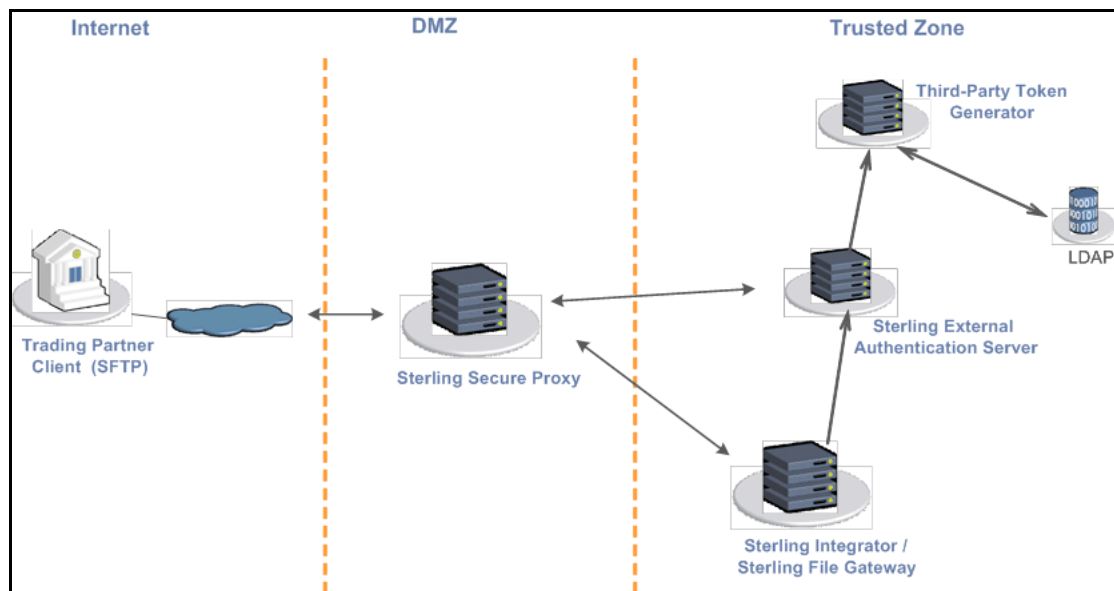
This section provides instructions on the additional features you can configure by modifying the basic single sign-on scenario. Variations include:

- Allow a third-party provider to create tokens
- Customize token definitions created by EA
- Configure Sterling Integrator or SFG to use multiple EA servers
- Configure Change Password Portal

Allow a Third-Party Provider to Create Tokens

You used the default OpenSAML token generation definition when you configured the basic single sign-on definition. The default configuration uses EA to manage tokens. To use a third-party application for token generation, you must modify the SSO Token setup in EA.

The following diagram illustrates the flow using a third-party application for token generation.



Configure EA to Enable a Third-Party Provider to Create Tokens

Before you configure EA to enable a third-party application to create tokens, gather the following information:

Configuration Manager Field	Feature	Value
Token Manager	The application that creates the tokens.	Custom
Class Name	Name of the Java class that implements the Token Manager interface.	

Configuration Manager Field	Feature	Value
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To configure EA and enable a third-party application to generate tokens:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. To configure a token manager other than EA, select Custom from the Token Manager field.
5. Type the class name in the Class name field.
6. To change how long a token can be used before it expires, type a new value in the Token Expiration Period field.
7. Click OK.

Customize Token Definitions Created by EA

You used the default token definition when you configured the basic single sign-on definition. To customize the token definition, complete the following procedure. You can modify the named identity provider, the token signing key, or how long a token can be used before it expires.

Before you customize token definitions, gather the following information:

Configuration Manager Field	Feature	Value
Named Identity Provider	The prefix appended to generated tokens to identify the provider. Note: If you change the identity provider name, any outstanding tokens are invalid.	
Token Signing Key	Alias of the key certificate used to sign the token.	
Token Expiration Period	How long a token is valid. Default is 15 minutes.	

To customize the token configuration in EA:

1. Log on to EA.
2. Select Manage>System Settings.
3. From the System Settings dialog, click the SSO Token tab.
4. Customize one or more of the following definitions:
 - ◆ Named Identity Provider
 - ◆ Token Signing Key

- ◆ Token Expiration Period
5. Click OK.

Configure Sterling Integrator or SFG to use multiple EA servers

If you are implementing single sign-on for HTTP with Basic Authentication, or for protocols other than HTTP, and you need to support additional EA servers, add the following parameters for each additional EA pool configuration to the `customer_overrides.properties` file located in the `install_dir\properties` directory.

Note: The SFG myFileGateway and FileGateway applications always use the default SSO_POOL EA connection to validate SSO tokens, regardless of which Authentication Host is selected for the user. Additional EA connection pools may only be used for HTTP Basic Auth applications, FTP, SFTP, and Connect:Direct.

```
authentication_policy.authentication_n.className=
com.sterlingcommerce.seas.gis.sso.plugin.SeasAuthentication
authentication_policy.authentication_n.display_name =
name to be used on the Sterling Integrator/SFG user administration UI. Use something
different than the default SEAS Authentication, which is used by the default SSO_POOL. This
is the Authentication Host name that is selected when you configure external User Accounts to
use this pool.
authentication_policy.authentication_n.enabled=true
seas-auth.authentication_n.profile = userAuth
seas-auth.authentication_n.ea_pool=unique name for your pool other than the default
SSO_POOL, which shares the EA connection pool with the SFG SSO configuration
```

Note: Change the "n" in the above example to a number greater than 1 to avoid overwriting the default SSO_POOL, which is shared with the FileGateway/myFileGateway SSO configuration. Also, make sure you avoid using a number already in use for LDAP authentication. Define a unique number for each entry.

To use another connection pool instead of the default SSO_POOL, configure the EA connection of the pool with the following parameters, where *pool* is the pool name defined in the preceding section:

```
seas-auth.pool.EA_HOST=IP address or host name of EA server
seas-auth.pool.EA_PORT=listen port of EA server
seas-auth.pool.EA_PS_NAME=perimeter server used to connect to EA
seas-auth.pool.EA_SECURE_CONNECTION=enables a secure EA
true sets connections to EA as secure and false sets the connection as clear. If this parameter is
true, you must also define the EA_SYSTEM_CERT and EA_TRUSTED_CERT[1].
seas-auth.pool.EA_SYSTEM_CERT=name of the system certificate in the system certificate
store, if the connection is secure
```

`seas-auth.pool.EA_TRUSTED_CERT[1]`=name of the trusted certificate used by EA for secure connections

`seas-auth.pool.TIMEOUT`=maximum time to wait for making EA connections and receiving responses

`seas-auth.pool.TIMEOUT_UNITS`=unit of time to use, minutes or seconds, for `seas-auth.pool.TIMEOUT` parameter

`seas-auth.pool.PERSISTENT_EA_CONNECTIONS`=whether to keep persistent connections to EA

true sets connections to EA as persistent and *false* sets the connections as not persistent.

`seas-auth.pool.MAX_EA_CONNECTIONS`=maximum number of EA connections

Note: Additional fields can be added if you wish to override the defaults shown below:

```
## SEAS-SSO Configuration
## HTTP cookie containing the SSO token
seas-ss0.SSO_TOKEN_COOKIE=SSOTOKEN

## Maximum time to wait for making EA connections and receiving responses
seas-ss0.SSO_TIMEOUT=30
seas-ss0.SSO_TIMEOUT_UNITS=seconds

## Whether to keep persistent connections to EA
seas-ss0.PERSISTENT_EA_CONNECTIONS=true

## Maximum number of EA connections
seas-ss0.MAX_EA_CONNECTIONS=1
```

Configure Change Password Portal

SSP provides a customizable self-service Change Password Portal that allows FTP, SFTP, and Connect:Direct users to manage and change their passwords from a web browser. The Change Password Portal is an optional, licensed component of SSP. Password management capabilities of the Change Password Portal include verification of the new password, password expiration notification, password will expire notification, display of password policy, and customizable screens. You can also configure SSP to use an external logon portal.

To support the Change Password Portal, configure SSP SSO for the HTTP protocol.

This topic describes how to configure and customize the SSP Change Password Portal.

Change Password Portal Page Flow

The workflow of the Change Password Portal is not configurable. In the following scenario, the trading partner decides to change his password.

When a user connects to SFG, SSP presents a Login page. The user provides user credentials. If the credentials are valid, a Change Password page is displayed. The user can view the password policy or change his password. If the user successfully changes his password, SSP presents a successful password change message. If the user fails to successfully change his password, a Change Password page is displayed with an error message.

To change a password, the user must follow the restrictions defined in the password policy. However, the user will not be locked out of SSP if he does not define a valid new password.

Note: The setting to allow the trading partner to change his password is in the EA group profile. This profile is specified in the protocol policy configuration.

Configuration Considerations

Before you complete the Change Password Portal configuration, be aware of the following considerations:

SSP SSO must be properly configured before you can use the change password functionality of the Change Password Portal.

Configure Active Directory or LDAP to allow the trading partner to change his password.

Comments are included in the default Change Password Portal .html pages to simplify editing. Remove all of these comments after you edit the pages to minimize security risks.

