

Sterling Connect:Direct Secure Plus for Microsoft
Windows



Implementation Guide

Version 4.6

Sterling Connect:Direct Secure Plus for Microsoft
Windows



Implementation Guide

Version 4.6

Note

Before using this information and the product it supports, read the information in "Notices" on page 99.

This edition applies to version 4.6 of IBM Sterling Connect:Direct Secure Plus for Microsoft Windows and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1995, 2011.**

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

Contents

Chapter 1. Overview 1

About Sterling Connect:Direct Secure Plus	1
Security Concepts.	1
Secure Sockets Layer Protocol (SSL) and Transport Layer Security Protocol (TLS).	2
Station-to-Station Protocol (STS)	4
Sterling Connect:Direct Secure Plus Tools	5
Plan the Sterling Connect:Direct Secure Plus Configuration	6
About Key Management for the STS Protocol	7
Override STS Functions from the COPY Statement	8
Summary of Processing Using Sterling Connect:Direct Secure Plus	9
TLS and SSL Sterling Connect:Direct Secure Plus Data Exchange.	9
STS Sterling Connect:Direct Secure Plus Data Exchange	10
Merge Sterling Connect:Direct Secure Plus Settings Using the STS Protocol.	11
IBM Sterling Connect:Direct Secure Plus for Microsoft Windows Documentation	12

Chapter 2. Set Up Sterling Connect:Direct Secure Plus. 13

Set Up Sterling Connect:Direct Secure Plus	13
Start Secure+ Admin Tool	13

Chapter 3. Prepare to Set Up Sterling Connect:Direct Secure Plus. 15

Obtain a Certificate and Generate a Key Certificate File	15
Generate a Key Certificate File for a CA Certificate	16
Generate a Key Certificate File for a Self-Signed Certificate	16
Exchange Trusted Root Files with Trading Partners	16
Populate the Sterling Connect:Direct Secure Plus Parameters File	17

Chapter 4. Configure Nodes 19

Node Configuration Overview	19
Configure the Sterling Connect:Direct Secure Plus .Local Node Record.	19
Customize Remote Node Records	21
Disable Sterling Connect:Direct Secure Plus for a Remote Node.	22
Add Certificate Information and Trusted Root Information to a Remote Node Record	23
Add Self-Signed Certificate Information to a Remote Node Record	23
Enable or Disable Client Authentication	24
Enable or Disable External Authentication for a Remote Node.	24
Identify the Cipher Suite to Use for Data Encryption	25

Define a Protocol for a Remote Node Record	25
Configure a Remote Node Record for the TLS or SSL Protocol	26
Configure a Remote Node for the STS Protocol	26
Manage STS Keys	30
Export Keys	30
Import Keys	30
Override STS Functions from the COPY Statement	31
Validate the Configuration	32
Configure External Authentication in the .SEAServer Record	33
Use the .Client Record to Prevent Non-Secure API Connections to a Sterling Connect:Direct Secure Plus-Enabled Server	33
Configure Strong Password Encryption	34
Disable Strong Password Encryption	34
Enable Strong Password Encryption	35
Reset Passwords When Strong Password Encryption Key and Password File are Out of Sync	35
Decryption Failure	36

Chapter 5. Automate Setup Using the CLI 37

Start and Set Up the Sterling Connect:Direct Secure Plus CLI	37
Use LCU Files to Encrypt Passwords for Use with the Sterling Connect:Direct Secure Plus CLI.	38
Sample Scripts	38
Manage the Parameters File	40
Update the .Local Node Record.	42
Manage Remote Node Records	45
Manage the Sterling External Authentication Server Record	54
Maintain the .Password File (Strong Password Encryption)	55
Manage STS Keys Using the CLI	55
Automatically Create STS Keys for Remote Node Records.	56

Chapter 6. Maintain Sterling Connect:Direct Secure Plus. 59

Sterling Connect:Direct Secure Plus Node List	59
View Sterling Connect:Direct Secure Plus Parameters File Information	60
View Sterling Connect:Direct Secure Plus Node Record Change History	60
Disable Sterling Connect:Direct Secure Plus.	60
Delete a Sterling Connect:Direct Secure Plus Remote Node Record	61
Resecure Sterling Connect:Direct Secure Plus Parameters and Access Files	61
Change Cipher Suites	62
Change STS Protocol Encryption Algorithms	62
Update Keys in an STS-Configured Node Record.	63

Clear Keys in an STS-Configured Node Record . . .	64	Set Up the Local and Remote Node Records for Testing	83
Chapter 7. View Statistics	65	Exchange Public Keys	84
View Statistics	65	Export Keys	84
Chapter 8. Audits	69	Import Keys	84
Sterling Connect:Direct Secure Plus Parameters File Auditing	69	Validate the Configuration	85
Sterling Connect:Direct Secure Plus Certificate Auditing	71	Exchange Data and Verify Results	85
Chapter 9. Troubleshoot Sterling Connect:Direct Secure Plus.	73	Chapter 13. Automation Scripts	87
Troubleshooting	73	Automation Scripts	87
Chapter 10. Configuration Worksheets 77	77	Automatically Add a Remote Node to a Sterling Connect:Direct Network	91
Local Node Security Feature Definition Worksheet	77	Configure Sterling Connect:Direct Secure Plus to Use the SSL or TLS Protocol.	94
Remote Node Security Feature Definition Worksheet	77	Chapter 14. Use LCU to Configure Encrypted Passwords	97
Chapter 11. Certificate File Layout	79	Configure Encrypted Passwords Using the LCU	97
Certificate File Layout	79	Create an LCU File	98
Chapter 12. Test Sterling Connect:Direct Secure Plus with STS	83	Notices	99

Chapter 1. Overview

About Sterling Connect:Direct Secure Plus

IBM® Sterling Connect:Direct® Secure Plus for Microsoft Windows provides enhanced security for Sterling Connect:Direct. It is available as a separate component. Sterling Connect:Direct Secure Plus uses cryptography to secure data during transmission. You select the security protocol to use.

Security Concepts

Cryptography is the science of keeping messages private. A cryptographic system uses encryption keys between two trusted communication partners. These keys encrypt and decrypt information so that the information is known only to those who have the keys.

There are two kinds of cryptographic systems: symmetric-key and asymmetric-key. Symmetric-key (or secret-key) systems use the same secret key to encrypt and decrypt a message. Asymmetric-key (or public-key) systems use one key (public) to encrypt a message and a different key (private) to decrypt it. Symmetric-key systems are simpler and faster, but two parties must somehow exchange the key in a secure way because if the secret key is discovered by outside parties, security is compromised. Asymmetric-key systems, commonly known as public-key systems, avoid this problem because the public key may be freely exchanged, but the private key is never transmitted.

Cryptography provides information security as follows:

- Authentication verifies that the entity on the other end of a communications link is the intended recipient of a transmission.
- Non-repudiation provides undeniable proof of origin of transmitted data.
- Data integrity ensures that information is not altered during transmission.
- Data confidentiality ensures that data remains private during transmission.

Sterling Connect:Direct Secure Plus enables you to select one of three security protocols to use to secure data during electronic transmission: Transport Layer Security (TLS), Secure Sockets Layer protocol (SSL) or Station-to-Station protocol (STS). Depending on the security needs of your environment, you can also validate certificates using the IBM Sterling External Authentication Server application.

Sterling Connect:Direct Secure Plus provides alternative cryptographic solutions depending upon the protocol enabled. The protocols available depend upon the version of Sterling Connect:Direct Secure Plus installed. The following table identifies the protocols available in each version of Sterling Connect:Direct Secure Plus and the encryption algorithms available for each protocol:

Sterling Connect:Direct Secure Plus	Protocol	Encryption Algorithms					
		ECC	IDEA	RC4	DES	Triple DES	AES

Sterling Connect:Direct Secure Plus	Protocol	Encryption Algorithms					
3.0	STS	x	x	x	x	x	
	SSL			x	x	x	
	TLS				x	x	
3.1 and 3.2	STS	x	x	x	x	x	x
	SSL			x	x	x	
	TLS				x	x	
4.4 and later	STS	x	x	x	x	x	x
	SSL			x	x	x	x
	TLS				x	x	

Secure Sockets Layer Protocol (SSL) and Transport Layer Security Protocol (TLS)

The SSL and the TLS protocols use certificates to exchange a key between the node that initiates the data transfer (the primary node, or PNODE) and the node that receives the data (the secondary node, or the SNODE). A certificate is an electronic document that associates a public key with an individual or other entity. It enables you to verify the claim that a given public key belongs to a given entity. A certificate authority (CA) is the entity responsible for issuing and revoking these certificates. The CA validates an applicant's identity, creates a certificate, and then signs the certificate, thus vouching for an entity's identity.

The SSL and TLS protocols provide three levels of security:

- The first level of security is activated when a trading partner connects to a Sterling Connect:Direct server. After the initial handshake, the Sterling Connect:Direct server sends its digital certificate to the trading partner. The trading partner checks that it has not expired and that it has been issued by a certificate authority the trading partner trusts. The trading partner must have a trusted root file that identifies the certificate authority.
If the security fails on any one of these checks, the trading partner is notified that the site is not secure and the connection fails.
- The second level of security, called client authentication, requires that the trading partner send its own certificate. If enabled, the Sterling Connect:Direct server requests certificate information from the trading partner, after it returns its certificate information. If the client certificate is signed by a trusted source, the connection is established.
In order to perform this security check, the trading partner must have a key certificate file available at its site and the Sterling Connect:Direct server must have a trusted root file that validates the identity of the Certificate Authority (CA) who issued the key certificate.
- The third level of security requires that a certificate common name be verified. The Sterling Connect:Direct Secure Plus server searches the certificate file it receives from the trading partner and looks for a matching certificate common name. If the server cannot find the certificate common name, communication fails.

To communicate using the SSL or TLS protocol, you must have both an X.509 certificate and a private key. The SSL and TLS protocols provide data security in the following areas:

- Strong authentication—Because the CA went through an established procedure to validate an applicant's identity, users who trust the CA can be sure the key is held by the owner. The CA prevents impersonation, and provides a framework of trust in associating an entity with its public and private keys.
- Proof of data origin and data integrity validation—The certificate provides proof of origin of electronic transmission and encryption validates data integrity. Encrypting the private key ensures that the data is not altered.
- Data confidentiality—Cipher suites encrypt data and ensure that the data remains confidential. Sensitive information is converted to an unreadable format (encryption) by the PNODE before being sent to the SNODE. The SNODE then converts the information back into a readable format (decryption).

Both the SSL protocol and the TLS protocol manage secure communication in a similar way. However, TLS provides a more secure method for managing authentication and exchanging messages, using the following features:

- While SSL provides keyed message authentication, TLS uses the more secure Key-Hashing for Message Authentication Code (HMAC) to ensure that a record cannot be altered during transmission over an open network such as the Internet.
- TLS defines the Enhanced Pseudorandom Function (PRF), which uses two hash algorithms to generate key data with the HMAC. Two algorithms increase security by preventing the data from being changed if only one algorithm is compromised. The data remains secure as long as the second algorithm is not compromised.
- While SSL and TLS both provide a message to each node to authenticate that the exchanged messages were not altered, TLS uses PRF and HMAC values in the message to provide a more secure authentication method.
- To provide more consistency, the TLS protocol specifies the type of certificate that must be exchanged between nodes.
- TLS provides more specific alerts about problems with a session and documents when certain alerts are sent.

The Sterling External Authentication Server application enables you to validate certificates that are passed during an SSL or TLS session. Using the Sterling External Authentication Server application, you can configure certificate chain validation, including the option to validate certificates against one or more Certificate Revocation Lists (CRLs) that are stored on an LDAP server. You can also configure the Sterling External Authentication Server application to return attributes associated with the incoming certificate, such as group information, that are stored on an LDAP server. See the *IBM Sterling External Authentication Server Release Notes* for installation information.

For more information on configuring Sterling Connect:Direct Secure Plus for external authentication, see the related links in the Help.

Related tasks:

“Enable or Disable External Authentication for a Remote Node” on page 24

“Configure External Authentication in the .SEAServer Record” on page 33

Station-to-Station Protocol (STS)

The STS protocol is a three-pass variation of the basic Diffie-Hellman protocol. It enables you to establish a shared secret key between two nodes with mutual entity authentication. Nodes are authenticated using digital signatures that sign and verify messages.

Each message is signed by the PNODE with its current authentication private key (and possibly its previous authentication private key) and verified by the SNODE using the corresponding public key of the PNODE. Each node uses two session keys to process control blocks: one for sending and the other for receiving. The encryption algorithms for control blocks and data copying functions are also determined. When strong authentication finishes successfully, control blocks are exchanged in an encrypted format for the entire session.

Providing Data Security

The STS protocol provides data security in the following areas:

- Strong authentication—The STS protocol uses a digital signature for strong authentication. After you enable this feature, control blocks are signed and verified. A digital signature uniquely authenticates the node signing an electronic document much like a human signature uniquely identifies the person signing his or her name to a physical document.
- Proof of data origin and data integrity validation—The digital signature verifies the sender of the message. The digital signature feature also provides data integrity validation. If the digital signature is verified, then an uncorrupted message was transmitted.
- Data confidentiality—The data encryption feature ensures confidentiality of the data sent in a Sterling Connect:Direct transfer. Sensitive information is converted to an unreadable format (encryption) by the PNODE before it is sent to the SNODE. The SNODE then converts the information back into a readable format (decryption). In order for the encryption/decryption process to work, each of these communicating nodes must have the public key value of the other.

Encryption Options

In a previous release, two versions of Sterling Connect:Direct Secure Plus were available for the STS protocol, based on government regulations regarding export laws. The difference in the versions is the encryption algorithms available.

The Limited Export version supports the following encryption algorithms:

- 56-bit DES using Cipher Block Chaining Mode (DESCBC56)
- 112-bit Triple DES in Cipher Block Chaining Mode (TDESCBC112)
- 128-bit IDEA in Cipher Block Chaining Mode (IDEACBC128)

The Export version supports only the 56-bit DES using Cipher Block Chaining Mode (DESCBC56) encryption algorithm. You must specify if a trading partner uses the Export version in the parameters file.

Sterling Connect:Direct Secure Plus Tools

Sterling Connect:Direct Secure Plus consists of five components: the Administration Tool (Secure+ Admin Tool), the parameters file, the access file, the Strong Password Encryption parameters file, and the Command Line Interface (Secure+ CLI). The following sections describe these components and their function.

Note: Only one instance of the Secure+ Admin Tool or the Secure+ CLI may be used at a time because they access the same configuration file. Do not open these tools at the same time or multiple copies of the same tool at the same time (two instances of Secure+ Admin or two instances of Secure+ CLI). Only the user who accessed the configuration file first will be able to save updates.

Administration Tool

The Secure+ Admin Tool enables you to configure and maintain the Sterling Connect:Direct Secure Plus environment. The Admin Tool is the only interface for creating and maintaining the Sterling Connect:Direct Secure Plus parameters file; operating system utilities and editing tools do not work.

Parameters File

The Sterling Connect:Direct Secure Plus parameters file contains information that determines the protocol and encryption method used during security-enabled Sterling Connect:Direct operations. To configure Sterling Connect:Direct Secure Plus, each site must have a parameters file that contains one local node record and at least one remote node record. The local node record defines the most commonly used security and protocol settings for the node at the site. Each remote node record defines the specific security and protocol used by a trading partner. You create a remote node record in the Sterling Connect:Direct Secure Plus parameters file for each Sterling Connect:Direct node that you communicate with.

At installation, a record named `.SEAServer` is created in the parameters file, which enables Sterling Connect:Direct Secure Plus to interface with Sterling External Authentication Server during SSL/TLS sessions. External authentication is configured in this record and enabled/disabled in the local and remote node records.

For additional security, the parameters file is stored in an encrypted format. The information used for encrypting and decrypting the parameters file (and private keys) is stored in the Sterling Connect:Direct Secure Plus access file.

Access File

The Sterling Connect:Direct Secure Plus access file is generated automatically when you create the Sterling Connect:Direct Secure Plus parameters file for the first time. You type a passphrase when you first initialize Sterling Connect:Direct Secure Plus. This passphrase encrypts and decrypts the private keys in the Sterling Connect:Direct Secure Plus parameters file. Your administrator must secure the access file. This file can be secured with any available file access restriction tools. Availability of the access file to unauthorized personnel can compromise the security of data exchange.

Strong Password Encryption Parameters File

Strong password encryption protects Sterling Connect:Direct passwords at rest on the Sterling Connect:Direct server using strong encryption. Strong password encryption parameters are stored in the parameters file (<CD installation directory>\Server\Secure+\Nodes\.Password). This feature is enabled by default.

Sterling Connect:Direct Secure Plus Command Line Interface

The Java-based Sterling Connect:Direct Secure Plus Command Line Interface (Secure+ CLI) is provided to enable you to create customized scripts that automate implementing Sterling Connect:Direct Secure Plus. Sample Microsoft Windows scripts are provided as models for your customized scripts. You can save these scripts with another name, modify them to reflect your environment, and distribute them throughout your enterprise. For more information about using the Secure+ CLI, commands and parameter descriptions, and the scripts, see “Start and Set Up the Sterling Connect:Direct Secure Plus CLI” on page 37.

Plan the Sterling Connect:Direct Secure Plus Configuration

Before you configure the Sterling Connect:Direct environment for secure operations, first plan how you will use Sterling Connect:Direct Secure Plus. Configure the Sterling Connect:Direct Secure Plus environment, based on company needs or preferences.

General Planning for Sterling Connect:Direct Secure Plus

Since all remote nodes are automatically enabled with the protocol defined in the local node record, determine which protocol will be used by most trading partners. Then configure the local node with this protocol. If a trading partner is not using the protocol defined in the local node record, you must configure the remote node record for that trading partner with the protocol.

Sterling Connect:Direct Secure Plus uses two files to initiate TLS or SSL sessions: a trusted root certificate file and a key certificate file.

Note: Sterling Connect:Direct Secure Plus does not support server gated crypto (SGC) certificates.

- The trusted root certificate file verifies the identity of trusted sources who issue certificates. To use Sterling Connect:Direct Secure Plus communication with a trading partner, exchange trusted root file with the trading partner. The trading partner must identify the trusted root file used to validate trusted sources in a certificate when it configures its Sterling Connect:Direct Secure Plus parameters file.
- A key certificate file is required at all sending locations and describes the identity of the secure node. This file includes information about the certificate being exchanged and the private key that defines the server.