Configuration Overview

Complete the following procedures to configure an SSP Change Password Portal for the SFTP protocol:

Create an SSO Configuration for Change Password Portal on page 445

Define a Policy for Change Password Portal on page 446

Define a Netmap for Change Password Portal on page 446

Define an HTTP adapter for Change Password Portal on page 446

Create an SSO Configuration for Change Password Portal

Before you create an SSO configuration, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSO configuration file	
Fully Qualified Host name the Trading Partner connects to	Fully qualified host name of the SFTP server	
Login Directory Name	Unique directory that contains Change Password Portal .html files	

To define an SSO configuration:

1. Click Advanced from the menu bar.
2. Click Actions>New SSO Configuration.
3. On the Basic tab, type a configuration name in the Name field.
4. Type the host name in the Fully Qualified Host name the Trading Partner connects to field.
5. On the Advanced tab, select Change Password Page in the Default Landing Page field.
6. Uncheck SSO Cookie Secure Flag only if the inbound connection is unsecure (HTTP, not HTTPS).
7. On the Logon Portal tab, define unique Login Directory Name, Login Page and Change Password Page if necessary or accept the defaults to use those included with SSP.
8. Click Save.

Define a Policy for Change Password Portal

Create a new HTTP policy or use an existing one.

Define a Netmap for Change Password Portal

Create a new HTTP netmap or use an existing one.

Define an HTTP adapter for Change Password Portal

Create a new HTTP adapter to connect to the Change Password Portal.

Note: The HTTP adapter is an optional component of SSP.

Before you create an HTTP adapter configuration, gather the following information:

Configuration Manager Field	Feature	Value
Adapter Name	Name to assign to the HTTP adapter configuration file	
Listen Port	Port for the Change Password Portal	
Netmap	Netmap you create for Change Password Portal	
SSO Configuration	SSO configuration you create for Change Password Portal	

To define a Change Password Portal HTTP adapter:

1. Click Configuration from the menu bar.
2. Click Actions>New Adapter>HTTP Reverse Proxy.

3. Type a configuration name in the Adapter Name field.
4. In the Listen Port field, enter a value for the listening port of the HTTP adapter.
This is the Change Password Portal listening port.
5. Select the Netmap for the Change Password Portal from the Netmap list.
6. Select noRouting from the Routing Type list.
This turns the HTTP adapter into a Change Password Portal and prevents the adapter from being used to connect to any outbound node.
7. Select the SSO configuration from the SSO Configuration list.
8. Complete the HTTP adapter configuration.

Organization of Change Password Portal Customization Scenarios

When you first configure SSP for SFTP, you use the default Login page, Change Password page, Logout page, and Password Policy page.

This document provides instructions on how to customize SSP Change Password Portal pages.

Customize the Login Page on page 447

Customize the Change Password Page on page 449

Customize the Logout Page on page 450

Customize Password Policy Page on page 452

Customize User Messages on page 452

Configure the Forgot Your User ID or Password Page on page 452

Configure SSP to Use External Logon Portal on page 452

Customize the Login Page

The default Login page is a simple page with no logo information and prompts the user to provide a user ID and password. This default page contains a link that can be used if the user forgets his user ID or password.

You can customize a Login page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Login page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Login Page	Custom page to display for the SSP single sign-on Login page.	

Configuration Manager Field	Feature	Value
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Login page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Login page:
 - ◆ Login Page
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. To change the text or graphics on the Login page, open the \install_dir\signon directory and modify the login .html file as required.

Note: If you modify the login .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the login .html file, create a copy of the \install_dir\Signon directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Change Password Page

The default Change Password page is a simple page with no logo information that prompts the user to provide his user ID, existing password, and new password. The default page provides a link to the Password Policy page.

You can customize a Change Password page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Change Password page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Change Password Page	Custom page to display for the SSP single sign-on Change Password page.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Change Password page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Change Password page:
 - ◆ Change Password Page
 - ◆ Login Directory Name

- ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
 6. If you want to change the text or graphics on the Change Password page, open the `\install_dir\signon` directory and modify the change password .html file as required.

Note: If you modify the change password .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the change password .html file, create a copy of the `\install_dir\Signon` directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize the Logout Page

The default Logout page is a simple page with no logo information. You can configure SSP to use the Login page in place of the Logout page.

You can customize a Logout page to define how you want the page to look and what information to include on the page. You can customize this page by modifying the labels or replacing the entire page.

Before you modify the Logout page, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
SSP Internal Portal		
◆ Logout Page	Custom page to display for the SSP single sign-on Logout page. If you want to use the Login page as the Logout page, specify the Login page here.	
◆ Login Directory Name	Custom directory in the engine installation directory where the HTML files that support single sign-on are stored.	
◆ Login Page Charset	Character encoding used to create the SSP Login page. This value is sent to the browser as part of the content-type header with the Login page.	

Configuration Manager Field	Feature	Value
◆ Login Page Media Type	Media type value sent to the browser in the content-type header with the Login page. Default is text/html.	

To customize the SSP Logout page:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select SSP Internal Portal.
4. Change one or more of the following fields to customize the Logout page:
 - ◆ Logout Password Page
Set the value to the login .html file name if you want to use the Login page in place of the Logout page.
 - ◆ Login Directory Name
 - ◆ Login Page Charset
 - ◆ Login Page Media Type
5. Click Save.
6. If you want to change the text or graphics on the Logout page, open the \install_dir\signon directory and modify the logout .html file as required.

Note: If you modify the logout .html file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

7. If you modify the logout .html file, create a copy of the \install_dir\Signon directory. Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Customize Password Policy Page

The Password Policy page is a simple page with no logo information that displays the password policy.

SSP obtains the password policy dynamically from Active Directory or Tivoli via EA. If you update the password policy, the Password Policy page displays the new password policy.

Customize User Messages

You can customize user messages that display on the Change Password Portal pages. Customize these messages by modifying the `messageBundle.properties` file, located in the `Signon/resources` directory.

By default, messages are associated with specific events. For example, if a logon attempt fails because the user password has expired, the logon page displays a message alerting the user that the password has expired. For security reasons, you might use the same general error message for all logon failures.

To customize user messages:

1. From the SSP `install_dir\Signon\resources` directory, open the `messageBundle.properties` file in a text editor.
2. Modify the message text for all user messages you want to customize.
3. Save the `messageBundle.properties` file.
4. Restart SSP.

Configure the Forgot Your User ID or Password Page

You can configure the Forgot Your User ID or Password page to display a customized user message. Customize this message by editing the Login page `.html` file.

To customize the Forgot Your User ID or Password page, open the `\install_dir\signon` directory and modify the `login.html` file as required.

Note: If you modify the `login.html` file, do not modify the following lines:

```
var ssoMsgText="#{ssoMsgText}";
var ssoMsgTitle="#{ssoMsgTitle}";
var ssoMsgType="#{ssoMsgType}";
var ssoMsgOnly="#{ssoMsgOnly}";
```

If you modify the `login.html` file, create a copy of the `\install_dir\Signon` directory.

Making a copy of the directory prevents the custom attributes from being overwritten if you upgrade the software or apply a patch.

Configure SSP to Use External Logon Portal

You can configure SSP to use an external logon portal.

Before you configure an external logon portal, gather the following information:

Configuration Manager Field	Feature	Value
Name	Name to assign to the SSP configuration	
External Portal		
◆ External Application Login URL	URL of external logon portal.	

To configure SSP to use the external logon portal:

1. Click Advanced from the menu bar.
2. Do one of the following:
 - ◆ To create a new SSO configuration:
 - a. Click Actions>New SSO Configuration.
 - b. Type an SSO configuration name in the Name field.
 - ◆ To edit an existing SSO configuration:
 - a. From the navigation menu, click SSO Configurations.
 - b. Click the configuration to modify.
3. On the Logon Portal tab, select External Portal.
4. Type the URL of the external logon portal in the External Application Login URL field.
5. Click Save.

Configure Failover Support

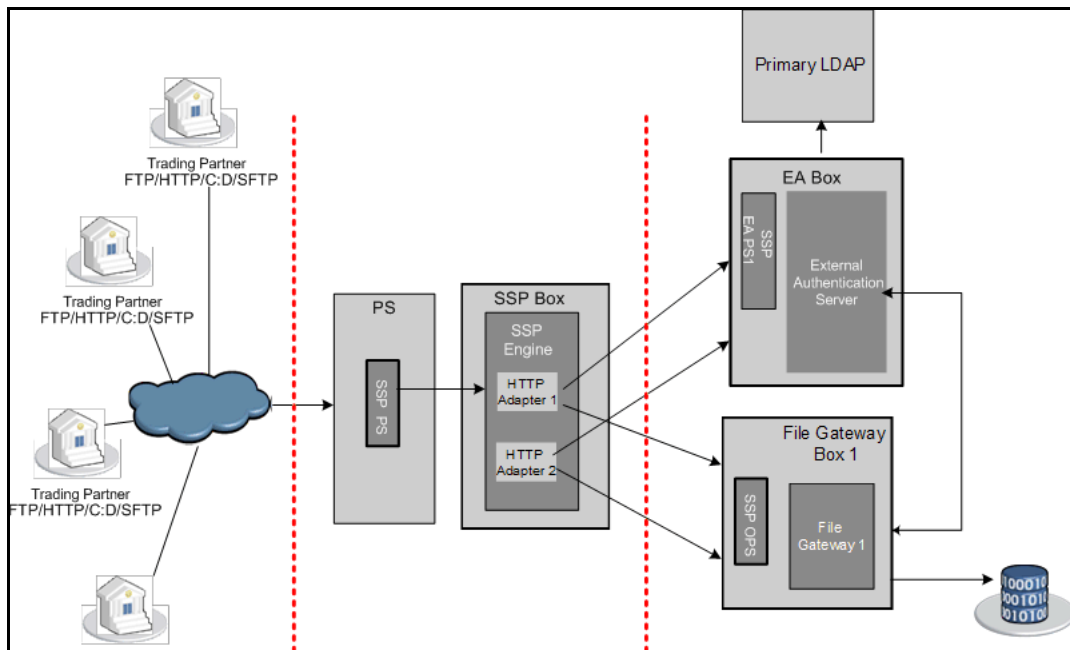
Overview of Failover Support

You can configure failover support in Sterling Secure Proxy (SSP) to manage connections from a trading partner to a company server and ensure that your operation functions even when a component such as a perimeter server, SSP engine, External Authentication server, or Sterling Integrator server in the configuration is not operational.