When a trading partner attempts to establish communications with a Sterling Connect:Direct node, the node sends the public key certificate to the trading partner to verify its identity. The location of the key certificate file is configured in the Sterling Connect:Direct Secure Plus parameters file. The private key in the key certificate file is never sent (disclosed) by Sterling Connect:Direct.

The following list summarizes the tasks to configure Sterling Connect:Direct Secure Plus:

- Populate the parameters file at your site by importing the Sterling Connect:Direct network map. This task creates a local node record and remote node records.
- Enable the TLS or SSL protocol in the local node record. Enabling the local node record configures remote nodes to default to the settings in the local node record. To enable TLS or SSL, activate the following options:
 - Identify the trusted root certificate file that authenticates the trusted authorities.
 - Identify the key certificate file.

If you identify the trusted root file and the key certificate file to use for secure communications in the local node record, the trusted root file must define the identity of all CAs for all trading partners, and the root certificate associated with the key certificate file must include certificate and private key information for all certificates.
 - Identify a cipher suite to use to encrypt data in both the sending and receiving node. After secure communication is established, Sterling Connect:Direct Secure Plus determines what cipher has been defined at both the sending and the receiving node and uses this cipher to encrypt data before sending it. If more than one cipher is enabled, the preferences defined in the server parameters file determine the cipher suite used for the SSL protocol and the preferences defined in the client parameters file determine the cipher suite used for the TLS protocol.
- If you want to enable a second level of security, activate client authentication.
- If you want to enable common name checking, you must enable this feature in the remote node record.
- For remote nodes that are using the protocol defined in the local node record, configure the remote nodes to implement any of the following additional security features:
 - Activate client authentication.
 - Configure the remote node record of trading partners with the same cipher suites enabled by the trading partner because trading partners must use the same cipher suite to enable data encryption.
- If a trading partner uses a protocol that is different from the protocol defined in the local node record, define the protocol in the remote node record. The remote node record must identify the same protocol as that used by the trading partner. Otherwise, Sterling Connect:Direct Secure Plus fails.
- If a trading partner does not use Sterling Connect:Direct Secure Plus, disable it in that remote node record.

About Key Management for the STS Protocol

When you configure a remote node record to use the STS protocol, you generate unique authentication and signature public keys. In addition, your trading partner generates authentication and signature public keys for that node. In order to communicate with the trading partner, all four keys must be defined in the parameters file for both your configuration and the trading partner's configuration. Therefore, you and your trading partner must exchange keys.

For the initial configuration, you manually exchange keys. You export keys and send them to the trading partner. Then you import the keys you receive from the trading partner into the parameters file. After the initial exchange, you can automate the exchange of key information.

If a remote node uses the STS protocol, you must decide how often to update keys and how to manage key files received from trading partners.

Key Exchange Method

After you exchange keys with a trading partner, both partners should enable the automatic key update feature for easier key management. When automatic key update is enabled, the updated key is sent to the trading partner node during the authentication process and the remote node record is updated with the new key values. Both you and your trading partner must enable automatic key update in order to use this feature.

Key Update Frequency

Decide how frequently to update authentication and signature keys. The more frequently you update key values, the more secure your environment is. When you turn on automated key updates, you can update keys daily, because the updated keys are sent to the trading partners automatically and securely during authentication.

Import Key File Management

Before you begin exchanging key files with a trading partner, you must consider how to manage key files. Sterling Connect:Direct Secure Plus names exported key files based on the name of the target node; therefore, new key files that you receive from a trading partner have the same name as the old key file. To avoid overwriting an old key file with a new one, you manage key files in one of the following ways:

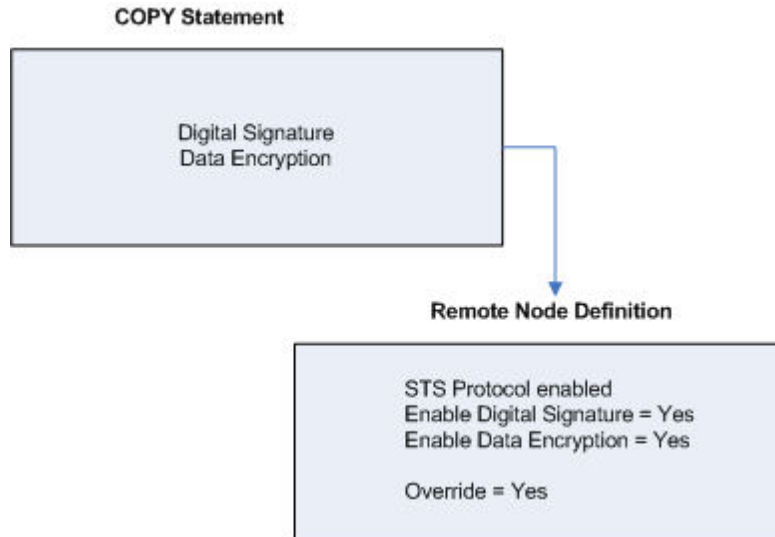
- Import the new key file immediately after receiving it from your trading partner and then delete the old key file.
- Rename the key file upon receipt or have your trading partner rename it before sending it.
- Create a directory for each remote node and store each key file separately in the associated directory for use if the configuration is lost or node records are accidentally deleted.

It is not necessary to retain key files since the files are stored in the configuration file after you import them. However, saving the key files allows you to reconfigure the parameters file if a configuration is lost or if the node record is accidentally deleted.

Override STS Functions from the COPY Statement

When you configure a node to use the STS protocol, you can use the COPY statement in Sterling Connect:Direct to override the settings in the parameters file, if override is enabled. It is not always possible to disable digital signatures and data encryption. If either node enables these options, the options are used.

The following illustration shows how the COPY statement overrides the security functions in a remote node:



Summary of Processing Using Sterling Connect:Direct Secure Plus

After you configure Sterling Connect:Direct Secure Plus, you are ready to exchange data securely with other security-enabled Sterling Connect:Direct nodes. Your node must also be defined in the parameters file of trading partners. Data is securely exchanged between two nodes using the protocol defined in the parameters file.

TLS and SSL Sterling Connect:Direct Secure Plus Data Exchange

Data exchange consists of three steps: authentication, sending data, and receiving data. The TLS or SSL protocol data exchange process is described in the following sections. The primary node initiates the data transmission, and the secondary node receives the data. The following description of processing depicts the PNODE as sending data and the SNODE as receiving data.

Authentication

The following steps occur during authentication:

1. The PNODE sends a control block to the SNODE. The SNODE confirms that it has a record defined in the Sterling Connect:Direct Secure Plus parameters file for the PNODE and determines the cipher suite to use for secure communication. If the SNODE finds a record for the PNODE and a common cipher suite can be negotiated, the session continues.
2. The SNODE sends its certificate back to the PNODE. Information for creating an encryption key is included. If client authentication is enabled, the SNODE also requests a certificate from the PNODE.
3. The PNODE verifies that the certificate of the SNODE is in its parameters file and generates a session key. If requested, it sends a client certificate to the SNODE for verification.
4. The SNODE confirms that a secure environment is established and returns a secure channel message.
5. The PNODE authenticates the SNODE and establishes communications.

Sending Customer Data

After communication is authenticated, the PNODE begins transmitting data.

- Information for encrypting data is exchanged in the control blocks.
- If data compression is enabled, the PNODE compresses the data.
- The PNODE encrypts the data with a cipher suite recognized by both communications nodes.

Receiving Customer Data

The SNODE receives the data.

- The SNODE decrypts the data using a cipher suite available for both the PNODE and the SNODE.
- If the data is compressed, the receiving node decompresses it.

STS Sterling Connect:Direct Secure Plus Data Exchange

Data exchange consists of three steps: authentication, sending data, and receiving data. The STS protocol data exchange process is described in the following sections. The primary node initiates the data exchange and the secondary node receives the data.

The following description of processing depicts the PNODE as sending data and the SNODE as receiving data.

Authentication

The following steps occur during authentication:

1. The PNODE sends a control block to the SNODE. Information for creating an encryption key for the PNODE is included. The SNODE confirms that it has a record defined in the Sterling Connect:Direct Secure Plus parameters file for the PNODE. If so, it retains the information for key encryption for processing later. If not, the session fails.
2. The SNODE sends a control block signed with its private authentication key. Information for creating an encryption key is included.
3. The PNODE verifies the signature of the SNODE using its public authentication key.
4. The PNODE returns a control block signed with its private authentication key.
5. The SNODE verifies the signature using the public authentication key of the PNODE.
6. When authentication is successful, each node generates a shared session encryption key for encrypting control blocks.

Sending Customer Data

After communication is authenticated, the PNODE begins transmitting data.

- If data encryption is enabled, information for creating an encryption key is exchanged in the control blocks.
- If digital signature is enabled, the SNODE applies the signature algorithm to the data using its private signature key to ensure that the data was sent by the PNODE and has not been altered.
- If data compression is enabled, the PNODE compresses the data.

- If data encryption is enabled, the PNODE encrypts the data with an encryption algorithm using a shared secret encryption key generated specifically for this transmission. The encryption algorithm is determined at authentication.

Receiving Customer Data

The SNODE receives the data.

- If data is encrypted, the SNODE decrypts the data using the encryption algorithm available for both the PNODE and the SNODE.
- If the data is compressed, the SNODE decompresses it.
- If digital signature is enabled, the SNODE verifies the origin and integrity of the data by applying a verification algorithm using the public digital signature key of the PNODE.

Merge Sterling Connect:Direct Secure Plus Settings Using the STS Protocol

When two nodes use the STS protocol to exchange secure data, Sterling Connect:Direct Secure Plus settings are exchanged during authentication. These settings are then merged and the resulting value for each security function is used for the Sterling Connect:Direct session. The result is based upon the values defined on the primary node (PNODE) and the secondary node (SNODE).

The following topics describe the results of these merged values based on the PNODE and SNODE values.

Digital Signature

When Sterling Connect:Direct Secure Plus settings are merged, the most secure setting from either node is used for the digital signature feature. If either node enables the digital signature feature, digital signatures are used for the session. If both nodes disable digital signatures, digital signatures are not used. The following table shows the digital signature setting after the PNODE and SNODE values are merged:

PNODE Value	SNODE Value	Merged Results
Y	Y	Y
Y	N	Y
N	Y	Y
N	N	N

Algorithm for Encrypting Control Blocks

The algorithm that encrypts Sterling Connect:Direct control blocks used for strong authentication is the first algorithm ID in the PNODE list that is also in the SNODE list. If the nodes do not share a common algorithm, authentication fails.

Auto Update Public Keys

If both nodes enable the auto update function, the authentication and signature public key values are dynamically updated during authentication if the remote node supplies different values. Enabling auto update eliminates much of the work that has to be performed by the Sterling Connect:Direct Secure Plus administrator.

Data Encryption

The most secure setting from either node is used for data encryption. If the nodes do not share a common algorithm, the copy operation fails. The following table shows the setting after the PNODE and SNODE values are merged.

PNODE Value	SNODE Value	Merged Results
N	N	N
N	Y	The first algorithm ID in the SNODE list that is in the PNODE list.
N	Algorithm ID	The SNODE algorithm ID if it is in the PNODE list.
Y	N Y algorithm ID	The first algorithm ID in the PNODE list that is in the SNODE list.
algorithm ID	N Y algorithm ID	The PNODE algorithm ID if it is in the SNODE list.

IBM Sterling Connect:Direct Secure Plus for Microsoft Windows Documentation

The *IBM Sterling Connect:Direct Secure Plus for Microsoft Windows Implementation Guide* describes how to implement point-to-point security into Sterling Connect:Direct operations with Sterling Connect:Direct Secure Plus. This document includes information to plan, install, configure, and use Sterling Connect:Direct Secure Plus.

This guide assumes knowledge of the Sterling Connect:Direct system, including its applications, network, and environment. If you are not familiar with Sterling Connect:Direct, refer to the Sterling Connect:Direct library of manuals.

Task Overview

The following table directs you to the information required to perform the tasks documented in the Sterling Connect:Direct Secure Plus documentation:

Task	For More Information See
Understanding Sterling Connect:Direct Secure Plus	"About Sterling Connect:Direct Secure Plus" on page 1
Setting up Sterling Connect:Direct Secure Plus	"Set Up Sterling Connect:Direct Secure Plus" on page 13
Automating the Set up of Sterling Connect:Direct Secure Plus	"Start and Set Up the Sterling Connect:Direct Secure Plus CLI" on page 37
Testing Sterling Connect:Direct Secure Plus for STS	"Set Up the Local and Remote Node Records for Testing" on page 83
Maintenance tasks such as viewing all nodes and their attributes	"Sterling Connect:Direct Secure Plus Node List" on page 59
Viewing Sterling Connect:Direct Secure Plus statistics	"View Statistics" on page 65
Understanding error messages and resolving errors	"Troubleshooting" on page 73

Chapter 2. Set Up Sterling Connect:Direct Secure Plus

Set Up Sterling Connect:Direct Secure Plus

Before you begin using Sterling Connect:Direct Secure Plus, you must configure nodes for secure operations.

You can install Sterling Connect:Direct Secure Plus using the Sterling Connect:Direct for Microsoft Windows installation script. For more information on installing Sterling Connect:Direct Secure Plus, see the *IBM Sterling Connect:Direct for Microsoft Windows Getting Started Guide*.

CAUTION:

After Sterling Connect:Direct Secure Plus is installed, the system administrator is responsible for securing access to the Sterling Connect:Direct Secure Plus Administration Tool, Sterling Connect:Direct Secure Plus CLI, and parameters files. The Sterling Connect:Direct Secure Plus administrator and Sterling Connect:Direct server need full access to the Sterling Connect:Direct Secure Plus directory. No other users require access.

Start Secure+ Admin Tool

About this task

Use the Secure+ Admin Tool to set up and maintain a Sterling Connect:Direct Secure Plus operation.

You can also use the Sterling Connect:Direct Secure Plus Command Line Interface (Secure+ CLI) to set up and manage Sterling Connect:Direct Secure Plus operations. See “Start and Set Up the Sterling Connect:Direct Secure Plus CLI” on page 37.

Note: The parameters file is not dynamically updated. When multiple users update the parameters file, each user must close and reopen the file to display new records added by all sources.

Procedure

1. Click **Start > All Programs**.
2. Select **IBM Sterling Connect Direct v4.6.0 > CD Secure+ Admin Tool**. The Secure+ Admin Tool Main Window is displayed.

The Secure+ Admin Tool starts and opens the Sterling Connect:Direct Secure Plus parameters file for the associated Sterling Connect:Direct node.

Chapter 3. Prepare to Set Up Sterling Connect:Direct Secure Plus

Before you configure the Sterling Connect:Direct Secure Plus environment, perform the following setup procedures.

- Complete a worksheet for the local node record and a worksheet for each trading partner for whom you plan to enable Sterling Connect:Direct Secure Plus. Use the Local Node Security Feature Definition Worksheet to record the settings you plan to enable for the local node. For each trading partner, complete a Remote Node Security Feature Definition Worksheet and record the settings to enable Sterling Connect:Direct Secure Plus for the trading partner.
- Obtain a certificate and generate a key certificate file. A certificate is created by a trusted certificate authority (CA) or you can create a self-signed certificate. Generate a key certificate file by combining the certificate file and the private key file.
- Exchange trusted root certificate files with your trading partners
- Populate the Sterling Connect:Direct Secure Plus parameters file to include a record for each node running Sterling Connect:Direct Secure Plus. To communicate with a node running Sterling Connect:Direct Secure Plus, the node must have a record in the Sterling Connect:Direct network map and the Sterling Connect:Direct Secure Plus parameters file.

Obtain a Certificate and Generate a Key Certificate File

The TLS and the SSL security protocols use a secure server RSA X.509V3 certificate to authenticate a site for any node that accesses the site. Obtain a certificate (X509 ASN.1 Base64 format) from a CA or create a self-signed certificate. Create a private key file (ASN.1/Base64 encoded PKCS8/PKCS5 v1.5 or “traditional SSLeay” format) using IBM Sterling Certificate Wizard or any Web server software.

Sterling Connect:Direct Secure Plus looks for a key certificate file to authenticate a site. This file combines information from the certificate file and the private key file.

Sterling Certificate Wizard creates the files needed to obtain a certificate and create a key certificate file. It can be used to:

- Generate a certificate signing request (CSR) that you send to the CA to request a certificate.
- Generate a self-signed certificate and act as your own CA.
- Generate a private key file. A private key file is created when you generate the CSR or the self-signed certificate.
- Create a key certificate file that combines the certificate file with the private key file.

For information on installing Sterling Certificate Wizard, refer to the *IBM Sterling Certificate Wizard Release Notes*. For information on using Sterling Certificate Wizard, refer to the Sterling Certificate Wizard online Help.

Generate a Key Certificate File for a CA Certificate

About this task

To generate a key certificate file from a certificate generated by a CA:

Note: While a key certificate may contain information about its intended use, such as e-mail, Sterling Connect:Direct Secure Plus does not use this information. It uses client or server authentication.

Procedure

1. Generate a certificate signing request (CSR) and a private key using Sterling Certificate Wizard or any Web server software.
2. Send the CSR to the CA to request a certificate.
3. When you receive the certificate from the CA, generate a key certificate file using Sterling Certificate Wizard or a text editor. The key certificate file combines information from the certificate file that you received from the CA and the private key file you generated.

Generate a Key Certificate File for a Self-Signed Certificate

About this task

You can use Sterling Certificate Wizard to generate a key certificate file for a site that is authenticated with a self-signed certificate.

When Sterling Certificate Wizard generates a self-signed certificate, it creates a private key called `privkey.txt` and a trusted root file called `cert.crt`. The key certificate file combines information from the certificate file and the private key file. Sterling Certificate Wizard creates a key certificate file called `keycert.txt`.

Procedure

1. Generate a self-signed certificate using Sterling Certificate Wizard.
2. Generate a key certificate file.
3. Copy the key certificate file to the Sterling Connect:Direct server.

Exchange Trusted Root Files with Trading Partners

About this task

When validating certificates, the trading partner must have a copy of the trusted root certificate file to verify the identity of the entity who issued your certificate, and you must have a copy of the trading partner's trusted root certificate file to validate the entity that issued the trading partner's certificate file.

Note: If the trading partner uses SSL for other secure communications, such as secure e-mail, the trading partner may already have a trusted root file for the CA used in the certificate.

To exchange trusted root files with a trading partner:

Procedure

1. Obtain a copy of the trusted root file and copy it to the Sterling Connect:Direct Secure Plus directory on the Sterling Connect:Direct server. You can maintain multiple trusted root files for each trading partner you need to verify, or you can maintain all trusted root file information in one file.
2. Do one of the following:
 - a. If you store all trusted root file information in one file, identify the location of this file in the local node record.
 - b. If you maintain a separate file for each trading partner, configure the name of each trading partner's trusted root file in the corresponding remote node record.