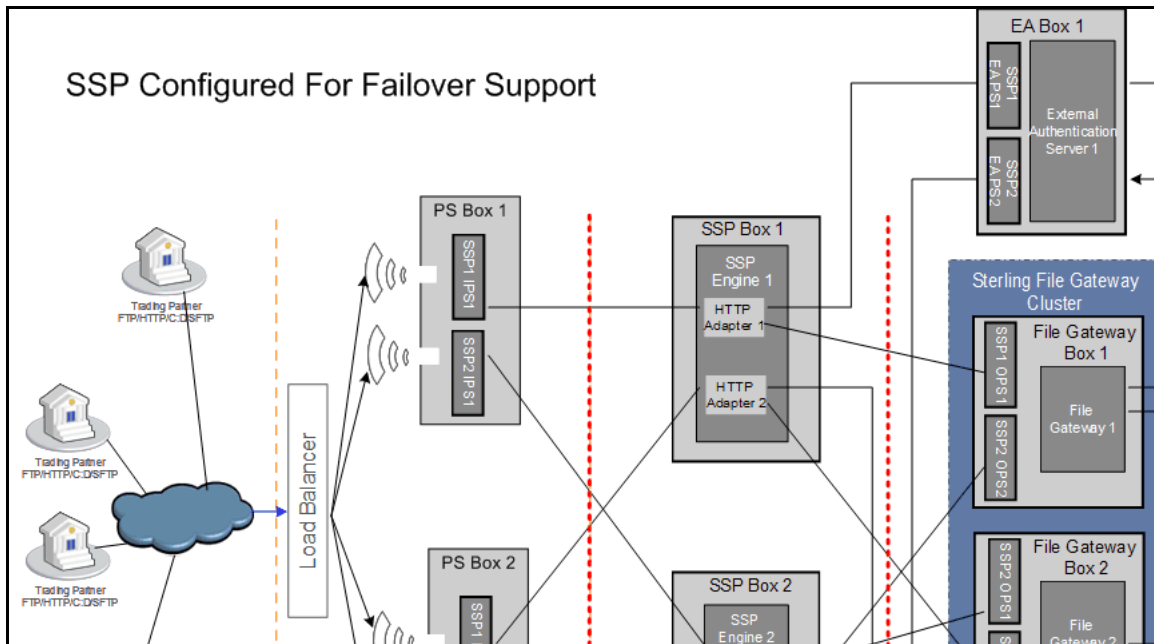
This document provides overview information about a failover environment and instructions on how to configure failover. It assumes that you have configured a basic Sterling Secure Proxy engine and defined adapters.

Illustration of Failover Support in SSP

Following is an illustration of a basic SSP configuration. It does not include any components to support failover.



The following illustration builds on the basic SSP configuration and illustrates one way to configure failover support:

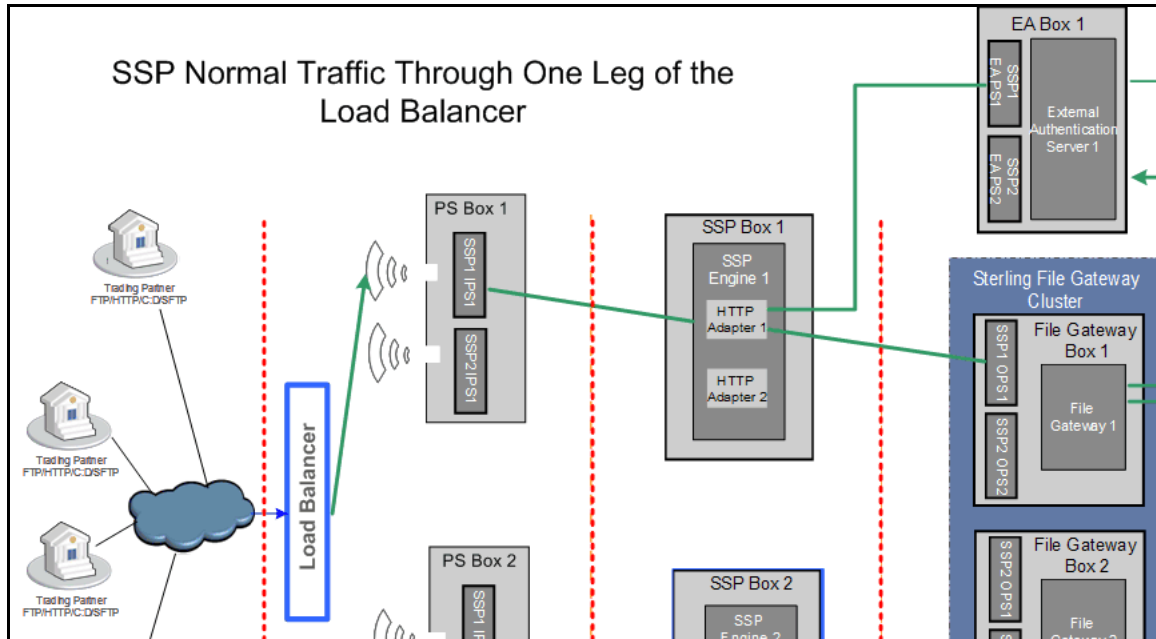


To set up the sample failover environment illustrated above, add the following components to the basic configuration:

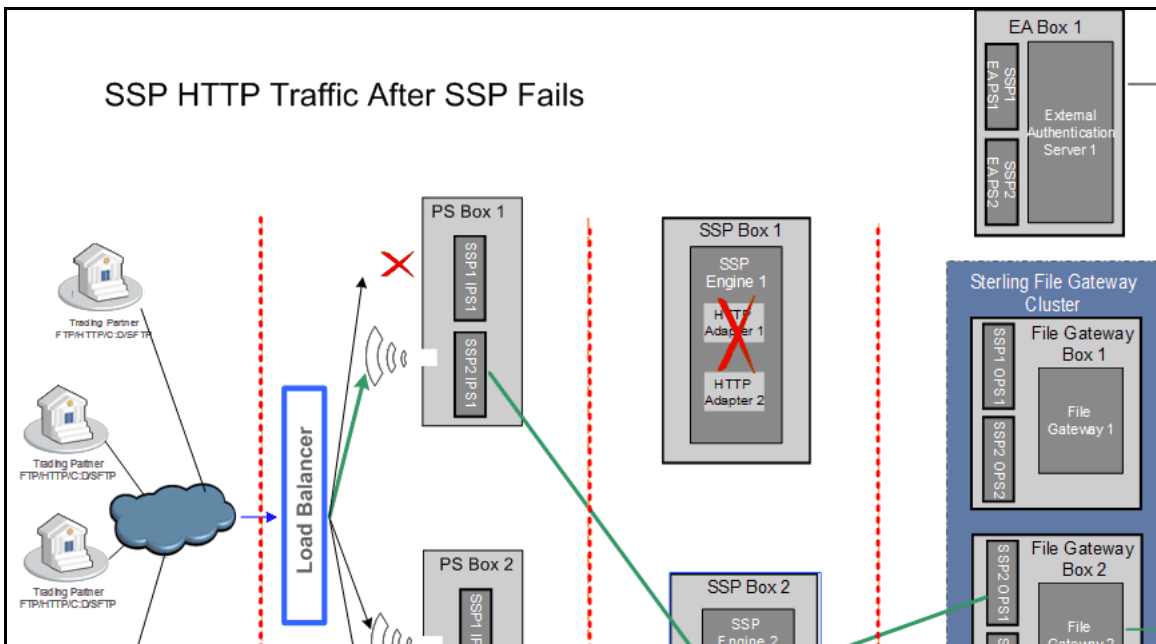
- A second external inbound PS computer (PS Box 2) with two perimeter server instances
- A second perimeter server instance on the initial inbound PS computer (PS Box 1)
- A second SSP engine (SSP Box 2)
- A second External Authentication (EA) server with two perimeter server instances (EA Box 2)
- A second perimeter server instance on EA server 1 (EA Box 1)
- A second Sterling File Gateway, configured as part of a two-node cluster (File Gateway Box 2)
- A second LDAP server (Secondary LDAP)

This sample configuration illustrates one way to set up SSP for failover support. Your failover setup depends upon your hardware configuration and security requirements.

The following diagram illustrates the flow of traffic through one leg of an SSP setup when failover support is configured and all components are operating. Traffic is allowed through the first leg of the setup and moves through the SSP engine 1.



The following diagram illustrates the flow of traffic when SSP engine 1 is not available and failover support is enabled:



Components to Configure for Failover Support

To configure a failover environment, you must add one or more of the following components to your setup:

Internal perimeter server—If you configure a perimeter server (PS) between the trading partners and SSP, configure a second PS to enable failover for the PS.

SSP engine—Install and configure a second engine in your SSP setup for failover support. You duplicate the setup of engine 1 in engine 2. However, the adapters for each engine must have unique names.

Sterling File Gateway (SFG)—Configure two instances of SFG as a cluster. The instances share a database.

External Authentication server and LDAP server—For each SFG installation, install an EA server and LDAP.

About Failover Support

When failover detection is enabled, the adapter periodically polls the status of the outbound perimeter server and the EA perimeter server. If one of the perimeter servers is down, the adapter stops the listener. When both perimeter servers are back online, the adapter restarts the listener.

If SSP tries to connect to an EA server and fails, the adapter stops the listener but continues to poll for the status of the EA server connection. When the EA server or any alternates are back online and the outbound and EA perimeter servers are connected, the listener is restarted.

If the connection to the standard outbound node and all alternate outbound nodes fail, the adapter stops the listener.

When the outbound node or any alternate nodes are back online and the outbound and EA perimeter servers are connected, the listener is restarted.

You can configure failover support for the HTTP, FTP, SFTP, and Connect:Direct protocols.

Failover Support for an External Authentication Server

Failover support for an EA server functions as follows:

1. The trading partner tries to connect to SSP Adapter 1 through the load balancer.
2. SSP authenticates the user through External Authentication Server 1 (EA). Either the EA server 1 or EA perimeter server is down and the session fails.
3. SSP does the following:
 - a. Disconnects the listen port for the Adapter 1.
 - b. Starts a background health check agent to determine when the EA server 1 and EA perimeter server are available.
4. The health check completes the following tasks:
 - a. Performs an SSL handshake if SSL is configured between SSP and EA.
 - b. Identifies if the LDAP connection between EA and LDAP is up, if a profile designated for this purpose is configured in EA.
5. The load balancer, configured to do a health check, recognizes that the adapter is not connected and no longer sends traffic to it.
6. When the health check agent detects that the External Authentication server 1 is reconnected, SSP makes the adapter 1 listen port available and the load balancer makes it available to receive traffic. The health check agent is then stopped.

Failover For a Back-end Server

After you configure failover for a back-end server, SSP functions as follows:

1. The trading partner connects to Adapter 1 through the load balancer.
2. SSP authenticates the user against information stored on the External Authentication Server 1 and connects to the back-end server.
3. If the internal perimeter server (SSP1 OPS1) or the back-end server (File Gateway 1) is not available, the session fails.
4. SSP does the following:
 - a. Disconnects the listen port for Adapter 1.
 - b. Begins a background health check agent to determine when the internal perimeter server and the back-end server are available.
 - c. Performs an SSL handshake if SSL is configured between SSP and the back-end server.
5. The load balancer recognizes that the adapter is not connected and no longer sends traffic to it.
6. After the internal perimeter server (SSP1 OPS1) and the back-end server (File Gateway 1) are back online, SSP enables the adapter 1 listen port and the load balancer makes it available for traffic.
7. The health check agent is stopped.

Overview of Failover Configuration

To configure failover support:

Configure the load balancer.

Configure additional perimeter servers.

Modify a basic SSP configuration to add support for failover.

Configure failover components for Sterling File Gateway and Sterling External Authentication.

Configure the Load Balancer

Configure the load balancer so that for each protocol service port, incoming traffic is routed to the corresponding listen port on the pool of destination inbound perimeter servers. After a client connection for a service is routed to a destination IP address, subsequent connections from the same client IP address for that service should be routed to the same destination IP address and port that the first connection was initially routed to (IP stickiness).

Summary of Steps to Set Up a Load Balancer for an HTTP Connection

To set up a load balancer for an HTTP connection:

1. Set up an HTTP monitor to monitor the HTTP adapter ports.

2. Match the value in the Send String/Receive String in the load balancer with the values in the Adapter HTTP Ping response and Ping URI fields in the adapter definition.
 - a. In the Send String field, type the following text:

```
GET pingURI
```

Following is a sample entry:

```
GET /pingResponse HTTP/1.1
```

- b. In the Receive String field, type the value of the HTTP ping response. Following is an example entry:

```
pingResponse
```

3. Provide a dummy user name and password. This information is not used to authenticate the user. It is used so the load balancer sends HTTP headers to SSP. You can use any dummy user ID and password.

Configure the Health Check Monitor for FTP

To configure the health check monitor for FTP:

1. Set up a TCP monitor to monitor the FTP adapter ports.
2. Leave the Send String field blank.
3. In the Receive string field, type the following text:

```
Server greeting banner text for FTP adapter
```

Following is a sample entry:

```
220 FTP Server ready.
```

Note: If the server greeting banner is defined in the FTP adapter, the default value is `FTP server ready.`

Configure the Health Check Monitor for SFTP

To configure the health check monitor for SFTP:

1. Set up a TCP monitor to monitor the SFTP adapter ports.
2. In the Send string field, type the following text:

```
SSH-2.0-text string of your choice
```

Following is a sample entry:

```
SSH-2.0-BigIP
```

3. In the Receive string field, type the following text:

```
pre auth banner for the SFTP adapter
```

Following is a sample entry:

```
SSH-2.0-Maverick_SSHD
```

Note: If no pre-authentication banner is defined in the SFTP adapter, the default value is `Maverick_SSHD`.

Configure the Health Check Monitor for Connect:Direct

To configure the health check monitor for Connect:Direct:

1. Set up an HTTP monitor to monitor the Connect:Direct adapter ports.

In the Send string field, type the following text:

```
GET / HTTP/1.1
```

Note: Type the Send string field exactly as identified above in order for the health check monitor to work.

In the Receive string field, type the following text:

```
HTTP/1.0 202 Will be ignored
```

Note: Type the Receive string field exactly as identified above, including the string `Will be ignored`, in order for the health check monitor to work.

The request is a ping request and SSP does not allow it to go to the back-end server.

Configure Failover Support for an HTTP Environment

To configure failover support for an HTTP adapter, install and create two SSP engines, two inbound perimeter servers, two outbound perimeter servers, two External Authentication servers, and a perimeter server to manage each EA server. Then configure the components for Each SSP engine as described in the table below.

SSP Engine 1 Configuration for HTTP

Configure the following components for Engine 1 in an HTTP environment:

Component	Field to Define	Value
HTTP Adapter 1	Adapter Name	HTTPAdapter1
	HTTP Ping Response	pingResponse
	HTTP Ping URI	/pingResponse
	External Authentication Server	External Authentication Server 1
	Inbound Perimeter Server	SSP1 IPS1
	Outbound Perimeter Server	SSP1 OPS1
	External Authentication Perimeter Server	SSP1 EA PS1
HTTP Netmap	Netmap Name	HTTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	FileGateway 1
HTTP Adapter 2	Adapter Name	HTTPAdapter2
	HTTP Ping Response	pingResponse
	HTTP Ping URI	/pingResponse
	Inbound Perimeter Server	SSP1 IPS2
	Outbound Perimeter Server	SSP1 OPS2
	External Authentication Perimeter Server	SSP1 EA PS2
	External Authentication Server	External Authentication Server 2
HTTP Netmap	Netmap Name	HTTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	FileGateway2

SSP Engine 2 Configuration for HTTP

Configure the following components for Engine 2:

Component	Field to Define	Value
HTTP Adapter 3	Adapter Name	HTTPAdapter3

Component	Field to Define	Value
	HTTP Ping Response	pingResponse
	HTTP Ping URI	/pingResponse
	External Authentication Server 2	External Authentication Server 2
	Inbound Perimeter Server	SSP1 IPS1
	Outbound Perimeter Server	SSP1 OPS1
	External Authentication Perimeter Server	SSP1 EA PS1
HTTP Netmap	Netmap Name	HTTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	FileGateway 2
HTTP Adapter 4	Adapter Name	HTTPAdapter4
	HTTP Ping Response	pingResponse
	HTTP Ping URI	/pingResponse
	External Authentication Server 1	External Authentication Server 2
	Inbound Perimeter Server	SSP1 IPS2
	Outbound Perimeter Server	SSP1 OPS2
HTTP Netmap	Netmap Name	HTTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	FileGateway1

Configure Failover Support for an FTP Environment

To configure failover support for an FTP adapter, install and create two SSP engines, two inbound perimeter servers, two outbound perimeter servers, two External Authentication servers, and a perimeter server to manage each EA server. Then configure the components each SSP engine as described in the table below.

To configure failover support, configure the following components in SSP.

SSP 1 Engine Configuration for FTP

Configure the following components for Engine 1:

Component	Field to Define	Value
FTP Adapter 1	Adapter Name	FTPAdapter1
	Inbound Perimeter Server	SSP1 IPS1
	Outbound Perimeter Server	SSP1 OPS1
	External Authentication Perimeter Server	SSP1 EA PS1
External Authentication Server	External Authentication Server Name	External Authentication Server 1
	Alternate External Authentication Server 1	External Authentication Server 2
FTP Netmap	Netmap Name	FTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI1
	Alternate Outbound Node Name	SI2
FTP Adapter 2	Adapter Name	FTPAdapter2
	Inbound Perimeter Server	SSP1 IPS2
	Outbound Perimeter Server	SSP1 OPS2
	External Authentication Perimeter Server	SSP1 EA PS2
FTP Netmap	Netmap Name	FTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI 2
	Alternate Outbound Node (SI2)	SI 1
External Authentication Server	External Authentication Server Name	External Authentication Server 1
	Alternate External Authentication Server 1	External Authentication Server 2

SSP Engine 2 Configuration for FTP

Configure the following components for Engine 2:

Component	Field to Define	Value
FTP Adapter 3	Adapter Name	FTPAdapter3

Component	Field to Define	Value
	Inbound Perimeter Server	SSP2 IPS1
	Outbound Perimeter Server	SSP2 OPS1
	External Authentication Perimeter Server	SSP2 EA PS1
External Authentication Server	External Authentication Server Name	External Authentication Server 2
FTP Netmap	Netmap Name	FTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI 2
	Alternate Outbound Node (SI 2)	SI 1
FTP Adapter 4	Adapter Name	FTPAdapter4
	Inbound Perimeter Server	SSP2 IPS2
	Outbound Perimeter Server	SSP2 OPS2
	External Authentication Perimeter Server	SSP2 EA PS2
External Authentication Server	External Authentication Server Name	External Authentication Server 1
FTP Netmap	Netmap Name	FTPNetmap2
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI1
	Alternate Outbound Node Name	SI2

Configure Failover Support for an Connect:Direct Environment

To configure failover support for a Connect:Direct adapter, install and create two SSP engines, two inbound perimeter servers, two outbound perimeter servers, two External Authentication servers, and a perimeter server to manage each EA server. Then configure the components for each SSP engine as described in the table below. To configure failover support, define the following components in SSP.