Populate the Sterling Connect:Direct Secure Plus Parameters File

About this task

To communicate with a trading partner using Sterling Connect:Direct Secure Plus, you define a node record for that partner in both the Sterling Connect:Direct network map and the Sterling Connect:Direct Secure Plus parameters file. To set up the Sterling Connect:Direct Secure Plus environment, you can populate the Sterling Connect:Direct Secure Plus parameters file from entries defined in an existing network map.

When you populate the parameters file from the network map, a record is automatically created in the parameters file for each node entry in the network map. Initially, Sterling Connect:Direct Secure Plus is disabled for each of the records created. You configure the local node record to activate Sterling Connect:Direct Secure Plus for all nodes in the parameters file.

Procedure

1. From the Secure+ Admin Tool Main Window, click **File > Sync with Netmap**. The **Available Netmaps** dialog box is displayed.
2. Select the netmap to open and click **Sync**. The **Select Netmap Entries to Add** dialog box is displayed.
3. Click **Add All**. The **Select Parameters File Entries to Delete** dialog box is displayed.
4. Click **Skip** to close the parameters file without deleting any entries. The Sterling Connect:Direct Secure Plus parameters file is populated and the Secure+ Admin Tool Main Window displays remote node records in the parameters file including the records you added from the network map.

Chapter 4. Configure Nodes

Node Configuration Overview

When you import the network map records into the Sterling Connect:Direct Secure Plus parameters file, Sterling Connect:Direct Secure Plus parameters are disabled.

To configure the nodes, complete the following procedures:

- Configure the Sterling Connect:Direct Secure Plus .Local node record
Define the security options for the local node. Because TLS and SSL provide the strongest authentication with easy-to-maintain keys, configure the local node for one of these protocols. Determine which protocol is used by most trading partners and configure the local node with this protocol.
- Disable remote nodes that do not use Sterling Connect:Direct Secure Plus
- Customize a remote node for the following configurations:
 - To use a unique certificate file to authenticate a trading partner
 - To use a different self-signed certificate for client or server authentication
 - To identify a unique cipher suite used by a trading partner
 - To activate common name validation
 - To activate client authentication
 - To activate external authentication
- Configure all remote nodes that use a protocol that is not defined in the local node
When you configure the local node, all remote nodes are automatically configured to the protocol defined in the local node. If a trading partner uses a different protocol, you must turn on the protocol in the remote node record. For example, if you activate the TLS protocol in the .Local node record and a trading partner uses the SSL protocol, configure the SSL protocol in the remote node record for the trading partner.
- If you want to use Sterling External Authentication Server to validate certificates:
 - Update the .SEAServer record with the Sterling External Authentication Server host name and port
 - Enable TLS or SSL
 - Enable external authentication
 - Specify the certificate validation definition to use
- If you want to prevent non-secure API connections from communicating with a Sterling Connect:Direct Secure Plus enabled server:
 - Define a remote node called .Client
 - Enable TLS or SSL
 - Disable override

Configure the Sterling Connect:Direct Secure Plus .Local Node Record

About this task

Configure the .Local node record with the protocol used by most of your trading partners. Because remote node records can use the attributes defined in the .Local

node record, defining the .Local node record with the most commonly used protocol saves time. After you define the protocol in the .Local node record, all remote nodes default to that protocol. Also, identify the trusted root file to be used to authenticate trading partners.

To configure the local node, refer to the Local Node Security Feature Definition Worksheet that you completed for the .Local node record security settings and complete the following procedure:

Procedure

1. From the Secure+ Admin Tool Main Window, double-click the .Local record. The Edit Record dialog box displays the Security Options tab, the node name, and the type of node.
2. Set the following Security Options for the local node:
 - Enable TLS Protocol or Enable SSL Protocol
 - Enable Override
3. If necessary, change the time-out value in the Authentication Timeout box. Refer to the following table for an explanation of the Security Options boxes:

Field Name	Field Definition	Valid Values
Node Name	Specifies the node record name.	.Local This is not an editable field.
Base Record	Specifies the name of the base record. If an alias record is selected, the base record name is displayed in this box.	Name of the local Sterling Connect:Direct node.
Type	Specifies the current record type.	Local for a local record and Remote for a remote record. This is not an editable field.
Disable Secure+	Disables Sterling Connect:Direct Secure Plus.	Default value is Disable Secure+. Note: If this option is selected, override is enabled, and no remote node definition exists for the remote node in the Sterling Connect:Direct Secure Plus parameters file, Sterling Connect:Direct Secure Plus is bypassed.
Enable TLS Protocol	Enables TLS protocol to ensure that data is securely transmitted.	The default value is Disable Secure+.
Enable SSL Protocol	Enables SSL protocol to ensure that data is securely transmitted.	The default value is Disable Secure+.
Enable STS Protocol	Enables STS protocol to ensure that data is securely transmitted.	The default value is Disable Secure+.
Disable Override	Disables the ability to override values in the .Local node record with values in the remote node record.	The default value is Disable Override.
Enable Override	Enables override to allow values in the remote node record to override values in the .Local node record.	The default value is Disable Override.

Field Name	Field Definition	Valid Values
Authentication Timeout	<p>Specifies maximum time, in seconds, that the system waits to receive the Sterling Connect:Direct Secure Plus blocks exchanged during the Sterling Connect:Direct Secure Plus authentication process.</p> <p>If you specify a value of 0, Sterling Connect:Direct waits indefinitely to receive the next message.</p> <p>Specify a time to prevent malicious entry from taking as much time as necessary to attack the authentication process.</p>	<p>A numeric value equal to or greater than 0, ranging from 0 to 3600.</p> <p>The default is 120 seconds.</p>

4. Click the **TLS/SSL Protocol** tab. The **TLS/SSL Options** dialog box is displayed.
5. Type the location of the trusted root certificate file to use to authenticate trading partners in the **Trusted Root Certificate File** box or click **Browse** and locate the file.
6. Click **Browse** next to the **Certificate File** box, locate the key certificate file and double-click the file to select it. The **Certificate Passphrase** dialog box is displayed.
7. Type the passphrase you specified when you created the certificate in the **Certificate Passphrase** box.
8. Click **OK**. The **Certificate File** box is populated with the certificate file name and location.
9. To enable client authentication, click **Yes** in the **Enable Client Authentication** box.
10. To view information about the trusted root file and the certificate, click **View Certificates**.
11. Highlight the cipher suites to enable in the **Available** list and click **Add**.
12. If necessary, use the Up and Down buttons to reorder the cipher suites. Place the cipher suites in order of preference.
13. Click the **External Authentication** tab. The **External Authentication** dialog box is displayed.
14. Choose one of the following options:
 - To enable external authentication on the remote node, click **Yes** in the **Enable External Authentication** box.
 - To disable external authentication on the remote node, click **No**.
15. Type the Certificate Validation Definition character string defined in Sterling External Authentication Server.
16. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

Customize Remote Node Records

After you configure the .Local node record, Sterling Connect:Direct Secure Plus enables the protocol and parameters that you configured for the local node for all remote node records. If all trading partners use the protocol and configuration defined in the .Local node record, you are now ready to begin using Sterling Connect:Direct Secure Plus.

However, even when a trading partner uses the same protocol as the one defined in the .Local node record, you may need to customize remote node records for the following configurations:

- Using a unique certificate file to authenticate a trading partner—During a TLS or SSL session, a certificate enables the PNODE to authenticate the SNODE. You identified a certificate in the .Local node record. If you want to use a unique certificate to authenticate a trading partner, you must identify this information in the remote node record.
- Using a self-signed certificate file to authenticate a trading partner—During a TLS or SSL session, a certificate enables the PNODE to authenticate the SNODE. If you want to use a self-signed certificate to authenticate a trading partner, you must identify this information in the remote node record.
- Activating client authentication—Client authentication requires that the SNODE validate the PNODE. If you want to enable client authentication, activate this feature in the remote node record. If you want another layer of security, you can activate the ability to validate the certificate common name.
- Identifying the cipher suite used by a trading partner—When configuring the TLS or SSL protocol, you enable cipher suites that are used to encrypt the transmitted data. When communicating with a trading partner, you and the trading partner must use the same cipher suite to encrypt data. If the trading partner does not enable a cipher suite that is enabled in your configuration, communication fails. If necessary, enable cipher suites in the remote node record.

If a trading partner uses a different protocol than the protocol defined in the .Local node record, you need to define the protocol in the remote node record. Refer to “Define a Protocol for a Remote Node Record” on page 25.

When customizing remote node records, refer to the Remote Node Security Feature Definition Worksheet you completed for each remote node.

Disable Sterling Connect:Direct Secure Plus for a Remote Node

About this task

If a trading partner is not using Sterling Connect:Direct Secure Plus, you must disable Sterling Connect:Direct Secure Plus for that remote node record.

To disable Sterling Connect:Direct Secure Plus for a remote node record:

Procedure

1. Open the remote node record for which you want to disable Sterling Connect:Direct Secure Plus.
2. Click the **Security Options** tab.
3. Click the **Disable Secure+ Option**.
4. Click the **Disable Override** option.
5. Click **OK** to update the node record.

Add Certificate Information and Trusted Root Information to a Remote Node Record

About this task

You identified a certificate and a trusted root file to use for authentication when you defined the .Local node record. If you want to use a different certificate to authenticate a trading partner or a different trusted root file to validate the identity of the trusted source who issues certificates, you need to add that information to the remote node record.

Complete the following procedure to identify unique certificate information or trusted root information for a remote node record:

Procedure

1. From the Secure+ Admin Tool Main Window, double-click the remote node record to open. The **Edit Record** dialog box is displayed.
2. Click the **TLS/SSL Protocol** tab.
3. Type the location of the trusted root certificate in the **Trusted Root Certificate File** box or click **Browse** and locate the file.
4. Click **Clear Certificate** to clear any associated certificate file from the remote node record.
5. Click **Browse** next to the **Certificate File** box, locate the key certificate file, and double-click the file to select it. The **Certificate Passphrase** dialog box is displayed.
6. Type the passphrase you specified when you created the certificate in the **Certificate Passphrase** box.
7. If you want to view trusted root files or key certificate files, click **View Certificates**.
8. Click **OK**.

Add Self-Signed Certificate Information to a Remote Node Record

About this task

If you want to use a self-signed certificate to authenticate a trading partner and this information has not been identified in the .Local node record, you need to add that information to the remote node record. First, obtain a copy of the trading partner's trusted root file and copy this file to a local drive. If you plan to perform server authentication, you must also send a copy of your trusted root certificate file to the trading partner.

Complete the following procedure to add self-signed certificate information to a remote node record:

Procedure

1. From the Secure+ Admin Tool Main Window, double-click the remote node record to open. The **Edit Record** dialog box is displayed.
2. Click the **TLS/SSL Protocol** tab.
3. Type the location of the trading partner's self-signed certificate file in the **Trusted Root Certificate File** box or click **Browse** and locate the file.

4. Click **OK**.

Enable or Disable Client Authentication

About this task

Client authentication provides another level of security. If you activate client authentication, two levels of certificate validation occur. First, the PNODE validates the certificate from the SNODE. Then, the SNODE performs client authentication by validating the certificate from the PNODE. Within client authentication, you can enable a third level of security. If you provide a certificate common name during setup, the client authentication process first validates the certificate from the PNODE and then searches for the common name in the certificate file. If the SNODE cannot validate the PNODE certificate or locate the common name, communication fails. You must enable common name checking in the remote node record to use this function. It cannot be activated in the .Local node record.

If client authentication is enabled in the .Local node record, client authentication will be performed for all connections. If you want to disable client authentication for select trading partners, turn off this option in the remote node record.

Complete the following procedure to activate or deactivate client authentication for a remote node:

Procedure

1. From the Secure+ Admin Tool main window, double-click the remote node record to open. The **Edit Record** dialog box is displayed.
2. Click the **TLS/SSL Protocol** tab.
3. To enable client authentication in a remote node record, click **Yes** in the **Enable Client Authentication** box.

Note: If client authentication is enabled in the .Local node record, it is automatically enabled in all remote node records.

4. To disable client authentication in the remote node record, click **No** in the **Enable Client Authentication** box.
5. To enable another level of security, type the certificate common name in the **Certificate Common Name** box.
6. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

Enable or Disable External Authentication for a Remote Node

About this task

On a node-by-node basis, you can specify whether a remote node uses external authentication or if that remote node defaults to the external authentication setting in the .Local node record.

Complete the following procedure to configure a remote node for external authentication:

Procedure

1. If necessary, open the remote node record. The **Edit Record** dialog box is displayed.

2. Click the **External Authentication** tab.
3. Choose one of the following options:
 - To enable external authentication on the remote node, click **Yes** in the **Enable External Authentication** box.
 - To disable external authentication on the remote node, click **No**.
 - To default to the external authentication setting defined in the .Local node record, click **Default to Local Node**.

Note: If external authentication is enabled in the .Local node record, it is automatically enabled in all remote node records.

4. Type the Certificate Validation Definition character string defined in Sterling External Authentication Server.
5. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

Related concepts:

“Secure Sockets Layer Protocol (SSL) and Transport Layer Security Protocol (TLS)” on page 2

Identify the Cipher Suite to Use for Data Encryption

About this task

Cipher suites encrypt the data you send to a trading partner. To encrypt data, Sterling Connect:Direct uses the cipher suite that is available for both the SNODE (trading partner) and the PNODE. If you want to specify a unique cipher suite for a trading partner, you can identify this information in the remote node record. Identifying a cipher suite ensures that the selected cipher suite is used for data encryption.

Complete the following procedure to identify the cipher suite used by a remote node.

Procedure

1. If necessary, open the remote node record. The **Edit Record** dialog box is displayed.
2. Click the **TLS/SSL Protocol** tab.
3. Turn off **Default to Local Node** from the **Cipher Suites** section.
4. Highlight the cipher suite to use in the **Available** list and click **Add**.
5. If more than one cipher suite was selected, use the Up and Down buttons to reorder the cipher suites. Place the cipher suites in order of preference.
6. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

Define a Protocol for a Remote Node Record

When you configured the .Local node record, all remote node records are automatically configured to use the same protocol. If a trading partner uses a different protocol, you define this protocol in the remote node record.

If you have not already done so, complete a Local Node Security Feature Definition Worksheet for each trading partner who plans to use a protocol not defined in the .Local node record.

Configure a Remote Node Record for the TLS or SSL Protocol

About this task

When the remote node record needs to be configured to use a different protocol (TLS or SSL) than is defined in the .Local node record, complete the following procedure:

Procedure

1. From the Secure+ Admin Tool Main Window, double-click the remote node record to edit. The **Edit Record** window displays the **Security Options** tab, the node name, and the type of node.
2. Select the protocol to use by selecting **Enable TLS Protocol** or **Enable SSL Protocol**.
3. If necessary, change the time out value in the **Authentication Timeout** box.
4. Click the **TLS/SSL Protocol** tab.
5. Type the location of the trusted root certificate in the **Trusted Root Certificate File** box or click **Browse** and locate the file.
6. Click **Clear Certificate** to clear any associated certificate file from the remote node record.
7. Click **Browse** next to the **Certificate File** box, locate the key certificate file, and double-click the file to select it. The **Certificate Passphrase** dialog box is displayed.
8. Type the passphrase you specified when you created the certificate in the **Certificate Passphrase** box.
9. If you want to activate client authentication for the node, do the following:
 - Click **Yes** in the **Enable Client Authentication** box.
 - To enable another level of security, type the certificate common name in the **Certificate Common Name** box.
10. To identify the cipher suites that can be used by the remote node, do the following:
 - Turn off **Default to Local Node** from the Cipher Suites section.
 - Highlight the cipher suite to use in the **Available** list and click **Add**.
11. Click **OK** to close the **Edit Record** dialog box and update the parameters file.

Configure a Remote Node for the STS Protocol

About this task

When a trading partner uses the STS protocol, configure the remote node record to use the STS protocol, including creating signature and authentication keys and identifying the export version of Sterling Connect:Direct Secure Plus being used by the trading partner.

If you have not already done so, complete a Remote Node Security Feature Definition Worksheet for each trading partner who plans to use the STS protocol. Then use the procedures in this section to configure remote node records for the STS protocol.

After you create the keys for STS-enabled remote node records, you are responsible for managing them. The first time you use the STS protocol, you manually exchange keys with the trading partner. After you exchange keys for the first

Sterling Connect:Direct Secure Plus enabled communication, you can then turn on the automatic key management function. The Public Key Auto Update function enables the public keys to be updated automatically during a communications session. This simplifies key management for ongoing communications.

If you activate the STS protocol for a remote node record, you can use the COPY statement in the Sterling Connect:Direct Process statements to override settings in the remote node records. Refer to “Override STS Functions from the COPY Statement” on page 31 for information about overriding remote node record settings.

Complete the following procedure to configure a remote node for the STS protocol:

Procedure

1. Double-click the remote node to configure. The **Edit Record** dialog box displays the **Security Options** tab.
2. On the **Security Options** tab, click **Enable STS Protocol**.
3. If you want to allow the COPY statement to override values in the remote node, click **Enable Override**.
4. If necessary, change the value in the **Authentication Timeout** box.
5. Click the **STS Protocol** tab. The **Edit Record** dialog box displays STS Protocol information.
6. Set the STS Options as desired. Refer to the following table for an explanation of the options:

Field Name	Field Definition	Valid Values
Enable Digital Signatures	Enables or disables digital signatures.	Yes No Default to Local Node
Enable Public Key Auto Updates	Enables or disables automatic update of public keys during authentication. If this option is enabled for the PNODE and the SNODE, the public keys of the PNODE and SNDE are automatically exchanged during authentication. Auto update can only occur over a secure connection; therefore, the initial exchange of keys between two nodes must be done manually.	Yes No Default to Local Node
Limited Export Version	If a trading partner uses an earlier version of Sterling Connect:Direct Secure Plus, this field identifies the export version of Sterling Connect:Direct Secure Plus being used.	Yes No

Field Name	Field Definition	Valid Values
Enable Encryption	<p>Enables or disables data encryption during the copy operation.</p> <p>If Yes is selected, data encryption is enabled and the algorithm used depends on the merged value between the PNODE and SNODE. See “Merge Sterling Connect:Direct Secure Plus Settings Using the STS Protocol” on page 11 for more information about the merged values for data encryption.</p> <p>If No is specified, data encryption is disabled.</p> <p>If Algorithm is specified and encryption is enabled, Algorithm must be populated and Enable STS Protocol must be enabled.</p> <p>If the remote node enables encryption, the local node cannot disable it.</p>	<p>Yes No Default to Local Node</p> <p>Algorithm name</p>
Encryption Algorithms	<p>Lists acceptable data encryption algorithms when Copy file encryption is requested. List in order of preference, with the most-preferred algorithm first.</p>	<p>Default to Local Node</p> <p>DESCBC56</p> <p>TDESCBC112</p> <p>IDEACBC128</p>

7. Click the **Authentication Keys** tab.
8. Click **Generate Key** to create the authentication keys for the node. The **Generate Random Number Seed** dialog box is displayed. If a previous key exists, a message is displayed, prompting you to save the previous key.
9. Do one of the following to generate a key:
 - Type an alphanumeric string at least 32 characters long in the **Random Number Seed** box and click **OK**. Sterling Connect:Direct Secure Plus uses the random number seed value to generate the authentication keys.
 - Click **Sample Value**, highlight all of the grid squares, and click **OK** to generate the random number seed.