SSP Engine 1 Configuration for Connect:Direct

Configure the following components for Engine 1 in a Connect:Direct environment:

Component	Field to Define	Value
Connect Adapter 1	Adapter Name	ConnectAdapter1
	Http Ping Response	pingResponse

Component	Field to Define	Value
	Inbound Perimeter Server	SSP1 IPS1
	Outbound Perimeter Server	SSP1 OPS1
	External Authentication Perimeter Server	SSP1 EA PS1
External Authentication Server	External Authentication Server Name	External Authentication Server 1
	Alternate External Authentication Server 1	External Authentication Server 2
Connect Netmap	Netmap Name	ConnectNetmap1
	Node Name	TPNode1
	Policy	PolicyEA
	Node Name	ConnectServer1
	Destination Service Name	EAServer1
	Alternate Destinations	Connect2
Connect Adapter 2	Adapter Name	ConnectAdapter2
	Http Ping Response	pingResponse
	External Authentication Server	External Authentication Server 2
	Inbound Perimeter Server	SSP1 IPS2
	Outbound Perimeter Server	SSP1 OPS2
	External Authentication Perimeter Server	SSP1 EA PS2
Connect Netmap	Netmap Name	ConnectNetmap1
	Node Name	TPNode1
	Destination Service Name	EAServer1
	Policy	PolicyEA
	Node Name	Connect2
External Authentication Server	External Authentication Server Name	External Authentication Server 1
	Alternate External Authentication Server 1	External Authentication Server 2

SSP Engine 2 Configuration for Connect:Direct

Configure the following components for Engine 2:

Component	Field to Define	Value
Connect Adapter 3	Adapter Name	ConnectAdapter3
	Http Ping Response	pingResponse

Component	Field to Define	Value
	Inbound Perimeter Server	SSP2 IPS1
	Outbound Perimeter Server	SSP2 OPS1
	External Authentication Perimeter Server	SSP2 EA PS1
External Authentication Server	External Authentication Server Name	ExternalAuthenticationServer2
	Alternate External Authentication Server 1	ExternalAuthenticationServer1
Connect Netmap	Netmap Name	ConnectNetmap1
	Node Name	TPNode1
	Policy	PolicyEA
	Node Name	Connect 2
Connect Adapter 4	Adapter Name	ConnectAdapter4
	Http Ping Response	pingResponse
	Inbound Perimeter Server	SSP2 IPS2
	Outbound Perimeter Server	SSP2 OPS2
	External Authentication Perimeter Server	SSP2 EA PS2
External Authentication Server	External Authentication Server Name	ExternalAuthenticationServer2
	Alternate External Authentication Server 1	ExternalAuthenticationServer1
Connect Netmap	Netmap Name	ConnectNetmap1
	Node Name	TPNode1
	Policy	PolicyEA
	Node Name	FileGateway1

Configure Failover Support for an SFTP Environment

To configure failover support for an SFTP adapter, install and create two SSP engines, two inbound perimeter servers, two outbound perimeter servers, two External Authentication servers, and a perimeter server to manage each EA server. Then configure the components each SSP engine as described in the table below.

SFTP Engine 1 Configuration for SFTP

Configure the following components for Engine 1:

Component	Field to Define	Value
SFTP Adapter 1	Adapter Name	SFTPAdapter1

Component	Field to Define	Value
	Inbound Perimeter Server	SSP1 IPS1
	Outbound Perimeter Server	SSP1 OPS1
	External Authentication Perimeter Server	SSP1 EA PS1
External Authentication Server	External Authentication Server Name	External Authentication Server 1
	Alternate External Authentication Server 1	External Authentication Server 2
SFTP Netmap	Netmap Name	SFTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI 1
	Alternate Outbound Node Name	SI 2
	Alternate External Authentication Server 1	External Authentication Server 2
SFTP Adapter 2	Adapter Name	SFTPAdapter2
	Pre-Authentication Banner Text	Maverick_SSHD
	Inbound Perimeter Server	SSP1 IPS2
	Outbound Perimeter Server	SSP1 OPS2
	External Authentication Perimeter Server	SSP1 EA PS2
SFTP Netmap	Netmap Name	SFTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI 2
	Alternate Outbound Node (SI2)	SI 1
External Authentication Server	External Authentication Server Name	External Authentication Server 1
	Alternate External Authentication Server 1	External Authentication Server 2

SSP Engine 2 Configuration for SFTP

Configure the following components for Engine 2:

Component	Field to Define	Value
SFTP Adapter 3	Adapter Name	SFTPAdapter3
	Pre-Authentication Banner Text	Maverick_SSHD
	Inbound Perimeter Server	SSP2 IPS1

Component	Field to Define	Value
	Outbound Perimeter Server	SSP2 OPS1
	External Authentication Perimeter Server	SSP2 EA PS1
EA Server	External Authentication Server Name	External Authentication Server 2
STTP Netmap	Netmap Name	SFTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI 2
	Alternate Outbound Node (SI 2)	SI 1
SFTP Adapter 4	Adapter Name	SFTPAdapter4
	Inbound Perimeter Server	SSP2 IPS2
	Outbound Perimeter Server	SSP2 OPS2
	External Authentication Perimeter Server	SSP2 EA PS2
EA Server	External Authentication Server Name	External Authentication Server 1
SFTP Netmap	Netmap Name	SFTPNetmap1
	Inbound Node Name	TPNode1
	Policy	PolicyEA
	Outbound Node Name	SI1
	Alternate Outbound Node Name	SI2

Configure Advanced Adapter Properties for Failover Support

Default failover properties are defined for failover support.

Failover Support Properties

Modify the default settings for one or more of the following reasons:

- Enable failover detection
- Modify the polling frequency
- Modify how long a connection can be tried, before the connection fails
- Change the name of the profile sent to EA to request user authentication
- Enable the debug log for failover
- Prevent load balance ping requests from being writing to the debug log

Change Failover Support Properties

To change a failover support property setting:

1. From the SSP menu, selecting **Configuration**.

2. Expand the Adapters selection in the navigation panel on the left.
3. Highlight the Adapter to modify.
4. Click the **Properties** tab.
5. Modify one or more of the following properties:
 - ◆ failover.detection.mode—Determines the mode used to poll the EA server and outbound nodes. Set this property to continuous to poll the EA server and outbound nodes at the same interval defined in the outbound and EA perimeter servers. Set the property to standard to detect that outbound or EA nodes are down only when a connection is attempted. Default=standard.
 - ◆ failover.detection.enabled—Enables failover detection. Set this property to true to enable failover detection. Default=false.
 - ◆ failover.poll.interval—To configure polling frequency, in seconds. Default=5.
 - ◆ failover.conn.timeout—To identify how much time is allowed to make a connection, before the connection fails. Default=15.
 - ◆ failover.ea.ping.profile —Name of the profile sent to EA to detect if LDAP is available. By default, a profile called sspDUMMYprofile is sent. Change this property to use an actual profile name to extend healthcheck to the LDAP server. Define a profile with this name in EA.
 - ◆ failover.debug—To enable debug logging for failover. By default, debug logging is disabled. To enable debug logging for failover, set this property to true. Output is written to the file called failover.log in the /logs directory.
 - ◆ load.balancer.addr—Internal IP of load balancer. When this property is defined, all log messages generated by inbound traffic for this address are suppressed.
6. Click **Save**.

Manage Certificates for SSL/TLS Transactions with Trading Partners

This section describes how to use certificates when implementing HTTP/S, FTP/S, or Connect:Direct Secure+ Option communications between SSP and your trading partner and target servers.

Topics include:

- About Certificates
- Certificate Implementation Models Using SSP
- Import a Public Certificate into a Trusted Certificate Store
- Create a New Trusted Certificate Store
- Create a New System Certificate Store

About Certificates

Certificates are used in secure communications to encrypt and decrypt data. You create certificates using certificate creation software such as Sterling Commerce Certificate Wizard. Each certificate is made up of two components: the public key and the private key. Always keep your private key secret.

As an added measure of security, you can obtain your certificate from a certificate authority (CA). A CA verifies all of the identity information in your certificate, then adds its signature. In an SSL or TLS transaction, your certificate is presented to your trading partner, who can recognize the signature of the CA using the CA root certificate. This assures your trading partner that you are who you say you are. There are many free and commercial certificate authorities. Some companies use an internal certificate authority.

If you use a certificate that is not validated by a CA, it is called a self-signed certificate. Self-signed certificates are used when identity verification is not required, such as internal communications or product testing.

To implement SSL or TLS over FTP or HTTP when using a CA, you need to acquire the CA root certificate from the trading partner, and you must make it available to SSP. You must also make your private key and certificate available to SSP.

To implement SSL or TLS over FTP or HTTP using self-signed certificates, provide your certificates to your trading partner. Also, acquire your trading partner certificates and make them available to SSP. You also make the private key available to SSP.

Public certificates and CA root certificates must be in base 64 or DER format. Private keys, accompanied by their matching public certificates, must be contained in a base 64 key certificate or a PKCS12 file.

Certificate Implementation Models Using SSP

The following sections topics present several models for using certificates and shows how to implement the model in SSP.