The **Authentication Keys** window displays the populated **Local Public Key** box.
10. If you are replacing an older key and you save the old key as the previous key, type the expiration date for the public key in the **Previous Key Pair Expiration Date** box. Refer to the following table for a definition of the Authentication Keys boxes.

Field Name	Field Definition	Valid Values
Local Public Key	Public key used for authentication.	Generated by Sterling Connect:Direct Secure Plus.
Previous Key Pair Expiration Date	Expiration date for previous authentication public keys. This eliminates the need to update Sterling Connect:Direct Secure Plus parameters files across all nodes in the network simultaneously when public keys for the local node are changed.	Format YYYY/MM/DD HH:MM:SS If time is not specified, 00:00:00 is used.
Remote Public Key	Displays the imported value from the trading partner.	Imported from the trading partner.

11. Click the **Signature Keys** tab.
 12. Click **Generate Key** to create the signature keys for the node. The **Generate Random Number Seed** dialog box is displayed. If a previous key exists, a message is displayed, prompting you to save the previous key.
 13. Do one of the following to generate a key:
 - Type an alphanumeric string at least 32 characters long in the **Random Number Seed** box and click **OK**. Sterling Connect:Direct Secure Plus uses the random number seed value to generate the signature keys.
 - Click **Sample Value**, highlight all of the grid squares, and click **OK** to generate the random number seed.

The **Signature Keys** window displays the populated **Public Key** box.
 14. If necessary, type the expiration date for the public key in the **Previous Key Pair Expiration Date** box.
- Refer to the following table for the name, definition, and valid values for the **Signature Keys** boxes.

Field Name	Field Definition	Valid Values
Local Public Key	Public key used for digital signature.	Generated by Sterling Connect:Direct Secure Plus.
Previous Key Pair Expiration Date	Expiration date for previous digital signature public keys. This eliminates the need to update Sterling Connect:Direct Secure Plus parameters files across all nodes in the network simultaneously when public keys for the local node are changed.	Format YYYY/MM/DD HH:MM:SS If time is not specified, 00:00:00 is used.
Remote Public Key	Displays the imported value from the trading partner.	Imported from the trading partner.

15. Click **OK** to close the remote node record and update the parameters file.

Manage STS Keys

When you define the STS protocol for a remote node, you must exchange keys with the trading partner before using Sterling Connect:Direct Secure Plus with that node.

To maintain the keys for the STS protocol, you perform the following procedures:

- Export keys to your trading partner
- Import keys from your trading partner

Export Keys

About this task

After you create signature and authentication keys for a node record, you must send this information to the trading partner. Export the information to a file that you can then send to the trading partner. Be sure that you export keys from the server associated with the local node.

Two ways of exporting keys are available. If you define several remote node records to use the STS protocol, you can create export key files for all remote node records at one time. If you want to export the key values for one remote node record, select the remote node record and export the selected remote node record key file.

To export the authentication and signature public key values for one or more remote node records:

Procedure

1. Create a specific directory for storing the public key files.
2. If necessary, open the Sterling Connect:Direct Secure Plus parameters file. The Secure+ Admin Tool Main Window is displayed.
3. Do one of the following:
 - If you want to export the key files for all remote node records, select **Export All Public Keys** from the **Key Management** menu item.
 - If you want to export the key file for one or more remote node records, highlight the node records to export and click the **Export Selection** option of the **Key Management** menu item.

The **Select Export Destination Directory** dialog box is displayed.

4. Highlight the directory that you created in step 1 and click **Select**.

If you selected **Export All Public Keys**, a file with an `sxp` extension is created for all remote node records. If you exported the keys of a selected remote node record, one file is created for the remote node record. Each file is named after the corresponding remote node record.

Import Keys

About this task

Before you can communicate with a trading partner, you must obtain their key data and import the information into the parameters file.

To import the authentication and signature public key values sent to you by the trading partner administrator:

Procedure

1. From the Secure+ Admin Tool Main Window, highlight the remote record to import the key from.
2. Click the **Import Public Keys** option of the **Key Management** menu item. The **Select Import Source File** dialog box is displayed.
3. Go to the directory that contains the signature and authentication key file for this remote node.
4. Highlight the `sxp` file for this node and click **Import**. The key is imported to the remote node record.
5. Ensure that the remote keys are imported by viewing the **Remote Public Key** box located in the **Authentication Keys** tab and **Signature Keys** tab.

Override STS Functions from the COPY Statement

After you set up the Sterling Connect:Direct Secure Plus environment, security is implemented each time that you use Sterling Connect:Direct with any node configured and enabled for Sterling Connect:Direct Secure Plus. You can, however, override some Sterling Connect:Direct Secure Plus functions from the COPY statement if you implemented the STS security protocol. This section tells you how to override the Sterling Connect:Direct Secure Plus features using the COPY statement.

Setting Sterling Connect:Direct Secure Plus Function Values from the COPY Statement

The SECURE COPY statement parameter enables you to set data encryption and digital signatures features from the Sterling Connect:Direct COPY statement. You can always enable these features from the COPY statement, but you cannot necessarily disable them from the COPY statement.

The SECURE parameter value specified in the COPY statement overrides the value specified in the Sterling Connect:Direct Secure Plus remote node record only if **Enable Override** is selected in that remote node record. After the security settings are merged between the PNODE and SNODE, the strongest setting is always used. Therefore, the value specified from the COPY statement cannot disable data encryption or digital signatures if the SNODE has enabled them.

If the override function is disabled in that remote node record and the values specified on the COPY statement are different from the values specified in the remote node record, the copy operation fails with a return code of 8 and message ID CSPA011E indicating the error.

The following table describes the SECURE parameters for the Sterling Connect:Direct COPY statement.

Parameter Name and Syntax	Parameter Description	Valid Values
secure=(encrypt.data=valid value) or secure=(enc=valid value)	Enables/disables Copy file encryption	Y N algorithm name DESCBC56 TDESCBC112 IDEACBC128 Default— parameters file value
secure=(signature=valid value) or secure=(sig=valid value)	Enables/disables digital signature creation	Y N Default—parameters file value

Note: Data encryption and digital signatures cannot necessarily be disabled from the COPY statement.

If both parameters are used from the COPY statement, the syntax is as follows:

```
secure=(enc=y,sig=y)
or
secure=(encrypt.data=y,signature=y)
```

The following sample Process, SECURE, copies the file README.TXT from the PNODE to the SNODE, renames it to VERIFY.TXT, and enables data encryption and digital signatures.

```
step1 copy
  from (
    file=\usr\ndm\bin\readme.txt
  )
  secure=(encrypt.data=y,signature=y)
  to (
    file=\usr\ndm\bin\verify.txt
    disp=(rpl)
  )
pend
```

Validate the Configuration

About this task

Perform this procedure to ensure that the nodes have been properly configured. The validation process checks each node to ensure that all necessary options have been defined and keys have been exchanged.

To validate the parameters file:

Procedure

1. In the Secure+ Admin Tool, click **File > Validate Secure+**. The **Secure+ Admin Tool - Validation Results** window is displayed.

If the parameters file is not correctly configured, warning and error messages are displayed.

2. Read each warning message. To correct each warning or error reported, go back to the parameters file and make changes as needed.

Note: Warning messages do not always mean that the parameters file is incorrectly configured. Some warning messages are informational only.

3. Click **Close** to close the **Validation Results** window.

Configure External Authentication in the .SEAServer Record

About this task

At installation, a record named `.SEAServer` is created in the parameters file, which enables Sterling Connect:Direct Secure Plus to interface with Sterling External Authentication Server during SSL/TLS sessions to validate certificates. Sterling External Authentication Server properties are configured in this record and enabled/disabled in the local and remote node records.

Complete the following procedure to configure the server properties that will allow Sterling Connect:Direct for Microsoft Windows to interface with Sterling External Authentication Server:

Note: The values specified for this procedure must match the values specified in Sterling External Authentication Server.

Procedure

1. Double-click the record called `.SEAServer`.
2. Type the Host Name for Sterling External Authentication Server.
3. Type the Port Number where Sterling External Authentication Server is listening. The default is 61366.
4. Click **OK** to update the record.

Related concepts:

“Secure Sockets Layer Protocol (SSL) and Transport Layer Security Protocol (TLS)” on page 2

Use the .Client Record to Prevent Non-Secure API Connections to a Sterling Connect:Direct Secure Plus-Enabled Server

About this task

Sterling Connect:Direct servers that use Sterling Connect:Direct Secure Plus support secure API connections. Client applications that use the API include IBM Sterling Control Center, Sterling Connect:Direct Java API, IBM Sterling Connect:Direct Requester, and IBM Sterling Connect:Direct Browser User Interface. Client applications created using the Sterling Connect:Direct Java API support secure API connections; client applications created using the Sterling Connect:Direct SDK do not support secure API connections.

If a Sterling Connect:Direct server has Sterling Connect:Direct Secure Plus enabled, an API configuration that connects to this server uses the `.Client` node record to determine if it has the authority to connect to the server. The override settings in the `.Client` node record determine if the client application has the authority to connect to the server. When override is enabled in the `.Client` record, secure and

non-secure API connections are allowed to connect to the Sterling Connect:Direct server. When override is disabled in the .Client record, all client connections must use the effective protocol configured in the .Client or .Local node record.

Valid settings for API connections include TLS, SSL, or security disabled. An API configuration follows the same rules as other remote node connections with the following exceptions:

- API connections must use either the SSL or the TLS security protocol
- Certificate-based client authentication is not supported
- The local node is automatically overridden by settings in the .Client node definition

Since the local node is automatically overridden, Sterling Connect:Direct Secure Plus configurations that normally disable this setting will not be affected. In order to disable override and prevent API connections that are either nonsecure or that do not use the settings defined in the local node, configure the .Client node record and disable override.

To prevent non-secure connections to a Sterling Connect:Direct server with Sterling Connect:Direct Secure Plus enabled:

Procedure

1. Double-click the remote node record called **.Client**.
2. Click the **Disable Override** option.
3. If necessary, configure the TLS or SSL protocol in this node, if it is not defined in the .Local node record. See “Configure a Remote Node Record for the TLS or SSL Protocol” on page 26.

Note: The TLS or SSL protocol must be configured in either the .Client or the .Local node record.

4. Click **OK** to update the node record.

Configure Strong Password Encryption

This feature uses strong encryption to encrypt all Sterling Connect:Direct passwords stored on disk. This feature is enabled by default.

Disable Strong Password Encryption

About this task

To disable strong password encryption, do the following:

Procedure

1. From the Secure+ Admin Tool main menu, select **Edit > Password Encryption**.
2. Click the **No** option for **Enable Strong Password Encryption**.
3. Click **OK**. The following message displays:

The Connect:Direct server must be restarted to update Strong Password Encryption.

Note: The change does not take effect until you restart Sterling Connect:Direct.

Enable Strong Password Encryption

About this task

To enable strong password encryption, do the following:

Procedure

1. From the Secure+ Admin Tool main menu, select **Edit > Password Encryption**.
2. Click the **Yes** option for **Enable Strong Password Encryption**.
3. Click **OK**. The following message displays:
The Connect:Direct server must be restarted to update Strong Password Encryption.

Note: The change does not take effect until you restart Sterling Connect:Direct.

Reset Passwords When Strong Password Encryption Key and Password File are Out of Sync

About this task

If the Strong Password Encryption key stored in the .Password file becomes out of sync with the Strong Encryption key used to encrypt the passwords, you must reset all Strong Password Encryption passwords.

The .Password file can get out of sync if one of the following occurs:

- You restore the .Password file from a backup—The .Password file is updated each time the Sterling Connect:Direct server is started, so the backup will probably not contain the current parameters.
- The .Password file is deleted—The .Password file is recreated as needed, so the Strong Password Encryption key used to encrypt the passwords no longer exists.
- The .Password file is corrupt—The Strong Password Encryption key used to encrypt the passwords is not accessible.

To reset the passwords:

Procedure

1. Delete the <CD installation directory>\Server\Secure+\Nodes\.password file.
2. Use Sterling Connect:Direct Requester to re-enter local user ID/password for any proxy records.
3. Use Sterling Connect:Direct Requester to manually delete all Processes in the TCQ.
4. Stop the Sterling Connect:Direct server.
5. Reset the message file database password by running the Sterling Connect:Direct Database Wizard.
6. Reset the SMTP password using the Change Notification Utility.
7. Start the Sterling Connect:Direct server.

Decryption Failure

If the Process KQV file fails decryption at startup or during runtime, the server places the Process in the Hold/Error queue to raise the visibility of the error.

Chapter 5. Automate Setup Using the CLI

Start and Set Up the Sterling Connect:Direct Secure Plus CLI

The following sections describe the commands and parameters used to start and set up the command line environment.

Start the Sterling Connect:Direct Secure Plus CLI

To start the Sterling Connect:Direct Secure Plus Command Line Interface:

1. Change to the following directory: C:\Program Files\ Sterling Commerce\Connect Direct v4.6.00\Server\Secure+.

2. Enter the following command:

```
spcli.cmd
```

Control the Display of Commands

Set the following parameters to define how error messages are captured:

Parameter	Definition	Values
-li	Switch to enable the display of commands to the terminal.	y n
-lo	Switch to enable the display of output and error messages to the terminal.	y n
-le	Switch to enable the display of errors to STDERR.	y n
-e	Switch to tell the Sterling Connect:Direct Secure Plus CLI to exit when the return code is higher than the specified number. If you do not include this parameter, Sterling Connect:Direct Secure Plus CLI runs even after an error occurs.	0 4 8 16
-p	Full path of the default parameters file directory. The file in this directory is opened automatically.	
-h	Switch to display the usage of the Sterling Connect:Direct Secure Plus CLI.	

Control Help

The Help command determines what help information is displayed. You can list all the Sterling Connect:Direct Secure Plus CLI commands and display help for individual commands.

Command	Description
help	Displays all the Sterling Connect:Direct Secure Plus CLI commands.
help <command>	Displays help for the specified command.

Specify Delimiter Characters

Define the following commands to determine how error messages are captured:

Command	Definition	Values
Set begdelim= enddelim=	Defines beginning and ending character to use to enclose keywords that use blanks and other special characters.	Any character The default value is " (double quotes).

Use LCU Files to Encrypt Passwords for Use with the Sterling Connect:Direct Secure Plus CLI

The Sterling Connect:Direct Secure Plus CLI displays passwords in plain text. If you are required to use encrypted passwords, use the Local Connection Utility (LCU) to create an LCU file that contains non-encrypted information for encrypting the password.

For more information on creating and using LCU files, see “Configure Encrypted Passwords Using the LCU” on page 97.

Sample Scripts

The following scripts are provided as models for creating custom scripts to define your Sterling Connect:Direct Secure Plus environment and automate its implementation. To prevent any loss of data, you cannot run the scripts, but you can save them with a different name and modify them to suit your needs.

The sample scripts are available in the Automation Scripts topic. The scripts are designed to assist you as follows:

Microsoft Windows Script	Explanation
spcust_sample1.cmd	An example of configuring Sterling Connect:Direct Secure Plus for the STS protocol with the CLI. It demonstrates statically generated public keys to facilitate a mass implementation
spcust_sample2.cmd	An example of configuring Sterling Connect:Direct Secure Plus for the STS protocol with the CLI. It demonstrates the use of statically generated public keys to facilitate the automated addition of a remote node to a Sterling Connect:Direct network. A script based on this sample is used to configure the existing nodes in the network map. The remote node itself is configured with a script based on spcust_sample1. In order for the operation to be successful, the two scripts must use the same seeds to generate keys.
spcust_sample3.cmd	An example of configuring the use of the SSL or TLS protocol with the Sterling Connect:Direct Secure Plus CLI. It configures Sterling Connect:Direct Secure Plus with the trusted root file, key certificates, and ciphers.

The following list describes the tasks necessary to write and implement the script for a mass implementation of an initial installation that uses the STS protocol. Excerpts from the `spcust_sample1` sample script are provided for reference.

1. Use the `Create STSKeyPair` command to generate a passphrase-protected file (default name: `keypair`) in the format that is already in use for importing and exporting public keys. In this key pair file, the private key is encrypted, but the public key is not. The public key string can be cut and pasted into `spcust_samplex` scripts for setting the remote keys for remote nodes.
2. Distribute the key pair file along with the `SPCLI` script when you distribute the installation to additional sites.
3. Use the `Update LocalNode` command to set the current key pair to the key pair stored in the key pair file. When using the key pair file to set the local keys in local and remote nodes, you need the key pair file name and its passphrase, as shown in the following example from the `spcust_sample1` script:

```
StsAuthLocalKey=set
StsAuthKeyPairFile=keypairfile
StsAuthKeyPairFilePassphrase=secret
StsSigLocalKey=set
StsSigKeyPairFile=keypairfile
StsSigKeyPairFilePassphrase=secret
```

4. For each remote node, use the `Update RemoteNode` command to set the public key that was generated in step 1. Copy the public key string from the key pair file and paste it into the script on the lines where the remote keys of all the remote nodes are set, as shown in this example from the `spcust_sample1` script:

```
update remotenode name=*
STSAuthRemoteKey=public key string
STSSigRemoteKey=public key string
```

5. Use the `Update LocalNode` command to generate a new key pair and save the existing key pair as the previous key pair value. For example, in `spcust_sample1`, after the authentication and signature keys were set to the ones stored in the key pair file generated in step 1, the `Update LocalNode` command was used again, as shown in the following example:

```
update localnode
stsauthlocalkey=gen
stsauthkeyseedType=dynamic
STSPrevAuthKeyExpDateTime=2005:01:01-20:13:44
stssiglocalkey=gen
stssigkeyseedType=dynamic
STSPrevSigKeyExpDateTime=2005:01:01-20:13:45
;
```

Note: The CLI script is called `spcli.cmd`. Each command within the script must be terminated with a semicolon (;). Use the `quit` command to exit the `spcli.cmd` script.

Related tasks:

“Automatically Create STS Keys for Remote Node Records” on page 56

Manage the Parameters File

The following table describes commands used to manage and maintain the parameters file from the command line interface.

Command	Description	Parameter	Values
Init Parmfile	Creates the parameters file. Must be initialized before you can define nodes.	localnode=Name of the local node where the parameters file will be created.	local node name
		path=Location where the parameters file will be created.	directory location For example, <i>d_dir/ndm/secure+/ node</i>
		passphrase=Arbitrary set of characters that encrypts the parameters file.	a string at least 32 characters long
Open parmfile	Opens a parameters file so that you can configure it.	path=Location where the parameters file will be created.	directory location For example, <i>d_dir/ndm/secure+/ node</i>
Close parmfile	Closes the parameters file. After this command is issued, no more updates can be performed on the parameters file.	None	None
Validate parmfile	Validates the parameters file and ensures that it is a valid file.	None	None
Rekey parmfile	Recreates the parameters file if it becomes corrupted.	passphrase=Arbitrary set of characters that encrypts the parameters file.	passphrase, up to 32 characters long

Command	Description	Parameter	Values
Sync netmap	Imports remote node records defined in the Sterling Connect:Direct network map.	path=Location and name of the network map file.	location of network map file
		name=Name of the node in the network map. Use wildcard characters to resync more than one node at a time.	node name wildcard Wildcard values are: Asterisk (*)—any number of characters. Example: kps.* syncs up all nodes with a name that starts with kps. Question mark (?)—a single character. Example: k?s.* syncs up kas.* and kbs.*.

Displaying Information

The following commands are available to display information:

Command	Description	Parameter
display info	Displays information about when the parameters file was last updated.	None
display all	Displays all nodes in the parameters file.	None
display localnode	Displays the values defined in the .Local node record.	None
display remotenode	Displays the values defined in remote node records.	name—The name of the node to display information about. Use wildcard characters to display information about a group of remote node records. The options are: Asterisk (*)—Indicates any number of characters. For example, kps.* displays all nodes with a name that starts with kps. Question mark (?)—Indicates a single character. For example: k?s.* displays kas.* and kbs.*.
display client	Displays the values defined in the .Client node record.	None
display seaserver	Displays the values defined in the .SEAServer record.	None

Update the .Local Node Record

The update localnode command configures the protocol for the .Local node record.

The update localnode command has the following parameters:

Parameter	Values
protocol=Protocol to use in the .Local node record.	sts tls ssl disable
override=Identifies if values in the remote node can override values defined in the .Local node record.	y n
AuthTimeout=maximum time, in seconds, that the system waits to receive the Sterling Connect:Direct control blocks exchanged during the Sterling Connect:Direct Secure Plus authentication process.	0–3600 seconds
StsEnableSig=Enables digital signatures.	y n
StsEnableAutoUpdate=Enables automatic update of public keys during authentication.	y n
StsLimitExportVersion=Identifies if a trading partner uses an earlier version of Sterling Connect:Direct Secure Plus designated as an export version.	y n
StsEnableEnc=Enables data encryption during the copy operation.	y n algorithm (that is, the actual text string, "algorithm," which is not case-sensitive). If y is selected, data encryption is enabled and the algorithm used depends on the merged value between the PNODE and SNODE. If the remote node enables encryption, the local node cannot disable it. If this parameter is set to algorithm, the algorithm must be defined in the StsEnableEncAlg parameter.
StsEnableEncAlg=Acceptable data encryption algorithms to use when the Copy file encryption is requested. List in order of preference, with the most-preferred algorithm first.	DESCBC56 TDESCBC112 IDEACBC128

Parameter	Values
StsEncAlgs=Acceptable data encryption algorithms to use when encrypting control data during secure communications. List in order of preference, with the most-preferred algorithm first.	DESCBC56 TDESCBC112 IDEACBC128 all null all—Makes all valid algorithms available for use. null—Clears any existing values from the node definition. To enable more than one algorithm, use parenthesis to enclose all algorithms listed and a comma to separate each algorithm. For example, to enable two algorithms define the following parameter: StsEncAlgs=(DESCBC56, TDESCBC112)
StsAuthLocalKey=Generates or clears the authentication keys for the node.	gen clear set gen—Generates the authentication key. If this parameter is defined as gen, the StsAuthKeySeed value must be defined. clear—Clears the authentication key value. If this command is defined as clear, the StsAuthKeysToClear must also be specified. set—Sets the value of the authentication key to the value that you define in the StsAuthKeySeed parameter.
StsAuthKeyPairFile=Key pair file that contains a pair of public and private keys.	key pair file name
StsAuthKeyPairFilePassphrase=The passphrase specified when the key pair file is created.	alphanumeric string LCU:filename Note: For more information on using LCU files to mask plain text passwords in the CLI, see “Use LCU Files to Encrypt Passwords for Use with the Sterling Connect:Direct Secure Plus CLI” on page 38.
StsAuthPreservePrev=Determines which authentication key values are saved.	y n
StsAuthKeySeedType=Identifies what type of key seed will be generated.	dynamic static value dynamic—The key seed is generated by the system each time. It is not necessary to define a value for the StsAuthKeySeed parameter. value—The key seed is generated based on the value defined in the StsAuthKeySeed parameter and changes each time the key is generated, even when the StsAuthKeySeed parameter is unchanged. static—the key seed is generated based on the value defined in the StsAuthKeySeed parameter. If the value of StsAuthKeySeed is not changed, the same Auth key is generated.

Parameter	Values
StsAuthKeySeed=An alphanumeric string used to generate the authentication keys. This value is required when StsAuthKeySeedType is defined as static or value.	alphanumeric string at least 32 characters
StsAuthKeysToClear=Identifies the authentication keys to clear.	previous current both
StsPrevAuthKeyExpDateTime=Specifies the date and time that the previous authentication key expires.	yyyy:mm:dd-hh:mm:ss The default value is 30 days.
StsSigLocalKey=Creates the signature keys for the node.	gen clear set gen—Generates the key. If this parameter is defined as gen, the StsSigKeySeed value must be defined. clear—Clears the key value. If this command is defined as clear, the StsSigKeysToClear must also be specified. set—Sets the value of the key to the value that you define in the StsSigKeySeed parameter.
StsSigKeyPairFile=Key pair file that contains a pair of public and private keys.	key pair file name
StsSigKeyPairFilePassphrase=Passphrase specified when the key pair file is created.	alphanumeric string LCU:filename Note: For more information on using LCU files to mask plain text passwords in the CLI, see "Use LCU Files to Encrypt Passwords for Use with the Sterling Connect:Direct Secure Plus CLI" on page 38.
StsSigPreserveKey=Determines if previous signature key values are saved.	current previous
StsSigKeySeed=Alphanumeric string used to generate the signature keys. This value is required when StsSigKeySeedType is defined as static or value.	alphanumeric string at least 32 characters
StsSigKeySeedType=Identifies what type of key seed will be generated.	dynamic static value dynamic—The key seed is generated by the system each time. It is not necessary to define a value for the StsSigKeySeed parameter. value—The key seed is generated based on the value defined in the StsSigKeySeed parameter and changes each time the key is generated, even when the StsSigKeySeed parameter is unchanged. static—The key seed is generated based on the value defined in the StsSigKeySeed parameter. If the value of StsSigKeySeed is not changed, the same Sig key is generated.

Parameter	Values
StsSigKeysToClear=Identifies the signature keys to clear.	previous current both
StsPrevSigKeyExpDateTime=The date and time that the previous signature key expires.	yyyy:mm:dd-hh:mm:ss The default value is 30 days
SslTlsTrustedRootCertFile=The location of the trusted root certificate file to use for client connections.	location null null—Clears any existing values from the node definition.
SslTlsCertFile=Identifies the location of the certificate file to use	location null null—Clears any existing values from the node definition.
SslTlsCertPassphrase=The passphrase specified when the certificate is created.	alphanumeric password assigned when the key was generated LCU:filename Note: For more information on using LCU files to mask plain text passwords in the CLI, see “Use LCU Files to Encrypt Passwords for Use with the Sterling Connect:Direct Secure Plus CLI” on page 38.
SslTlsEnableClientAuth=Enables client authentication in a .Client node record.	y n
SslTlsEnableCipher=Specifies the cipher suites enabled	name of a cipher suite null all all—Enables all ciphers. null—Clears any existing values from the node definition.
SslTlsEnableFipsMode=Turns on Sterling Crypto-C module that provides a FIPS 140-2 validated cryptographic module.	y n
SslTlsSeaEnable=Enables certificate validation by Sterling External Authentication Server.	y n
SeaCertValDef=Character string defined in Sterling External Authentication Server.	character string null null—Clears any existing values from the node definition.

Manage Remote Node Records

This section contains the commands and parameters used to create, update, display, and delete remote node records.

Create a Remote Node Record

The create remotenode command creates a remote node record and configures the protocol settings. The command has the following parameters:

Parameter	Values
model=Name of an existing node to use as a model to copy from.	name of a valid remote node

Parameter	Values
Name=Name for the remote node record.	name
protocol=Protocol to use in the node record.	sts tls ssl disable DefaultToLN
override=Identifies if values in the copy statement can override values defined in the remote node record.	y n DefaultToLN
AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the Sterling Connect:Direct control blocks exchanged during the Sterling Connect:Direct Secure Plus authentication process.	0–3600. The default is 120 seconds.
StsEnableSig=Enables digital signatures.	y n DefaultToLN
StsEnableAutoUpdate=Enables automatic update of public keys during authentication.	y n DefaultToLN
StsLimitExportVersion=Identifies if a trading partner is using an earlier version of Sterling Connect:Direct Secure Plus designated as an export version.	y n
StsEnableEnc=Enables data encryption during the copy operation.	y n DefaultToLN algorithm (that is, the actual text string, “algorithm,” which is not case-sensitive). If y is selected, data encryption is enabled and the algorithm used depends on the merged value between the PNODE and SNODE. If the remote node enables encryption, the local node cannot disable it. If this parameter is set to algorithm, the algorithm must be defined in the StsEnableEncAlg parameter.
StsEnableEncAlg=Acceptable data encryption algorithms to use when the Copy file encryption is requested. List in order of preference, with most-preferred algorithm first.	DESCBC56 TDESCBC112 IDEACBC128
StsEncAlgs=Acceptable data encryption algorithms to use when encrypting control data during secure communications. List in order of preference, with most-preferred algorithm first. ‘	DESCBC56 TDESCBC112 IDEACBC128 All null all—Makes all valid algorithms available for use. null—Clears any existing values from the node definition. To enable more than one algorithm, use parenthesis to enclose all algorithms listed and a comma to separate each algorithm. For example, to enable two algorithms define the following parameter: StsEncAlgs=(DESCBC56, TDESCBC112)

Parameter	Values
StsAuthLocalKey=Generates, clears, or sets the authentication keys for the node.	<p>gen clear set</p> <p>gen—Generates the authentication key. If this parameter is defined as gen, the StsAuthKeySeed value must be defined.</p> <p>clear—Clears the authentication key value. If this command is defined as clear, the StsAuthKeysToClear must also be specified.</p> <p>set—Sets the value of the authentication key to the value that you define in the StsAuthKeySeed parameter.</p>
StsAuthKeySeedType=Identifies what type of key seed will be generated.	<p>dynamic static value</p> <p>dynamic—The system generates the key seed each time. It is not necessary to define a value for the StsAuthKeySeed parameter.</p> <p>value—The key seed is generated based on the value defined in the StsAuthKeySeed parameter and changes each time the key is generated, even when the StsAuthKeySeed parameter is unchanged.</p> <p>static—The key seed is generated based on the value defined in the StsAuthKeySeed parameter. If the value of StsAuthKeySeed is not changed, the same Auth key is generated.</p>
StsAuthKeyPairFile=Key pair file that contains a pair of public and private keys for authentication.	key pair file name
StsAuthKeyPairFilePassphrase= Passphrase specified when the key pair file is created.	alphanumeric string
StsAuthKeySeed=An alphanumeric string used to generate the authentication keys. This value is required when StsAuthKeySeedType is defined as static or value.	alphanumeric string at least 32 characters long
StsAuthRemoteKey=The authentication keys used to authenticate the remote node or clear to clear the value of the remote key.	<p>key value clear</p> <p>key value—The value assigned to the authentication key.</p> <p>clear—Clears the value of the authentication key. If clear is specified, the StsAuthKeysToClear must also be specified.</p>
StsAuthKeysToClear=Identifies the authentication keys to clear.	previous current both
StsPrevAuthKeyExpDateTime= Identifies the date and time that the previous authentication key expires.	yyyy:mm:dd-hh:mm:ss

Parameter	Values
StsSigLocalKey=Creates the signature keys for the node.	gen clear set gen—Generates the key. If this parameter is defined as gen, the StsSigKeySeed value must be defined. clear—Clears the key value. If this command is defined as clear, the StsSigKeysToClear must also be specified. set—Sets the value of the key to the value that you define in the StsSigKeySeed parameter.
StsPrevSigKeyExpDateTime=The date and time that the previous signature key expires.	yyyy:mm:dd-hh:mm:ss
StsSigKeySeedType=Identifies what type of key seed will be generated.	dynamic static value dynamic—The key seed is generated by the system each time. It is not necessary to define a value for the StsSigKeySeed parameter. static—The key seed is generated based on the value defined in the StsSigKeySeed parameter and changes each time the key is generated, even when the StsSigKeySeed parameter is unchanged. value—The key seed is generated based on the value defined in the StsSigKeySeed parameter. If the value of StsSigKeySeed is not changed, the same Sig key is generated.
StsSigKeySeed=An alphanumeric string used to generate the signature keys. This value is required when StsSigKeySeedType is defined as static or value.	alphanumeric string at least 32 characters long
StsSigRemoteKey=The signature key used to authenticate the remote node or clear to clear the value of the remote key.	key value clear key value—The value assigned to the key. clear—Clears the value of the key. If clear is specified, the StsSigKeysToClear must also be specified.
StsSigKeyPairFile=Key pair file that contains a pair of public and private keys.	key pair file name
StsSigKeyPairFilePassphrase= Passphrase specified when the key pair file is created.	alphanumeric string
StsSigKeysToClear=Identifies the signature keys to clear.	previous current both
SslTlsTrustedRootCertFile=Location of the trusted root certificate file to use for client connections.	location null null—Clears any existing values from the node definition.

Parameter	Values
SslTlsCertFile=Identifies the location of the certificate file to use	location null null—Clears any existing values from the node definition.
SslTlsCertPassphrase=Passphrase specified when the certificate is created.	alphanumeric password assigned when the key was generated.
SslTlsEnableClientAuth=Enables client authentication in a .Client node record.	y n DefaultToLN DefaultToLN—Defaults to the setting specified in the .Local node record
SslTlsSeaEnable=Enables certificate validation by Sterling External Authentication Server.	y n DefaultToLN DefaultToLN—Defaults to the setting specified in the .Local node record
SeaCertValDef=Character string defined in Sterling External Authentication Server.	character string null null—clears any existing values from the node definition
SslTlsCertCommonName=The certificate common name defined in the certificate	name null null—Clears any existing values from the node definition.
SslTlsEnableCipher=The cipher suites enabled	name of a cipher suite all null all—Enable all ciphers. null—Clears any existing values from the node definition.

Update the Remote Node Record

The update remotenode command creates a remote node record and configures the protocol settings. The command has the following parameters:

Parameter	Values
Name=Name for the remote node record.	remote node name wildcard Use wildcard characters to update a group of remote node records. The options are: Asterisk (*)—Any number of characters. Example: kps.* displays remote nodes with a name that starts with kps. Question mark (?)—Single character. Example: k?s.* displays kas.* and kbs.*.
protocol=The protocol to use in the node record.	sts tls ssl disable DefaultToLN
override=Identifies if values in the copy statement can override values defined in the remote node record.	y n DefaultToLN

Parameter	Values
AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the Sterling Connect:Direct control blocks exchanged during the Sterling Connect:Direct Secure Plus authentication process.	0–3600 seconds
StsEnableSig=Enables digital signatures.	y n DefaultToLN
StsEnableAutoUpdate=Enables automatic update of public keys during authentication.	y n DefaultToLN
StsLimitExportVersion=Identifies If a trading partner is using an earlier version of Sterling Connect:Direct Secure Plus designated as an export version.	y n
StsEnableEnc=Enables data encryption during the copy operation.	y n algorithm DefaultToLN y—Enable data encryption. If y is selected, data encryption is enabled and the algorithm used depends on the merged value between the PNODE and SNODE. If the remote node enables encryption, the local node cannot disable it. n—Disable data encryption DefaultToLN—Use the values defined in the .Local node record. algorithm—The text string, “algorithm,” which is not case-sensitive. If this parameter is set to algorithm, the algorithm must be defined in the StsEnableEncAlg parameter.
StsEnableEncAlg—Acceptable data encryption algorithms to use when the Copy file encryption is requested. List in order of preference, with most-preferred algorithm first.	DESCBC56 TDESCBC112 IDEACBC128
StsAuthLocalKeygenerates, clears, or sets the authentication keys for the node.	gen clear set gen—Generates the authentication key. If this parameter is defined as gen, the StsAuthKeySeed value must be defined. clear—Clears the authentication key value. If this command is defined as clear, the StsAuthKeysToClear must also be specified. set—Sets the value of the authentication key to the value that you define in the StsAuthKeySeed parameter.