- Implement Certificates that Use a Common Certificate Authority

Implement Self-Signed Certificates

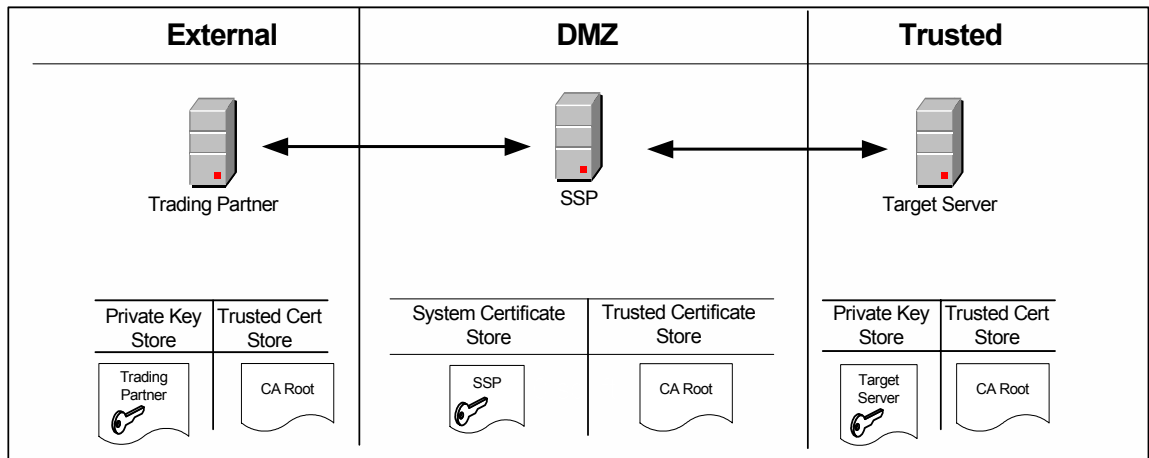
Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections

Configure a Secure Connection to Sterling External Authentication Server (EA)

Use Multiple Key Stores in SSP

Implement Certificates that Use a Common Certificate Authority

In this scenario, SSP, the target server, and the trading partner use the same CA. The certificate distribution looks like this:



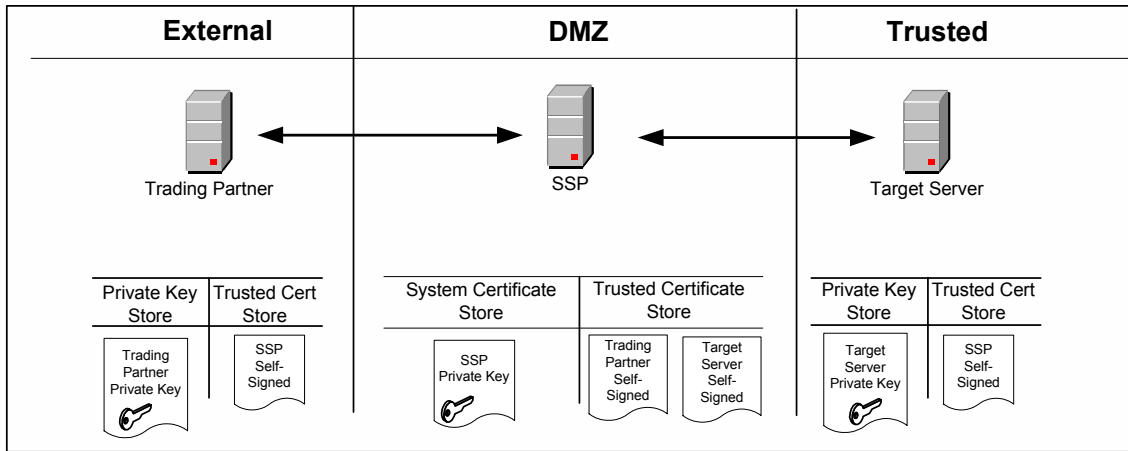
SSP has its private key and the root certificate from the CA. The trading partner has its private key and the root certificate from the CA. The target server has its private key and the root certificate from the CA.

Use the following procedure to implement this model in SSP:

1. Acquire the root certificate from the common CA.
2. Import the CA root certificate into the default store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477.
3. Import the SSP private key into the default system certificate store. Refer to *Import Private Keys into a System Certificate Store* on page 477.

Implement Self-Signed Certificates

In this scenario, there are no CA certificates. Self-signed certificates are used by all entities. The certificate distribution looks like this:



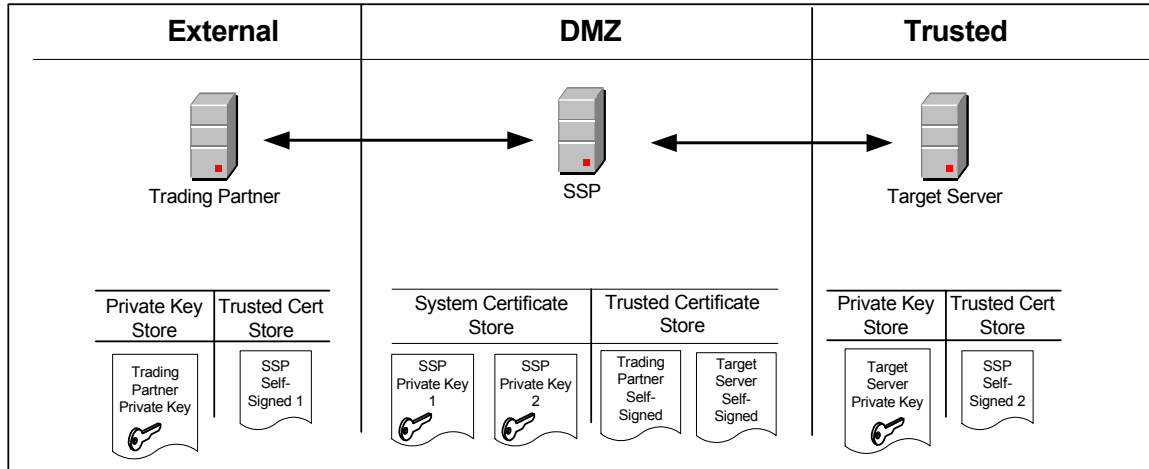
SSP has its private key and the self-signed certificates from the trading partner and the target server. The trading partner has its private key and the self-signed certificate of SSP. The target server has its private key and the self-signed certificate of SSP.

Use the following procedure to implement this model:

1. Provide the SSP self-signed certificate to your trading partner and your target server.
2. Acquire the self-signed certificates from the trading partner and target server.
3. Import the trading partner and target server self-signed certificates into the default Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477.
4. Import the SSP private key into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store* on page 477.

Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections

In this scenario, there are no CA certificates. Separate self-signed certificates are used for the inbound and outbound connections. The certificate distribution looks like this:



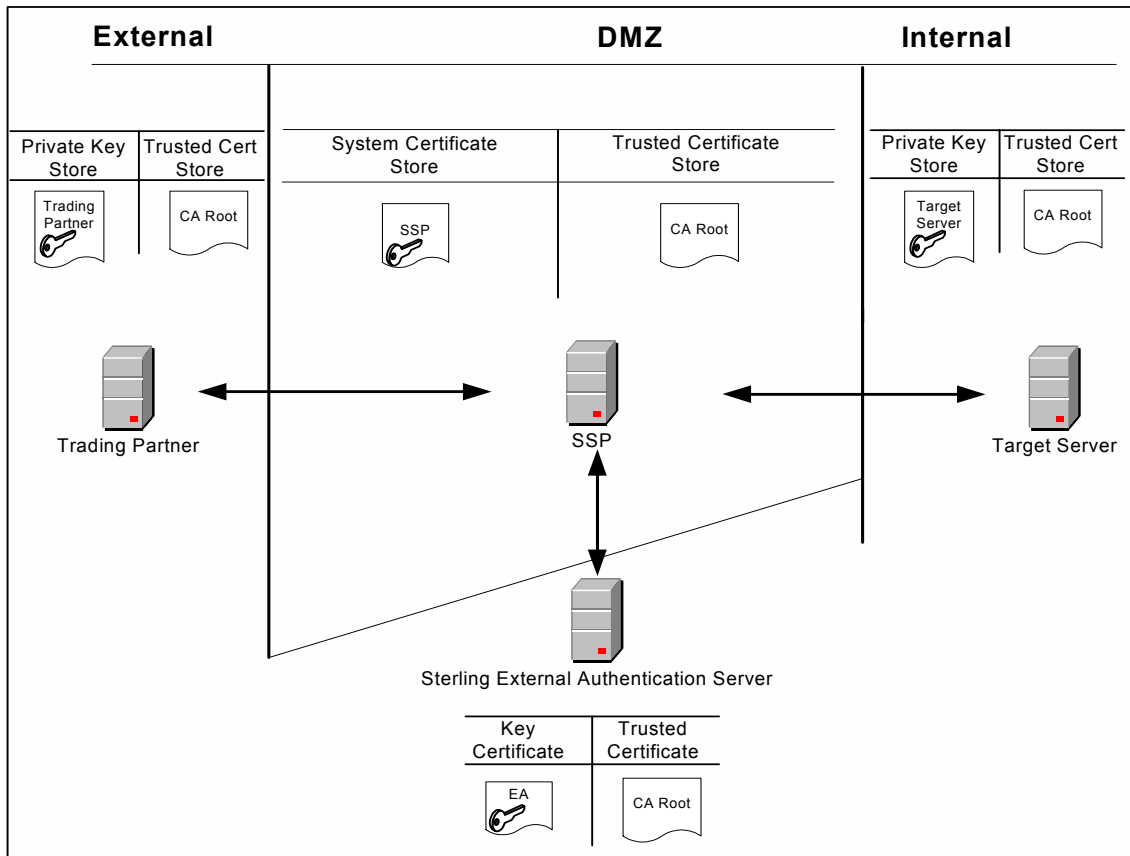
SSP has two private keys and the self-signed certificates from the trading partner and the target server. The trading partner has its private key and one self-signed certificate from SSP. The target server has its private key and the other self-signed certificate from SSP.