Parameter	Values
StsEncAlgs=Acceptable data encryption algorithms to use when encrypting control data during secure communications. List in order of preference, with most-preferred algorithm first.	DESCBC56 TDESCBC112 IDEACBC128 All null All—Sets all valid algorithms available for use. null—Clears any existing values from the node definition. To enable more than one algorithm, use parenthesis to enclose all algorithms listed and a comma to separate each algorithm. For example, to enable two algorithms define the following parameter: StsEncAlgs=(DESCBC56, TDESCBC112)
StsAuthPreserveKey=Determines if the previous authentication key values are saved.	y n current previous
StsAuthKeysToClear=Identifies the authentication keys to clear.	current previous both
StsAuthKeySeedType=Identifies what type of key seed will be generated.	dynamic static value dynamic—The system generates the key seed each time. It is not necessary to define a value for the StsAuthKeySeed parameter. value—The key seed is generated based on the value defined in the StsAuthKeySeed parameter and changes each time the key is generated, even when the StsAuthKeySeed parameter is unchanged. static—The key seed is generated based on the value defined in the StsAuthKeySeed parameter. If the value of StsAuthKeySeed is not changed, the same authentication key is generated.
StsAuthKeySeed=An alphanumeric string used to generate the authentication keys. This value is required when StsAuthKeySeedType is defined as static or value.	an alphanumeric string at least 32 characters long
StsAuthRemoteKey=Authentication keys used to authenticate the remote node or clear to clear the value of the remote key.	key value clear key value—Value assigned to the authentication key. clear—Clear the value of the authentication key. If clear is specified, the StsAuthKeysToClear must also be specified.
StsPrevAuthKeyExpDateTime=Date and time that the previous authentication key expires.	yyyy:mm:dd-hh:mm:ss The default value is 30 days
StsAuthKeyPairFile=Key pair file that contains a pair of public and private keys.	key pair file name

Parameter	Values
StsAuthKeyPairFilePassphrase= Passphrase specified when the key pair file is created.	alphanumeric string
StsSigLocalKey=Creates the signature keys for the node.	gen clear set gen—Generates the key. If this parameter is defined as gen, the StsSigKeySeed value must be defined. clear—Clears the key value. If this command is defined as clear, the StsSigKeysToClear must also be specified. set—Sets the value of the key to the value that you define in the StsSigKeySeed parameter.
StsSigPreserveKey=Determines if previous signature key values are saved.	current previous
StsSigKeySeedType=Identifies what type of key seed will be generated.	dynamic static value dynamic—The key seed is generated by the system each time. It is not necessary to define a value for the StsSigKeySeed parameter. value—The key seed is generated based on the value defined in the StsSigKeySeed parameter and changes each time the key is generated, even when the StsSigKeySeed parameter is unchanged. static—The key seed is generated based on the value defined in the StsSigKeySeed parameter. If the value of StsSigKeySeed is not changed, the same Signature key is generated.
StsSigKeySeed=Alphanumeric string used to generate the signature keys. This value is required when StsSigKeySeedType is defined as static or value.	alphanumeric string at least 32 characters long
StsPrevSigKeyExpDateTime=Date and time that the previous signature key expires.	yyyy:mm:dd-hh:mm:ss The default value is 30 days
StsSigRemoteKey=Signature key used to authenticate the remote node or clear to clear the value of the remote key.	key value clear key value—The value assigned to the key. clear—Clears the value of the key. If clear is specified, the StsSigKeysToClear must also be specified.
StsSigKeysToClear=Identifies the signature keys to clear.	previous current both
StsSigKeyPairFile=Key pair file that contains a pair of public and private keys.	key pair file name
StsSigKeyPairFilePassphrase=Passphrase specified when the key pair file is created.	alphanumeric string

Parameter	Values
SslTlsTrustedRootCertFile=Location of the trusted root certificate file to use for client connections.	location null null—Clears any existing values from the node definition.
SslTlsCertFile=Identifies the Location of the certificate file to use.	location null null—Clears any existing values from the node definition.
SslTlsCertPassphrase=Passphrase specified when the certificate is created.	alphanumeric password assigned when the key was generated.
SslTlsEnableClientAuth=Enables client authentication in a .Client node record.	y n DefaultToLN DefaultToLN—Defaults to the setting specified in the .Local node record
SslTlsSeaEnable=Enables certificate validation by Sterling External Authentication Server.	y n DefaultToLN DefaultToLN—Defaults to the setting specified in the .Local node record
SslTlsCertCommonName=Certificate common name defined in the certificate	name null null—Clears any existing values from the node definition.
SeaCertValDef=Character string defined in Sterling External Authentication Server.	character string null null—clears any existing values from the node definition
SslTlsEnableCipher=Cipher suites enabled	name of a cipher suite null all all enables all ciphers. null—Clears any existing values from the node definition.

Display a Remote Node Record

The display remotenode command displays information about one or more remote node records. The command has the following parameter:

Parameter	Values
name=Name of the remote node record to display information about.	node name wildcard value To display information about more than one remote node record, use wildcard characters. Use wildcard characters to display information about a group of remote node records. The options are: Asterisk (*)—Any number of characters. Example: kps.* displays remote nodes with a name that starts with kps. Question mark (?)—A single character. Example: k?s.* displays kas.* and kbs.*.

Delete a Remote Node Record

The delete remotenode command deletes one or more remote node records. The command has the following parameter:

Parameter	Values
name=Name of the remote node record to display information about. Use wildcard characters to delete a group of remote node records.	remote node name wildcard value To display information about more than one remote node record, use wildcard characters. Use wildcard characters to display information about a group of remote node records. The options are: Asterisk (*)—Any number of characters. Example: kps.* displays remote nodes with a name that starts with kps. Question mark (?)—A single character. Example: k?s.* displays kas.* and kbs.*.

Manage the Sterling External Authentication Server Record

This section contains the commands and parameters used to update and display the .SEAServer record.

Update the Sterling External Authentication Server Record

The update seaserver command configures properties for Sterling External Authentication Server in the .SEAServer record that is created at installation. The command has the following parameters:

Parameter	Values
Protocol=Protocol to use in the .SEAServer record.	sts tls ssl disable DefaultToLN
SeaHost=External authentication host name defined in Sterling External Authentication Server.	host name null null—Clears any existing values from the node definition
AuthTimeout=Specifies the maximum time, in seconds, that the system waits to receive the Sterling Connect:Direct control blocks exchanged during the Sterling Connect:Direct Secure Plus authentication process.	0–3600 seconds
Override=Identifies if values in the copy statement can override values defined in the remote node record.	y n DefaultToLN
SeaPort=External authentication server port number (listening) defined in Sterling External Authentication Server.	port number 61366

Display the Sterling External Authentication Server Record

The display SEAServer command displays information about the .SEAServer record.

Maintain the .Password File (Strong Password Encryption)

This section contains the commands and parameters used to update and display the .Password file through the Sterling Connect:Direct Secure Plus CLI.

Update the .Password File

The **Update Password** command enables or disables strong password encryption. The update goes into effect after you start the Sterling Connect:Direct server. The command has one parameter, `SpeEnable`, which can be set to Y or N to enable or disable strong password encryption. Following is an example:

```
Update Password
    SpeEnable=<Y>
;
```

If you enable or disable strong password encryption, the server displays the following warning:

```
The Connect:Direct Server must be restarted to update Strong
Password Encryption.
```

Display the .Password File

The **Display Password** command displays the strong password encryption setting.

The **Display All** command displays the strong password encryption setting and .Password history.

Manage STS Keys Using the CLI

The commands and parameters in this section describe how to create, export, and import STS keys.

Create an STS Key Pair

The `create STSKeyPair` command creates a key pair of public and private keys. The only required parameter is `passphrase`. The command has the following parameters:

Parameter	Values
<code>KeyPairFile=</code> File in which to store the encrypted key pair.	Valid file name. The default value is <code>keypair</code> .
<code>passphrase=</code> Arbitrary set of characters.	a string LCU:filename Note: For more information on using LCU files to mask plain text passwords in the Sterling Connect:Direct Secure Plus CLI, see "Use LCU Files to Encrypt Passwords for Use with the Sterling Connect:Direct Secure Plus CLI" on page 38.

Parameter	Values
KeySeedType=Type of key seed to generate.	dynamic static dynamic—The key seed is generated by the system each time. static—The key seed is generated based on the value defined in the KeySeed parameter. If the value of KeySeed is not changed, the same key pair is generated.
KeySeed=Alphanumeric string at least 32 characters long used to generate the key pair. This parameter is required when static is specified for KeySeedType.	key seed value, at least 32 characters long
StsLimitExportVersion=Identifies If a trading partner is using an earlier export version of Sterling Connect:Direct Secure Plus.	y n

Export STS Keys

The export STSKeys command allows you to export STS keys. The command has the following parameters.

Parameter	Values
dir=Directory in which to export the key file.	valid directory name
name=Name of the remote node record to export keys for. Use wildcard characters to export information about a group of remote node records.	remote node name wildcard Wildcard options are: Asterisk (*)—Any number of characters. For example, kps.* exports all nodes with a name that starts with kps. Question mark (?)—A single character. For example, k?s.* exports kas.* and kbs.*.

Import STS Keys

The import STSKeys command allows you to import STS keys. The command has the following parameter:

Parameter	Values
path=The directory and name of the file that contains the public keys.	valid path value

Automatically Create STS Keys for Remote Node Records

About this task

Before you can begin using Sterling Connect:Direct Secure Plus and the STS protocol for secure communications, you must exchange keys with the trading partner. To facilitate automatic configuration, you can now generate unique keys for a trading partner without the initial exchange of keys.

For information on using scripts to automatically update STS keys, see the “Sample Scripts” on page 38 and “Automation Scripts” on page 87.

To create unique keys for the STS protocol and eliminate the need to exchange keys manually with a trading partner:

Procedure

1. In the .Local node record for both locations, configure the same previous authentication key and the same previous digital signature key.
2. In the .Local node records, configure unique keys in the current authentication and current digital signature keys.
3. Configure every remote node record with the following information:
 - The same current authentication public key
 - The same current signature public key
4. Turn on auto update of public keys.

Turning on auto update causes the public keys in the remote node records to be automatically updated to the value in the current authentication field and the current signature public key field from the remote node.

Completing the procedure results in the following activity:

- When the first connection between the local and remote node occurs, STS authentication is performed based on the values defined in the previous authentication key.
- If authentication is successful, enabling auto update causes each node to automatically import the public authentication key and public signature key from the remote node record from control information exchanged on the communications link. It also updates the remote node record in the local parameters file.
- For subsequent connections, the newly imported keys are used for authentication.

Related concepts:

“Automation Scripts” on page 87

“Sample Scripts” on page 38

Chapter 6. Maintain Sterling Connect:Direct Secure Plus

Sterling Connect:Direct Secure Plus Node List

After you set up node records in Sterling Connect:Direct Secure Plus, you can view all of the nodes and their attributes from the Secure+ Admin Tool Main Window. From the **Node Name** list, you can view the record of an individual node by double-clicking on a node name.

The following table shows all the fields in the Node Name List:

Field Name	Description	Valid Values
Node Name	Displays the node record name.	.Local remote node name .client
Type	Displays the current record type.	L—Local record R—Remote record
Secure+	Displays the status of Sterling Connect:Direct Secure Plus	N—Disabled TLS—TLS protocol SSL—SSL protocol STS—STS protocol *—Default to local node
Override	Displays the status of override. Enable override in the local node to allow remote node records to override the settings in the local node record. If the STS protocol is enabled, enabling override in a remote node allows the values in the COPY statement to override key values in the remote node record.	Y—Enabled N—Disabled *—Default to local node
CipherSuites	Displays the TLS or SSL cipher suites that are enabled for the node record.	Varies, based on the cipher suites enabled.
ClientAuth	Displays the status of client authentication. If the TLS or SSL protocol is used, enabling client authentication means the SNODE verifies the identity of the PNODE.	Y—enabled N—Disabled *—Default to local node
Encryption	Indicates if encryption is enabled in the STS protocol.	Y—Enabled N—Disabled
Signature	Identifies if digital signature is enabled in the STS protocol.	Y—Enabled N—Disabled *—Default to local node

Field Name	Description	Valid Values
LimExpr	Identifies if the Limited Export version is being used by a remote node.	Y—Enabled N—Disabled *—Default to local node
AutoUpdate	Indicates if the option to automatically update key values during communication is enabled.	Y—enabled N—disable *—default to local node
Base Record	Displays the name of the base record for the alias records.	

View Sterling Connect:Direct Secure Plus Parameters File Information

About this task

To view information about the Sterling Connect:Direct Secure Plus parameters file:

Procedure

1. Open the Secure+ Admin Tool.
2. Click **File > Info**. The following fields are displayed in the **File Information** dialog box.

Field Name	Description
Current File	The name of the parameters file opened.
Number of Records	The number of nodes defined in the parameters file.
Number of Updates	How many times the parameters file has been updated.
Last 3 Updates	The name of the last three nodes updated.

View Sterling Connect:Direct Secure Plus Node Record Change History

To view the history of changes to a node record:

Procedure

1. From the Secure+ Admin Tool Main Window, double-click the node record name.
2. Click the **Security Options** tab.
The history of changes is displayed in the **Update History** field.

Disable Sterling Connect:Direct Secure Plus

About this task

You can use this procedure to disable all nodes in a configuration or one remote node.

Procedure

1. Do one of the following:
 - To disable all nodes in a configuration, open the local node record.
 - To disable one node, open the remote node record for that node.
2. Click the **Security Options** tab.
3. Click **Disable Secure+**.
4. Click **OK** to update the node record.

Note: In order to continue Sterling Connect:Direct operations with Sterling Connect:Direct Secure Plus disabled, both trading partners must disable Sterling Connect:Direct Secure Plus.

Delete a Sterling Connect:Direct Secure Plus Remote Node Record

About this task

If a remote node record is no longer defined in the network map, you can remove it from the parameters file. The following procedure deletes nodes that are defined in the Sterling Connect:Direct Secure Plus parameters file but not in the selected network map.

CAUTION:

Do not delete the remote node record that is named for the Sterling Connect:Direct node. It is the base record for the .Local node record. You cannot delete the .Local node record.

Procedure

1. From the Secure+ Admin Tool Main Window, click **File > Sync with Netmap**.
2. Click the network map to use from the pulldown list.
3. Click **OK**.
4. Click **Skip** to move through the **Select Netmap Entries** to the **Add** dialog box.
5. To delete node records, do one of the following:
 - To delete selected node records, highlight the remote nodes to delete and click **Delete Selection**.
 - To delete all remote node records not found in the network map, click **Delete All**.

Resecure Sterling Connect:Direct Secure Plus Parameters and Access Files

About this task

Routinely, or if your access file is compromised, perform the following steps to resecure Sterling Connect:Direct Secure Plus:

CAUTION:

Do not type a new passphrase if an error occurs. If an error occurs while you are resecurating the files, restore the node records from the ACFSave directory. This directory is created after the Rekey Secure+ feature is executed.

Procedure

1. From the Secure+ Admin Tool Main Window, click **File > Rekey Secure+**. The **Rekey Secure+** dialog box is displayed.
2. Type an alphanumeric string of at least 32 characters in the **Passphrase** field. Sterling Connect:Direct Secure Plus uses the passphrase to re-encrypt the Sterling Connect:Direct Secure Plus parameters and access files. You do not have to remember this passphrase value.
3. Click **OK** to accept the new passphrase. Sterling Connect:Direct Secure Plus decrypts and re-encrypts the parameters file and access file.

Change Cipher Suites

About this task

When you activate the TLS or SSL protocol for a node, cipher suites are used to encrypt the data being transmitted. The same cipher suite must be used at both ends of the transmission. Sterling Connect:Direct Secure Plus searches the enabled cipher suite list at both nodes and locates the first cipher suite that is common for communications at both the SNODE and the PNODE.

You enable the cipher suites in the preferred order, or enter them randomly and order them using the following procedure:

Procedure

1. If necessary, open the Secure+ Admin Tool.
2. Double-click the node you want to edit.
3. Click the **TLS/SSL Protocol** tab.
4. If you are configuring a remote node, turn off **Default to Local Node** in the Cipher Suites section.
5. To modify cipher suites, do one of the following:
 - To enable a cipher suite, highlight it in the **Available** list and click **Add**.
 - To remove a cipher suite, highlight it in the **Enabled** list and click **Remove**.
 - To change a cipher suite's priority, highlight the cipher suite and click **Up** or **Down** until it is in the correct priority position.
6. Click **OK**.

Change STS Protocol Encryption Algorithms

About this task

When you activate the STS protocol for a node, it uses algorithms to encrypt the data being transmitted. The same algorithms must be used at both ends of the transmission.

Sterling Connect:Direct Secure Plus searches the enabled algorithm list and locates the first algorithm that is common for communications at both ends. You can enable the algorithms in the preferred order, or enter them randomly and order them using the following procedure:

Procedure

1. If necessary, start the Secure+ Admin Tool.
2. Double-click the node you want to edit.
3. Click the **STS Options** tab.
4. Turn off **Default to Local Node** in the Encryption Algorithms section.
5. To modify algorithms, do one of the following:
 - To enable an algorithm, highlight it in the **Available** list and click **Add**.
 - To remove an algorithm, highlight it in the **Enabled** list and click **Remove**.
 - To change the priority of an algorithm, highlight the item and click Up or Down until it is in the correct priority position.
6. Click **OK**.

Update Keys in an STS-Configured Node Record

About this task

To maintain communications with a trading partner when updating your keys, you must maintain a copy of the previous keys until your trading partner receives the updated keys.

If you and your trading partner both enable the Public Key Auto Updates feature in the parameters file, Sterling Connect:Direct Secure Plus uses the existing keys to establish a session and then creates new keys during authentication. The parameters files for you and your trading partner are automatically updated with the new key values.

To update signature and authentication keys:

Procedure

1. Open the Secure+ Admin Tool.
2. Double-click the node record to update.
3. Click the **STS Protocol** tab.
4. Click the tab for the type of new keys you want to generate: **Authentication Keys** or **Signature Keys**.
5. Click **Generate Key** to create new public keys. A message asks if you want to save the current key pair as previous.
6. Click **Yes** to save the current key pair as the previous. The **Generate Random Number Seed** dialog box is displayed.
7. Do one of the following to generate a key:
 - Type an alphanumeric string of at least 32 characters in the **Random Number Seed** field and click **OK**. Sterling Connect:Direct Secure Plus uses the random number seed value to generate authentication keys.
 - Click **Sample Value**, highlight all of the grid squares, and click **OK** to generate the random number seed.The **Public Keys** window displays the populated **Public Key** field.
8. If necessary, type the expiration date in the **Previous Key Pair Expiration Date** field.
9. Click **OK** to save the changes to the parameters file.

Clear Keys in an STS-Configured Node Record

About this task

To clear the keys in an STS-configured node record:

Procedure

1. Open the Secure+ Admin Tool.
2. Double-click the node record to update with key information.
3. Click the **STS Protocol** tab.
4. Click the tab for the type of new keys you want to clear: **Authentication Keys** or **Signature Keys**.
5. Click **Clear Key**.
6. Click the appropriate button for the type of keys to clear: **Current**, **Previous**, **Both**, or **Cancel**.