Use the following procedure to implement this model:

1. Provide the SSP self-signed certificate to your trading partner and your target server.
2. Acquire the self-signed certificates from the trading partner and target server.
3. Import the trading partner and target server self-signed certificates into the default Trusted Certificate Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477.
4. Import the SSP private keys into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store* on page 477.

Configure a Secure Connection to Sterling External Authentication Server (EA)

You can configure a secure connection between SSP and Sterling External Authentication Server (EA) as shown in the following diagram:



In this scenario, SSP has the private key in the system certificate store and the CA root certificate in the trusted certificate store. The trading partner has a private key and the CA root certificate. The target server has a private key and the CA root certificate. EA has a private key in its own key certificate store and the CA root certificate. Use the following procedure to implement this model.

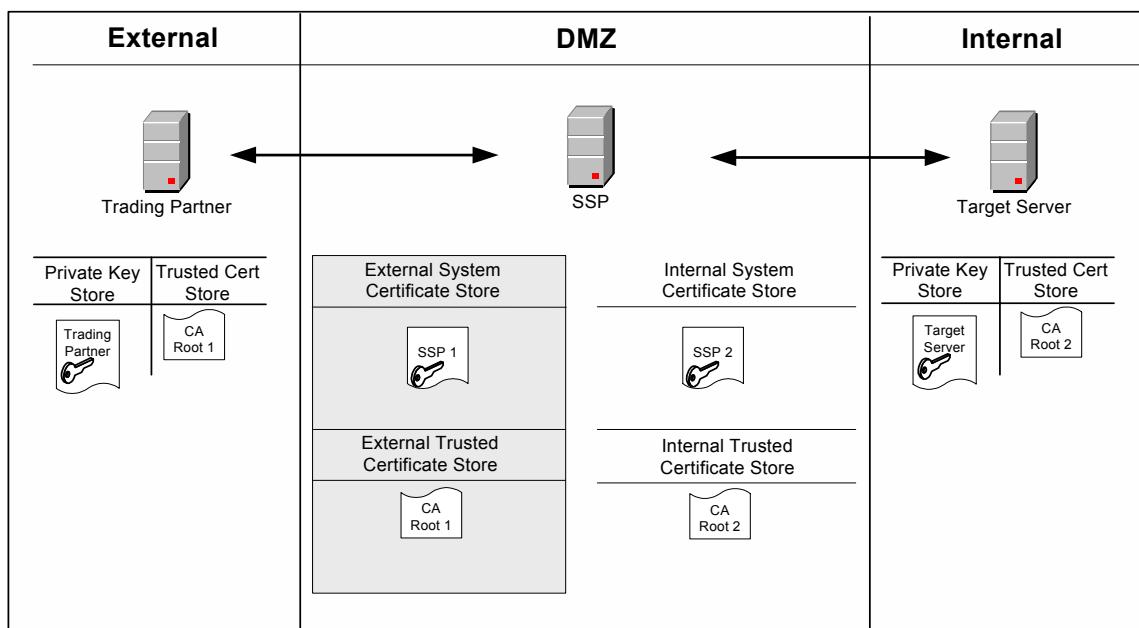
This example shows the EA implementation with a single certificate. You can also use a multiple SSP certificates model.

Use the following procedure to implement this model:

1. Acquire the root certificate from the common CA.
2. Import the CA root certificate into the default Trusted Certificate Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477.
3. Import the SSP private key into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store* on page 477.
4. Configure certificates for use by EA. Refer to the Sterling External Authentication Server documentation library for instructions.

Use Multiple Key Stores in SSP

SSP gives you the option of having multiple key stores or trust stores. This is useful if you do not want all your keys in a single location. Also, if you are running multiple SSP engines, it may be better to have a separate system certificate store or trusted certificate store for each engine. The following diagram shows a very basic model using multiple key stores:



In this scenario, SSP has two key certificates: SSP1 in the external system certificate store and SSP2 in the internal system certificate store. Different CAs are used for internal and external communications. The CA root certificate for external communication (CA Root 1) is in the external trusted certificate store. The CA root certificate for internal communication (CA Root 2) is in the internal trusted certificate store. The trading partner has its own private key and the CA Root 1 certificate. The target server has its own private key and the CA Root 2 certificate. Use the following procedure to implement this model:

1. Acquire the external CA root certificate.
2. Acquire the internal CA root certificate.
3. Create a new trusted certificate store for your external communications (External Store in diagram above). Refer to *Create a New Trusted Certificate Store* on page 477.
4. Import the external CA root certificate into the External Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477.
5. Create a new trusted certificate store for your internal communications (Internal Store in diagram above). Refer to *Create a New Trusted Certificate Store* on page 477.
6. Import the internal CA root certificate into the internal trusted certificate store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477.

7. Create a new system certificate store for external communications (External System Certificate Store in diagram above). Refer to *Create a New System Certificate Store* on page 478.
8. Import the SSP1 private key into the new external system certificate store. Refer to *Import Private Keys into a System Certificate Store* on page 477.
9. Create a new system certificate store for internal communications (Internal System Certificate Store in diagram above). Refer to *Create a New System Certificate Store* on page 478.
10. Import the SSP2 private key into the new internal system certificate store. Refer to *Import Private Keys into a System Certificate Store* on page 477.

Import a Public Certificate into a Trusted Certificate Store

Use the following procedure to import the public certificate from your trading partner, target server, or CA into a trusted certificate store:

1. Click Credentials from the menu bar.
2. Expand the Certificate Stores tree and then the Trusted Certificates Stores tree.
3. Select a trust store. The default trust store is `dfltTrustStore`.
4. Click New.
5. Specify a name in the Trusted Certificate Name field.
6. Click Browse to select the certificate to import.
7. Double-click the certificate to select.
8. Click OK.

Import Private Keys into a System Certificate Store

Use the following procedure to import an SSP private key into a system certificate store:

1. Click Credentials from the menu bar.
2. Expand the Certificate Stores tree and then the System Certificate Stores tree.
3. Select a key store. The default key store is `dfltKeyStore`.
4. Click New.
5. Specify values for the following:
 - ◆ System Certificate Name
 - ◆ Password (passphrase associated with the system certificate)
 - ◆ Confirm Password
6. Click Browse and select the certificate to import.
7. Click OK.

Create a New Trusted Certificate Store

Use the following procedure to create a new trusted certificate store:

1. Click Credentials from the menu bar.

2. Click Actions > New Certificate Store > Trusted Certificate Store.
3. Specify a name for the certificate store in the Trusted Certificate Store Name field.
4. Click Save.

Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 477 to add certificates.

Create a New System Certificate Store

To create a new system certificate store:

1. Click Credentials from the menu bar.
2. Click Actions > New Certificate Store > System Certificate Store.
3. Specify a name for the certificate store in the System Certificate Store Name field.
4. Click Save.

Refer to *Import Private Keys into a System Certificate Store* on page 477 to add certificates to the certificate store.

Start and Stop remote perimeter Servers

Use the procedures in this section chapter to start and stop a remote perimeter server. Topics include:

- Start a Perimeter Server on UNIX or Linux
- Stop a Perimeter Server on UNIX or Linux
- Start Perimeter Servers in a Windows Environment
- Stop a Perimeter Server on Windows

Start a Perimeter Server on UNIX or Linux

To start a perimeter server on UNIX or Linux:

1. Change the directory to `/install_dir/bin` where `install_dir` is the directory where the perimeter server is installed.
2. Type `startupPs.sh` and press **Enter**.

Stop a Perimeter Server on UNIX or Linux

To stop a perimeter server:

1. Change the directory to `/install_dir/bin` where `install_dir` is the directory where the perimeter server is installed.
2. Type `stopPs.sh` and press **Enter**.

Start Perimeter Servers in a Windows Environment

To start a perimeter server:

1. Change to the installation directory where the perimeter server is installed.
2. Type `startPSService.cmd` to start the perimeter server.

Stop a Perimeter Server on Windows

The remote perimeter server is installed as a Windows service. You can stop the remote perimeter server using the Windows service option or you can stop the perimeter server from the command line.

To stop a perimeter server on Windows from the command line:

1. Change the directory to `install_dir\bin` where `install_dir` is the directory where the perimeter server is installed.
2. Type `stopPSService.cmd`.

Prepare for Production

After you configure SSP and test to ensure that connections are working, you are ready to move to a production environment. This section describes production considerations.

Configure SSP to Interface with a Load Balancer

If you configure a Connect:Direct or HTTP environment, you can define an HTTP ping response to perform a health check, such as when using a load balancer tool. If you define these options, you can create a configuration for a BigIP connection and perform different levels of security checks.

Following are some possible scenarios for configuring tools like BigIP to monitor the status of the HTTP or Connect:Direct proxy adapter. The scenarios are presented in increasing order of security.

- ◆ **Simple health check**—In this scenario the monitoring agent makes a TCP connection to the listening port of the adapter and immediately disconnects. A successful connection indicates that the adapter is running. This health check has the least effect on performance.
- ◆ **Medium health check**—The monitoring agent sends an HTTP GET request with a specific URI. If the information matches the ping URI specified in the HTTP Reverse Proxy adapter, the adapter responds with the configured ping response. This allows the monitoring agent to determine that the adapter is alive and responsive.
- ◆ **Comprehensive health check**—The request from the monitoring agent is sent all the way to the SI HTTP server via the HTTP proxy adapter. To allow this connection, either the URI of the GET request sent by the monitoring agent should not match the ping URI specified in the adapter configuration, or the ping URI in the adapter configuration should be empty. In either case, the adapter passes the request to the SI server or to another SSP in the chain, depending upon the configuration. It is the responsibility of the monitoring agent and the backend server to ensure that the ping URI and response match.

Modify the Node-Level TCP Timeout Value in a Connect:Direct Node

TCP timeout identifies the maximum number of seconds SSP waits for a TCP buffer when communicating with a Connect:Direct node. For inbound sessions, this field is used after the first buffer is received from the remote node and the connecting node is identified. For outbound sessions from the proxy, this field is used from the start of the session. The default value is 90 seconds. Use this procedure to modify the TCP timeout value.

To modify the TCP timeout value in a Connect:Direct node:

1. From the Configuration navigation panel, click Netmap to expand the list of available netmaps.
2. Click the netmap where the node is defined.
3. Click the radio button beside the node you want to modify and click Edit.
4. Click the Advanced tab.

5. Change the value in the TCP timeout field.
6. Click OK.
7. Click Save.

Manage Your SSP Configuration

After you set up your SSP configuration, use CM to edit and manage the definitions you created.

Topics include:

- Change the Logging Level for a Connect:Direct Node
- Change the Logging Level for an Inbound Node
- Change the Logging Level for an Outbound Node
- Change the Logging Level for a Local Perimeter Server
- Modify Properties in an Adapter Definition
- Copy and Delete Engines, Adapters, Netmaps, Nodes, and Policies
- Change the User Store Associated With an Engine
- Filter a Node List

Change the Logging Level for a Connect:Direct Node

When you configure a Connect:Direct node, the logging level is set to None and no log is created.

To change the logging level for a Connect:Direct node so that a log is created:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select the netmap that contains the node to modify.
3. Select a node and click Edit.
4. Click the Advanced tab.
5. Select the logging level in the Logging level field.
6. Click Save.

Change the Logging Level for an Inbound Node

When you configure an inbound node for the HTTP, FTP, or SFTP protocol, the logging level for the node is set to None and no log is created for the node.

To change the logging level for an inbound HTTP, FTP, or SFTP node:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and select the netmap where the inbound node to modify is defined.
3. Select the inbound node to modify and click Edit.
4. Click the Advanced tab.
5. Select the logging level in the Logging level field.
6. Click Save.

Change the Logging Level for an Outbound Node

When you configure an outbound node for the HTTP, FTP, or SFTP protocol, the logging level is set to None and no log is created for the node.

To change the logging level for an outbound HTTP, FTP, or SFTP node:

1. From the Configuration navigation panel, click Netmap to expand the list of netmaps.
2. Click the netmap where the outbound node to modify is defined.
3. Click the Outbound Node tab.
4. Select an outbound node to modify and click Edit.
5. Click the Advanced tab.
6. Select the logging level in the Logging level field.
7. Click Save.

Change the Logging Level for a Local Perimeter Server

When you configure an engine, the logging level for the local perimeter server is set to Error by default. Error logging level writes all error messages for the local perimeter server to the log.

To change the logging level for a local perimeter server:

1. If necessary, click Configuration from the menu bar.
2. Expand the Engines tree and click the engine to modify.
3. Click the Advanced tab.
4. Select the logging level in the Local Perimeter Server Logging Level field.
5. Click Save.

Modify Properties in an Adapter Definition

Adapters are configured with default settings. Use this procedure to modify a property. For FTP and HTTP adapters, the properties and default values are displayed. To change a property, type a new value for the property key. For SFTP and Connect:Direct adapters, the properties are not displayed. Refer to the field level help for a description of the properties. To change a property, type the property name and its key value.

To modify an adapter property:

1. Click Configuration from the menu bar.
2. Expand the Adapters tree and click the adapter to modify.
3. Click the Properties tab.
4. Click New to add a new property definition.
5. For each property, specify values for the following:
 - ◆ Key
 - ◆ Value
6. Click Save.

Copy and Delete Engines, Adapters, Netmaps, Nodes, and Policies

After you create an engine, adapter, netmap, node, or policy, you can copy or delete it as necessary. For nodes, you can filter the list to view only those nodes that meet your requirements. Use the following procedures to copy or delete an engine, adapter, netmap, node, or policy:

Copy an Engine, Adapter, Netmap, or Policy

Copy a Node

Copy a Connect:Direct Node

Delete an Engine, Adapter, Netmap, or Policy

Delete an Inbound Node or Outbound Node

Delete a Connect:Direct Node

Copy an Engine, Adapter, Netmap, or Policy

To quickly create an adapter, netmap, or policy, you can copy an existing definition and make the changes necessary to create a new item.

To copy a configured engine, adapter, netmap, or policy:

1. Click Configuration from the menu bar.
2. Expand an Engine, Adapter, Netmap, or Policy tree and click the item to copy.
3. Select Actions > Copy Selected.

A new item is created and renamed to *CopyofItemName* where *ItemName* is the name of the original item you created.

4. Modify the item as necessary.
5. Click Save.

Copy a Node

To quickly create an inbound or outbound node definition, you can copy an existing definition and make the changes necessary to create a new one.

To copy a node:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and click the netmap where the node is defined.
3. Click the radio button beside the node to copy and click Copy.

A new node is created and renamed to *CopyofItemName* where *ItemName* is the name of the original node you created.

4. Modify the node definition as necessary.
5. Click Save.

Copy a Connect:Direct Node

To quickly create a Connect:Direct node definition, you can copy an existing definition and make the changes necessary to create a new item.

To copy a Connect:Direct node:

1. Click Configuration from the menu bar.
2. Expand the Netmap tree and click the Connect:Direct netmap where the node is defined.
3. Click the radio button beside the node to copy and click Copy.

A new node is created and renamed to *CopyofItemName* where *ItemName* is the name of the original node you created.

4. Modify the node definition as necessary.
5. Click OK.
6. Click Save.

Delete an Engine, Adapter, Netmap, or Policy

If you determine that an engine, adapter, netmap, or policy is no longer needed, you can delete it. Before you can delete the item, you must remove any references to it in other items. For example, if a netmap is associated with an adapter definition, it cannot be deleted.

To delete a configured engine, adapter, netmap, or policy:

1. Click Configuration from the menu bar.
2. Expand an Engine, Adapter, Netmap, or Policy tree and click the item to delete.
3. Select Actions > Delete Selected.
4. Click Delete.

Delete an Inbound Node or Outbound Node

If you determine that a node definition is no longer needed, you can delete it.

To delete a node:

1. Click Configuration from the menu bar.
2. Expand the Netmaps tree and click the netmap where the node is defined.
3. Select a node to delete and click Delete.
4. Click Save.

Delete a Connect:Direct Node

If you determine that a node definition is no longer needed, you can delete it.

To delete a Connect:Direct node:

1. Click Configuration from the menu bar.
2. Expand the Netmap tree and click the Connect:Direct netmap to where the node is defined.
3. Select the node to delete and click Delete.
4. Click Save.

Change the User Store Associated With an Engine

When you configure an engine, the user store associated with the engine is automatically configured to use the default user store called `defUserStore`. If you have created a user store and defined users in it, you must modify the engine definition to identify the user store.

To change the user store associated with an engine definition:

1. From the Configuration navigation panel, click Engine to expand the list of available engines.
2. Click the engine to modify.
3. Click the Advanced tab.
4. Select the user store in the User store field.
5. Click Save.

Filter a Node List

If you define a large set of inbound or outbound nodes, all of the nodes cannot be displayed on the main page. To view a subset of all available inbound nodes or outbound nodes, use the filter function. You can filter the list to display nodes that match the criteria you specify.

To filter a list:

1. Click Configuration from the menu bar.
2. Expand the Netmap tree and click the netmap to modify.
3. To filter an outbound node list:
 - a. Click the Outbound Node tab.
 - b. Type filter criteria to limit the list. For example, type `HTTP*` to view all node definitions that begin with `HTTP`.

Note: Filters are case sensitive.

4. To filter an inbound node list:
 - a. Click the Inbound Node tab.
 - b. Type filter criteria to limit the list.

Note: Filters are case sensitive.

Configure Logon Portal or Change Password Portal

This topic describes how to configure a self-service Logon Portal and Change Password Portal for external trading partners.

Configure Logon Portal

SSP provides a customizable self-service Logon Portal that allows Sterling File Gateway users to manage and change their passwords. The Logon Portal is a separately licensed feature of SSP with capabilities that include verification of the new password, password expiration notification, password will expire notification, display of password policy, and customizable welcome and logon screens. To customize the Logon Portal, you can modify the Logon Portal pages and user messages, or you can configure SSP to use an external logon portal.

To support the Logon Portal, configure the HTTP protocol in SSP for SSO. Refer to *Customize the Logon Portal* on page 371 for instructions on how to configure this feature.

Configure Change Password Portal

SSP provides a customizable self-service Change Password Portal that allows FTP, SFTP, and Connect:Direct users to manage and change their passwords from a web browser. The Change Password Portal is an optional, licensed component of SSP. Password management capabilities of the Change Password Portal include verification of the new password, password expiration notification, password will expire notification, display of password policy, and customizable screens.

To support the Connect:Direct Change Password Portal, configure SSP SSO for the Connect:Direct protocol. Refer to *Configure Change Password Portal* on page 396 for instructions on how to configure this feature.

To support the FTP Change Password Portal, configure SSP SSO for the FTP protocol. Refer to *Configure Change Password Portal* on page 421 for instructions on how to configure this feature.

To support the SFTP Change Password Portal, configure SSP SSO for the SFTP protocol. Refer to *Configure Change Password Portal* on page 444 for instructions on how to configure this feature.

Notices

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

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

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

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual

Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT,

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA__95141-1003
U.S.A.

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

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

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

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

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

This information is for planning purposes only. The information herein is subject to change before the products described become available. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are 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 2010. Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2010.

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

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: <http://www.ibm.com/legal/copytrade.shtml>.

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

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

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

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

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

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

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

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

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

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

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

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