Chapter 7. View Statistics

View Statistics

Sterling Connect:Direct logs statistics for Sterling Connect:Direct Process activity. The Sterling Connect:Direct statistics include Sterling Connect:Direct Secure Plus information for a Process. Sterling Connect:Direct information is included in the Process statistics information only when you attach to a Sterling Connect:Direct server.

Sterling Connect:Direct Requester Select Statistics

Use the Select Statistics function of Sterling Connect:Direct Requester to view the information about a Sterling Connect:Direct Process, including statistics information about a particular Process. If Sterling Connect:Direct Secure Plus is enabled, view Sterling Connect:Direct Secure Plus by scrolling to the bottom of the **Statistics Detail** dialog box, in the Session Start Record (SSTR) and Copy Termination Record (CTRC).

For more information on using Sterling Connect:Direct Requester to view statistics information, see the Sterling Connect:Direct for Microsoft Windows Help or the *IBM Sterling Connect:Direct for Microsoft Windows System Guide*.

The Sterling Connect:Direct Secure Plus fields and valid values available using the Select Statistics function of the Sterling Connect:Direct Requester are shown in the following table:

Field Name	Description	Valid Values
Secure+ Enabled	Specifies whether Sterling Connect:Direct Secure Plus is enabled.	Y N
Secure+ Protocol	Which protocol is enabled.	SSL 3.0 STS 1.0 TLS 1.0
Cipher Suite	Displays the cipher suite used during a session. cipher suite name, for example:	SSL_RSA_EXPORT_WITH_RC4_40_MD5
PNode Cipher List	Specifies the encryption algorithms available for the PNODE during the session.	IDEACBC128 TDESCBC112 DESCBC56
PNode Cipher	Specifies the preferred data encryption as specified in the Sterling Connect:Direct Secure Plus parameters file of the PNODE.	Y N algorithm name
SNode Cipher List	Specifies the encryption algorithms available for the SNODE during the session as specified in the Sterling Connect:Direct Secure Plus parameters file of the SNODE.	IDEACBC128 TDESCBC112 DESCBC56
SNode Cipher	Specifies the preferred data encryption algorithm as defined in the Sterling Connect:Direct Secure Plus parameters file of the SNODE.	Y N algorithm name

Field Name	Description	Valid Values
Control Block Cipher	Specifies the algorithm used for encrypting control blocks. This value is determined during authentication when the PNODE and SNODE are merged.	IDEACBC128 TDESCBC112 DESCBC56
Copy Data Cipher	Specifies the encryption method used for encrypting data. The value is determined after the values in the SNODE and the PNODE are merged.	IDEACBC128 TDESCBC112 DESCBC56
PNODE Signature Enabled	Indicates whether digital signatures are enabled for the PNODE. This value is obtained from the parameters file settings. If the COPY statement overrides the Sterling Connect:Direct Secure Plus parameters file, the value from the COPY statement is used.	Y N
SNODE Signature Enabled	Indicates whether digital signatures are enabled for the SNODE. This value is obtained from the Sterling Connect:Direct Secure Plus parameters file settings.	Y N
Signature Enabled	Identifies the digital signature value used for a copy operation. In the Session Start record, this value is the result of the merged value between the PNODE and SNODE. In the Copy Termination record, if the COPY statement overrides the parameters file value, the merged value depends on the value supplied in the COPY statement. (The unprocessed value from the COPY statement is recorded in the Signature Enabled field of the PNODE).	Y N
Current Signature Verified	Indicates whether the current digital signature was verified.	Y N
Previous Signature Verified	Indicates whether the previous digital signature was verified.	Y N

Sterling Connect:Direct CLI Select Statistics Detail

When you use the CLI Select Statistics function to view the information about a Sterling Connect:Direct Process, you see statistics information about a particular Process. The Sterling Connect:Direct fields are shown in bold in the following samples.

Session Start (SSTR) Record

The following sample Session Start Record (SSTR) displays the output of an SSL session:


```

Record Id      => SSTR
Process Name   =>
Process Number => 0
Submitter Id   =>
Start Time     => 15:23:20
Stop Time      => 15:23:21
SNODE         => JKTIB8100
Completion Code => 0
Message Id     => LSMI004I
Message Text   => PNODE session started - remote node &NODE
Secure+ Protocol => SSL 3.0
SSL Cipher Suites => SSL_RSA_WITH_RC4_128_MD5
-----

```

Copy Termination (CTRC) Record

The Copy Termination Record (CTRC) sample below uses the SSL protocol:

```

Record Id      => CTRC
Process Name   => XX
Process Number => 195
Submitter Id   => user1
Start Time     => 15:23:47
Stop Time      => 15:26:32
SNODE         => DLAS8100
Completion Code => 0
Message Id     => SCPA000I
Message Text   => Copy operation successful.
COPY DETAILS: Ckpt=> Y Lkfl=> N Rstr=> N XLat=> N Scmp=> N Ecmp=> N
From node     => S
Src File      => D:\long path
Dest File     => D:\long path
Src CCode     => 0
Src Msgid     => SCPA000I
Bytes Read    => 23592960
Records Read  => 1024
Bytes Sent    => 23791420
RUs Sent     => 30721
Secure+ Protocol =>SSL 3.0
SSL Cipher Suites =>SSL_RSA_WITH_RC4_128_MD5
-----

```

Sterling Connect:Direct CLI Select Process Detail

When you use the CLI select process command to view information about a Sterling Connect:Direct Process, you see statistics about a Process. The following sample Select Process Detail uses the STS protocol. If Sterling Connect:Direct Secure Plus is not enabled, no Sterling Connect:Direct Secure Plus information is displayed:

Process Name	=> XX	Class	=> 32
Process Number	=> 197	Priority	=> 10
Submitter Node	=> DALLAS	PNode	=> DALLAS
Submitter	=> user1	SNode	=> DALLAS
Retain Process	=> N		
Submit Time	=> 15:55:55	ScheduleTime	=>
Submit Date	=> 10/19/2009	ScheduleDate	=>
Queue	=> EXEC		
Process Status	=> EX		
Message Text	=>		
Function	=> COPY		
Step Name	=> TWO		
Type	=> Send		
File Bytes	=> 3202560	File Recs	=> 0
Xmit Bytes	=> 3247926	Xmit Buffers	=> 0
Secure+ Protocol	=> STS 1.0		
Ctrl Block Cipher	=> TDESCB112		
Copy Data Cipher	=> IDEACBC128		
Signature Enabled	=> Y		

Chapter 8. Audits

Sterling Connect:Direct Secure Plus Parameters File Auditing

The Secure+ Admin Tool and the Sterling Connect:Direct Secure Plus Command Line Interface log changes made to the Sterling Connect:Direct Secure Plus parameters file.

The following events are logged:

- Application Startup
- Init Parmfile
- Open Parmfile
- Sync Netmap
- Rekey Parmfile
- Create Node
- Update Node
- Delete Node

The parameters file logging feature has the following operational characteristics:

- The logging feature is always enabled and cannot be disabled.
- If errors occur when the log is being updated, the application terminates.
- Each log entry contains a timestamp, user ID, and a description of the action/event.
- When an existing node is updated, any changed fields are reported.
- When a node is created or deleted, the values of all non-empty fields are reported.
- Any commands that modify a node are logged.

Note: The certificates used by Sterling Connect:Direct Secure Plus are individual files that can be stored anywhere on the system. As a result, the logging feature cannot detect when existing certificate files are modified. Sterling Connect:Direct Secure Plus only stores the certificate path name and detects changes to this field only.

Accessing Parameters File Audit Logs

The parameters file audit logs are stored in a dedicated directory, ..\secure+\log. The log file naming convention is SP[YYYY][MM][DD].001 (using local time), and the contents of a log file are limited to a single calendar date. You can view these log files using any text editor. Log files are not deleted by Sterling Connect:Direct Secure Plus.

Parameters File Audit Log Entries

Each audit log has the following header:

[YYYYMMDD] [HH:MM:SS:mmm] [userid]

When a parameter file is created or opened, an ID is generated that associates the change with the node being updated, as shown in the following:

[YYYYMMDD] [HH:MM:SS:mmm] [userid] [ParmFileID]

The following fields may appear in a create, update, or delete audit record.

Field Name	Description
Name	Name of the node
BaseRecord	Name of the base record
Type	Record type of local, remote, or alias
Protocol	Enables Sterling Connect:Direct Secure Plus protocol
Override	Enables overriding the current node
AuthTimeOut	Authentication timeout
StsEnableSig	Enable STS signatures
StsEnableAutoUpdate	Enable STS auto update remote public keys
StsEnableEnc	Enable STS copy encryption stream
StsEncAlgs	Enable STS control block encryption streams
StsAuthLocalKey	STS authentication local public key
StsAuthRemoteKey	STS authentication remote public key
StsPrevAuthKey	STS authentication previous private key (masked)
StsPrevAuthKeyExpDateTime	STS authentication previous private key expiration date
StsSigLocalKey	STS signature local public key
StsSigRemoteKey	STS signature remote public key
StsPrevSigKey	STS signature previous private key (masked)
StsPrevSigKeyExpDateTime	STS signature previous private key expiration date
SslTlsTrustedRootCertFile	Pathname to trusted roots file
SslTlsCertFile	Pathname to key certificate file
SslTlsCertPassphrase	Key certificate passphrase (masked)
SslTlsEnableClientAuth	Enable client authentication
SslTlsCertCommonName	Common name of the remote certificate to verify
SslTlsEnableCipher	List of SSL/TLS cipher suites
SslTlsSeaEnable	Enable external authentication
SeaCertValDef	External authentication validation definition
SeaHost	External authentication host name
SeaPort	External Authentication port number

Parameters File Audit Log Error Reporting

Errors are reported for the following logging functions: open log, write log, and lock log. If an error occurs during one of these functions, an error message is displayed and the application is terminated. The lock function times out after 30 seconds. Typically, Secure+ Admin Tool or the CLI hold the lock for less than one second per update.

Sterling Connect:Direct Secure Plus Certificate Auditing

In an SSL/TLS session, audit information about the identity certificate and its signing certificate is logged in the statistics log in the Session Start (SSTR) and Copy Termination (CTRC) records. The audit information is included in the response data from a Select Statistics command in the SSTR and CTRC records.

In an SSL/TLS session, the PNODE (client) always logs the audit information. The SNODE (server) only logs the information when client authentication is enabled. For logging to occur, the session handshake must succeed and progress to the point of logging the SSTR and CTRC records.

Certificate Audit Log Entries

The audit consists of the subject name and serial number of the identity and its signing certificate. The identity certificate also contains an issuer attribute, which is identical to the signing certificate subject name. Although many signing certificates may exist between the identity and final root certificate, the audit includes only the last two certificates in a chain: an intermediate certificate and an end certificate.

In the SSTR and CTRC records, the CERT contains the common name and serial number of the key certificate, and the CERI contains the common name of the issuer and the serial number of an intermediate or root CA. They may also contain the certificate serial number, for example:

```
CERT=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/CN=Test ID/SN=99c0ce01382e6c83) |  
CERI=(/C=US/ST=MA/L=Marshfield/O=test.org/CN=root CA/SN=da870666bbfb5538)
```

Sterling Connect:Direct Secure Plus certificate audits may contain the following fields:

Field Name	Abbreviation	Max Lengths (RFC 2459)
Common Name	CN	64
Country	C	2
Locality	L	128
State	ST	128
Organization	O	64
Organization Unit	OU	64
Email Address	emailAddress	128
Serial Number	SN	128 (estimated)

Accessing Certificate Audit Logs

Certificate audit information located in the SSTR and CTRC records cannot be accessed directly using Sterling Connect:Direct Requester or Sterling Connect:Direct Browser User Interface. To access certificate information, you can issue a query directly to the database or use an SDK-based or JAI-based program to issue a Select Statistics command. The response to the Select Statistics command contains the AuditInfo field of the statistics records, including the SSTR and CTRC records. This field contains certificate audit information.

The following example was generated using a database query.

```
'2009-05-21 14:50:27', 2, 'SSTR', 'CAEV', '', 0, '2009-05-21 14:50:26', '2009-05-21
14:50:27', '', '', 'JLYON-XP.4500', 0,
'MSGI=LSMI004I|SBST=(&NODE=JLYON-XP.4500)|PNOD=JLYON-XP.4500|CSPE=Y|CSPP=TLSv1|CSPS=
TLS_RSA_WITH_AES_256_CBC_SHA|
CERT=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/
CN=Example Test ID/SN=a9febbeb4f59d446)|
CERI=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/CN=Example
IntermediateCA/SN=a69634a8a7830268)|STSD=2|TZDI=-14400|'
'2009-05-21 14:50:28', 2, 'CTRC', 'CAPR', 'SAMPLE', 1, '2009-05-21 14:50:27',
'2009-05-21 14:50:28', 'JLYON-XP.4500', 'jlyon', 'JLYON-XP.4500', 0,
'MSGI=SCPA000I|LCCD=0|LMSG=SCPA000I|OCCD=0|OMSG=SCPA000I|PNAM=SAMPLE|PNUM=1|SNAM=STE
P1|SBND=JLYON-XP.4500|SBID=jlyon|PNOD=JLYON-XP.4500|SNOD=JLYON-XP.4500|LNOD=P|FROM=P
XLAT=N|ECZI=N|ECMP=N|SCMP=N|OERR=N|CKPT=Y|LKFL=N|RSTR=N|RUSZ=65535|PACC=|SACC=|PPMN
=|SFIL=C:\Program Files\Sterling Commerce\Connect Direct
v4.6.00\Server\Process\Sample.html|SDS1= |SDS2= |SDS3=
|SFSZ=0|SBYR=861|SRCR=1|SBYX=863|SRUX=1|SNVL=-1|SVOL=|DFIL=C:\Program Files\Sterling
Commerce\Connect Direct v4.5.00\Server\Process\Verify.html|PPMN=|DDS1=R|DDS2= |DDS3=
|DBYW=861|DRCW=1|DBYX=863|DRUX=1|DNVL=0|DVOL=|CSPE=Y|CSPP=TLSv1|CSPS=TLS_RSA_WITH_AE
S_256_CBC_SHA|CERT=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/CN=Example Test
ID/SN=a9febbeb4f59d446)|CERI=(/C=US/ST=MA/L=Marshfield/O=test.org/OU=Dev/
CN=Example Intermediate CA/SN=a69634a8a7830268)
|PCRC=N|ETMC=60|ETMK=10|ETMU=0|STSD=2|TZDI=-14400|'
```

Certificate Audit Log Error Reporting

If an error occurs when the subject name is extracted from the identity (CERT) or issuer's (CERI) certificates, the following message ID is logged:

```
CERT=(MSGI=CSPA310E)|CERI=(MSGI=CSPA310E)
```

Only the message ID is displayed with the CERT or CERI tokens; the standard Sterling Connect:Direct error function is not used. After the error occurs, the session continues.

Chapter 9. Troubleshoot Sterling Connect:Direct Secure Plus

Troubleshooting

Use the following table to help troubleshoot problems with Sterling Connect:Direct Secure Plus.

Problem	Possible Cause	Solution
Sterling Connect:Direct Secure Plus features are enabled in the parameters file, but the statistics record indicates that the functions are disabled.	The Sterling Connect:Direct network maps do not contain entries for the PNODE and SNODE.	Verify that the netmap entries for both the PNODE and the SNODE exist.
Running a Process with a remote node fails with an authentication error.	Unique public/private key pairs are generated for the remote node record and the .Local node record is set to Enable Override=N.	Change the .Local node record to Enable Override=Y.
The Sterling Connect:Direct Secure Plus parameter, ENCRYPT.DATA, specified from the COPY statement causes the copy step to fail with error message CSPA080E.	The algorithm name used in the COPY statement is not in the supported algorithm list for both nodes.	Verify that the algorithm name in the COPY statement is in the supported algorithm list for both nodes.
Sterling Connect:Direct Secure Plus is installed, but error message CSPA001E occurs on non-Sterling Connect:Direct Secure Plus transfers.	Remote node records do not exist.	<ul style="list-style-type: none"> • A remote node record must exist for every node in the netmap. Use the Sync with Netmap feature to create any missing nodes. • Disable Sterling Connect:Direct Secure Plus by clicking Disable Secure+ in the .Local node record.
Signature verification fails with error message CSPA002E.	Configuration settings missing or incorrect.	<ul style="list-style-type: none"> • If this is a non-secure node, make sure the remote node record has Disable Secure+ selected. • Check the Sterling Connect:Direct Secure Plus settings for the node.
Signature verification fails with error message CSPA003E, CSPA004E, or CSPA005E.	<ul style="list-style-type: none"> • Configuration settings missing or incorrect. • A security attack in progress. 	<ul style="list-style-type: none"> • Correct the STS settings for the remote node. • Execute standard operating procedure for investigating security violation.

Problem	Possible Cause	Solution
Signature verification fails with error message CSPA007E.	Expired Signature Key Pair. Date exceeded or keys have been changed.	If Auto Update is disabled, check the expiration date for the signature key pair for both nodes. Check the update history log on both nodes for the last change to the record. Verify that the signature public key is correct for both nodes.
Running a Process with a remote node fails with an authentication error, CSPA008E.	Authentication Key Pair Expiration Date exceeded or keys have been changed.	If Auto Update is disabled, check the authentication previous key pair expiration date for both nodes. Check the update history log on both nodes for the last change to the record. Verify the authentication public key is correct for both nodes.
Strong authentication fails with the error, CSPA010E.	<ul style="list-style-type: none"> The time allowed for strong authentication expired. A security attack in progress. 	<ul style="list-style-type: none"> Increase the timeout value. Execute standard operating procedure for investigating security violation.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA011E.	An illegal attempt to override Sterling Connect:Direct Secure Plus parameters.	<ul style="list-style-type: none"> Turn on Enable Override in the remote node record to allow the COPY statement to override the node settings. Check the COPY statement and remove the override statements.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA014E.	Sterling Connect:Direct Secure Plus cannot read the remote node definition.	Check the remote node definition settings.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA016E.	Sterling Connect:Direct Secure Plus is not enabled in the local node definition.	Make sure Sterling Connect:Direct Secure Plus is enabled for the local node.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA019E.	Error generating digital signature.	<ul style="list-style-type: none"> Resubmit the Process. Call IBM Support.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA077E.	The COPY statement requested Sterling Connect:Direct Secure Plus parameters but Sterling Connect:Direct Secure Plus is not configured.	Remove the SECURE= parameter from the COPY statement.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA079E.	Invalid encryption algorithm identified in COPY statement.	Change the ENC.DATA parameter and specify one of the following values: Y, N, IDEACBC128, TDESCBC112, or DESCBC56 and resubmit the Process.

Problem	Possible Cause	Solution
Sterling Connect:Direct Secure Plus session fails with the error, CSPA080E.	No common algorithms are available for both nodes.	Verify the algorithm list for both nodes contains at least one common algorithm name.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA091E.	Session attempted but remote node is not configured.	Make sure both nodes are defined and Enable STS Protocol is selected for the remote node record.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA200E.	Both nodes are not configured for the same protocol.	<ul style="list-style-type: none"> • Check the protocol setting at both sites and verify that the same protocol is configured at each site. • If necessary, edit the remote node record and activate the STS protocol.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA202E.	SSL or TLS protocol handshake failed.	Edit the cipher suite list and add a cipher suite used by the trading partner.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA203E or CSPA204E.	The SSL or TLS protocol could not validate the server's certificate.	Make sure the certificate information is typed into the node record.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA205E.	A trading partner is not using TCP/IP for communication.	Make sure that both ends of the communication use TCP/IP.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA206E.	The SSL or TLS protocol could not validate the server's certificate.	Make sure the certificate information is entered into the node record.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA208E.	The common name in the certificate received does not match the Sterling Connect:Direct Secure Plus configuration.	Make sure the certificate common name is spelled correctly and uses the same case as that in the certificate.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA209E.	The certificate has expired or is invalid.	Obtain a new certificate and reconfigure the node record.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA210E.	The COPY statement attempts to override settings in the STS protocol.	If desired, change the Process statement and remove the COPY override options.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA211E.	The remote trading partner failed to send a certificate.	Notify the trading partner that a certificate is required.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA280E.	The trusted root certificate could not be loaded.	Check the local node configuration and make sure the location of the trusted root certificate is correctly identified.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA281E.	The trusted root certificate is empty.	Check the local node configuration and make sure the location of the trusted root certificate is correctly identified.

Problem	Possible Cause	Solution
Sterling Connect:Direct Secure Plus session fails with the error, CSPA282E.	The user certificate file cannot be loaded.	Check the local node configuration and make sure the location of the user certificate file is correctly identified.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA303E.	The parameters files have not been initialized.	Run the Admin Tool to initialize the parameters files.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA309E.	The SSL library failed during the handshake.	Examine all related errors to determine the cause of the failure.
Sterling Connect:Direct Secure Plus session fails with the error, CSPA311E.	Certificate validation failed.	Verify that the root certificate is properly configured. An alternate certificate may be required.

Chapter 10. Configuration Worksheets

Local Node Security Feature Definition Worksheet

Record the security definitions for the local Sterling Connect:Direct node.

Local Node Name	
Configured Security Functions <ul style="list-style-type: none">• Enable TLS protocol (Yes No)• Enable SSL protocol (Yes No)• Authorization Timeout• Trusted Root Certificate File location• Certificate File• Certificate Passphrase• Cipher Suite(s) Enabled	
External Authentication <ul style="list-style-type: none">• Enable External Authentication (Yes No)• Certificate Validation Definition	

Remote Node Security Feature Definition Worksheet

Make a copy of this worksheet for each remote node defined in the Sterling Connect:Direct Secure Plus parameters file that you are configuring for Sterling Connect:Direct Secure Plus operations. Record the security feature definitions for a remote node record on this worksheet.

Remote Node Name	
Security Options <ul style="list-style-type: none">• Protocol defined in the .Local node record (TLS SSL)• Is the remote node using the protocol defined in the .Local node record? (Y N)<ul style="list-style-type: none">– If you answered No to the question above, identify the protocol to use for the Remote Node (TLS SSL)– Enable TLS protocol (Y N)– Enable SSL protocol (Y N)– Enable STS protocol (Y N)– To use the same protocol defined in the local node, select Default to Local Node.– Enable Override (Y N) Note: To use the COPY statement to override settings in the parameters file for STS-enabled nodes, set Enable Override for the remote node. The COPY statement cannot override settings in SSL-enabled or TLS-enabled remote nodes.– Authorization Timeout	

<p>TLS or SSL Protocol Functions</p> <ul style="list-style-type: none"> • Trusted Root Certificate File location • Certificate File • Certificate Passphrase • Cipher Suite(s) Enabled • Enable Client Authentication (Y N Default to local node) • Certificate Common Name) <p>Note: If you want to add a second level of security, enable client authentication for the remote node and type the certificate common name.</p>	
<p>External Authentication</p> <ul style="list-style-type: none"> • Enable External Authentication (Y N Default to local node) • Certificate Validation Definition 	

Chapter 11. Certificate File Layout

Certificate File Layout

The SSL and TLS security protocols use a secure server RSA X.509V3 certificate to authenticate your site to any client that accesses the server and provides a way for the client to initiate a secure session. When you obtain a certificate file from a certificate authority or create a self-signed certificate, a trusted root certificate file and key file are created.

Sterling Connect:Direct Secure Plus uses two certificate files to initiate TLS or SSL sessions: a trusted root certificate file and a key certificate file.

When you obtain a root certificate from a certificate authority, you receive a trusted root certificate file. To configure Sterling Connect:Direct Secure Plus, add the name and location of the trusted root certificate file to the node record using the Secure+ Admin Tool.

A sample trusted root certificate file called `trusted.txt` is installed in the `Secure+\certificates` directory when you install Sterling Connect:Direct Secure Plus. Use any text editor to add or delete certificate information to this file. In simple configurations, only one trusted root certificate file is used. In more sophisticated configurations, you may associate individual certificate files with one or more node records.

When you use a certificate signing request (CSR) tool, such as the Sterling Certificate Wizard, you do not need to change the contents of the key certificate file. This is created for you by the Sterling Certificate Wizard.

If you set up your own PKI infrastructure, you may chain more than two certificates, including a CA root certificate, one or more intermediate CA certificates, and an identity certificate. You can create chained certificates using one of the following methods:

- Using the Local Key Certificate File—In a chain of two certificates, the local key certificate file contains a private key and an identity certificate. In a longer chain, the key certificate file contains the private key and the identity key, followed by the intermediate CA certificates.
- Using the Remote Trusted File— In a chain of two certificates, the remote trusted file contains the CA root certificate. In a longer chain, the remote trusted file contains the CA root certificate and all the intermediate CA certificates.

General Object Format

All objects are formatted in the Privacy Enhanced Mail (PEM) style, beginning with a line in the following format:

```
-----BEGIN <object>-----
```

and ending with a line in this format:

```
-----END <object>-----
```

In the above examples, <object> is a placeholder for the name of the object type: CERTIFICATE or ENCRYPTED PRIVATE KEY.

Certificate Format

A certificate is encoded as a general object with the identifier string CERTIFICATE or X.509 CERTIFICATE. The base64 data encodes a Bit Error Rate (BER)-encoded X.509 certificate. This is the same format used for PEM. Anyone who provides or understands PEM-format certificates can accommodate the certificate format. For example, VeriSign commonly fulfills certificate requests with certificates in this format, SSLeay supports them, and SSL servers understand them. Most browsers support this format for importing root CA certificates.

Private Key Format

A private key is encoded as a general object with the identifier string ENCRYPTED PRIVATE KEY. The base64 data encodes a BER-encoded PKCS#8 Private Key object. The passphrase associated with the Private Key is required for Sterling Connect:Direct Secure Plus and is stored in the Sterling Connect:Direct Secure Plus parameters file. Additional encryption is used to prevent the passphrase from being discovered.

Sample Certificate Files

In the sample user certificate below, a private key is followed by the server certificate, which is followed by the root certificate.

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIICCDAAaBgqhkiG9w0BBQMwDQIIIfYyAEFKaEECAQUEggHozdmgGz7zbC1mcJ2r
.
.
.
IGpupStY5rLqqQ5gwLn45UWgzy6DM96CQg6+Dyn0N9d1M5lIg2w1nUwE8vI=
-----END ENCRYPTED PRIVATE KEY-----
User/Server Certificate
-----BEGIN CERTIFICATE-----
MIICUDCCAadoCBDAmitYwDQYJKoZIhvcNAQEEBQAwgY8xCzAJBgNVBAYTA1VTMRMw
.
.
.
iK1sPBRbNdq5cNIuIfPS8emrYMs=
-----END CERTIFICATE-----
// Final Root Certificate (optional)
-----BEGIN CERTIFICATE-----
MIICUDCCAadoCBDAmitYwDQYJKoZIhvcNAQEEBQAwgY8xCzAJBgNVBAYTA1VTMRMw
.
.
.
iK1sPBRbNdq5cNIuIfPS8emrYMs=
-----END CERTIFICATE-----
```

In the sample root certificate below, the trusted.txt file contains a list of trusted root certificates.

```
RSA Commercial CA - exp. Dec 31, 2003
-----BEGIN CERTIFICATE-----
MIICUDCCAdoCBDAmitYwDQYJKoZIhvcNAQEEBQAwwY8xCzAJBgNVBAYTA1VTMRMw
.
.
.
iK1sPBRbNdg5cNIuIfPS8emrYMs=
-----END CERTIFICATE-----
RSA Commercial CA - exp. Dec 31, 2010
-----BEGIN CERTIFICATE-----
MIICUDCCAdoCBDAmitYwDQYJKoZIhvcNAQEEBQAwwY8xCzAJBgNVBAYTA1VTMRMw
.
.
.
iK1sPBRbNdg5cNIuIfPS8emrYMs=
-----END CERTIFICATE-----
```

Chapter 12. Test Sterling Connect:Direct Secure Plus with STS

Set Up the Local and Remote Node Records for Testing

About this task

Before using the STS protocol in a production environment, test the installation to ensure Sterling Connect:Direct Secure Plus is operational.

The parameters file is initially populated with two node records: the .Local record and a remote record for the associated + node. The remote record for this local node is created to support loopback (PNODE=SNODE) processing to test Sterling Connect:Direct Secure Plus. To test the installation, you need to configure the sample .Local node record and remote node record.

Procedure

1. Click **Start > Programs > Sterling Connect Direct v4.6.00 > Secure+ Admin Tool**.
2. To configure the .Local node record, double-click **.Local**. The **Edit Record** dialog box opens to the **Security Options** tab.
3. Change the Security Options values (for the local or remote node record) as follows:

Field	Local Node Setting	Remote Node Setting
Secure+ Protocol	Enable STS Protocol	Enable STS Protocol
Node or Copy Statement Override	Enable Override	Default to Local Node
Authentication Timeout	90	90

4. Click the **STS Protocol** tab. The **Edit Record - STS Protocol** window displays the **STS Options**.
5. On the **STS Options** tab, click the following values:

Field	Local Node Setting	Remote Node Setting
Enable Digital Signatures	Yes	Default to Local Node
Enable Public Key Auto Updates	No	Default to Local Node
Enable Encryption	Yes	Default to Local Node

6. Click the **Authentication Keys** tab.
7. Generate the authentication keys as follows:
 - a. Click the **Generate Key** button. The **Generate Random Number Seed** dialog box is displayed.
 - b. Type an alphanumeric string of at least 32 characters in the **Random Number Seed** field and click **OK**.
8. Click the **Signature Keys** tab.
9. Generate the signature keys as follows:

- a. Click the **Generate Key** button. The **Generate Random Number Seed** dialog box is displayed.
 - b. Type an alphanumeric string of at least 32 characters in the **Random Number Seed** field and click OK. The Secure+ Admin Tool generates the Signature keys and populates the **Local Public Key** field.
10. Click **OK** to save the changes.

Exchange Public Keys

To complete the setup for testing the STS protocol, you need to exchange public keys. In a production environment, you create and configure a remote node for each trading partner. Then you export keys from the remote node and send this information to the trading partner. Finally, you import keys that the trading partner sends to you into the remote node record.

Export Keys

About this task

The first step to exchange keys is to export the keys of the remote node. Exporting keys copies the public keys to a file that can be sent to the trading partner.

To export the remote node key:

Procedure

1. Create a specific directory for storing the public key files.
2. From the Secure+ Admin Tool Main Window, highlight the remote node record. Then click **Key Management > Export Selection**. The **Select Export Destination Directory** dialog box is displayed.
3. Click **Select** and save the key file to the directory you created in step 1. A file with an **sxp** extension is created for the remote node entry for this local node. The file is named after the local node.

Import Keys

About this task

When you are ready to import keys, have your trading partner send this file to your production environment. However, for test purposes, use the information you exported from the Export Keys process.

Procedure

1. From the Secure+ Admin Tool Main Window, highlight the remote node record and click **Key Management > Import Public Keys**. The **Select Import Source File** dialog box is displayed.
2. Go to the directory where you saved the **sxp** file.
3. Highlight the **sxp** file for this node and click **Import**. The key is imported to the remote node record.
4. Ensure that the remote keys are imported by viewing the **Remote Public Key** field located in the **Authentication Keys** tab and **Signature Keys** tab.

Note: The remote node public keys are not displayed in the .Local node record, but are displayed in the base remote node record.

Validate the Configuration

About this task

Perform this procedure to ensure that the nodes have been properly configured. The validation process checks each node to ensure that all necessary options have been defined and keys have been exchanged.

To validate the parameters file:

Procedure

1. In the Secure+ Admin Tool, click **File > Validate Secure+**. The **Secure+ Admin Tool - Validation Results** window is displayed.
If the parameters file is not correctly configured, warning and error messages are displayed.
2. Read each warning message. To correct each warning or error reported, go back to the parameters file and make changes as needed.

Note: Warning messages do not always mean that the parameters file is incorrectly configured. Some warning messages are informational only.

3. Click **Close** to close the **Validation Results** window.

Exchange Data and Verify Results

To exchange data and verify the results, submit the sample Process that is provided with Sterling Connect:Direct.

To verify the success of the sample Process and review the Sterling Connect:Direct Secure Plus statistics for the session, refer to *IBM Sterling Connect:Direct for Microsoft Windows Getting Started Guide*.

Chapter 13. Automation Scripts

Automation Scripts

The following scripts are provided as models for creating custom scripts to define your Sterling Connect:Direct Secure Plus environment and automate the implementation of it. To prevent any loss of data, you cannot run the scripts, but you can save them with a different name and modify them to suit your needs.

Automate Sterling Connect:Direct Secure Plus Implementation for the STS Protocol

The two sample scripts in this section illustrate using the CLI to automate a mass implementation of Sterling Connect:Direct Secure Plus and add a remote node, respectively. Both scripts illustrate using statically generated public keys.

Use Statically Generated Public Keys for Automated Mass Implementation

The `spcust_sample1` script demonstrates a mass implementation of Sterling Connect:Direct Secure Plus using statically generated public keys.

```
REM spcust_sample1.cmd contains an example of configuring
REM Secure+ with the Secure+ CLI. The example demonstrates
REM the use of statically generated public keys to facilitate
REM an automated mass rollout of Secure+.
REM
REM Variables
REM The return code.
REM spcli.cmd returns the highest return code of the commands
REM it executed. Possible return codes and their meanings are
REM 0 success
REM 4 warning
REM 8 error
REM 16 fatal error
set cdInstallDir=
set spDir=%cdInstallDir%\Server\Secure+
pushd "%spDir%"
REM Main script
REM
echo.
echo This script has been prevented from running because it will alter
echo the configuration of Secure+. Before removing this warning and its
echo exit call, please modify the script so that it carries out only
echo desired modifications to the configuration of Secure+.
echo.
goto :EOF
all :initCustom
call :invokeCLI
call :terminateCustom
REM End of main script
goto :EOF
REM
REM Functions
REM
REM Custom initialization logic written by customer.
REM :initCustom
```

(Continued)

```
REM Customer adds custom initialization code here.
echo Init custom...
echo.
REM del /F "%spDir%\Nodes"
REM End ofinitCustom
goto :EOF
REM Invoke CLI to configure Secure+.
:invokeCLI
set tempFile=clicmds.txt
echo ;
>%tempFile%
echo ; -- Create a new parmfile
>>%tempFile%
echo ;
>>%tempFile%
echo init parmfile
>>%tempFile%
echo localnode=KCHEN-TW-4500
>>%tempFile%
echo path="%spDir%\Nodes"
>>%tempFile%
echo passphrase=12345678901234567890123456789012
>>%tempFile%
echo ;
>>%tempFile%
echo display info
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Synch with netmap
>>%tempFile%
echo ;
>>%tempFile%
echo sync netmap
>>%tempFile%
echo path=v4.3.00\KCHEN-TW-4500
>>%tempFile%
echo name=*
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Update localnode, enable protocol and add keys.
>>%tempFile%
echo ;
>>%tempFile%
echo update localnode
```

(Continued)

```
>>%tempFile%
    echo protocol=sts
>>%tempFile%
    echo override=y
>>%tempFile%
    echo STSEnableSig=y
>>%tempFile%
    echo STSEnableEnc=y
>%tempFile%
echo STSEnableAutoUpdate=y
>>%tempFile%
echo stsAuthlocalkey=set
>>%tempFile%
echo StsAuthKeyPairFile=keypairfile
>>%tempFile%
echo StsAuthKeyPairFilePassphrase=secret
>%tempFile%
echo stsSiglocalkey=set
>>%tempFile%
echo StsSigKeyPairFile=keypairfile
>>%tempFile%
echo StsSigKeyPairFilePassphrase=secret
>>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Display localnode
>>%tempFile%
echo ;
>>%tempFile%
echo display localnode
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Update remotenode records
>>%tempFile%
echo ;
>>%tempFile%
echo update remotenode
>>%tempFile%
echo name=*
>>%tempFile%
echo STSAuthRemoteKey=0200.5F8E.F736.1E38.85C7.2BD0.E312.BD4D.FBC5.A2D3.8E1F
>>%tempFile%
echo
STSSigRemoteKey=0200.5F8E.F736.1E38.85C7.2BD0.E312.BD4D.FBC5.A2D3.8E1F
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
```

(Continued)

```
echo ; -- Verify localnode does not have remote keys
>>%tempFile%
echo ;
  >>%tempFile%
echo display localnode
  >>%tempFile%
echo ;
>>%tempFile%
  echo ;
>>%tempFile%
echo ; -- Verify all remote nodes have remote pub keys set
>>%tempFile%
echo ;
>>%tempFile%
  echo display remotenode
  >>%tempFile%
  echo name=*
>>%tempFile%
  echo ;
>>%tempFile%
echo ;
>>%tempFile%
  echo ; -- Update localnode, enable protocol and add keys.
>>%tempFile%
echo ;
>>%tempFile%
echo update localnode
>>%tempFile%
echo stsauthlocalkey=gen
  >>%tempFile%
echo stsauthkeyseedType=dynamic
>>%tempFile%
echo STSPrevAuthKeyExpDateTime=2005:01:01-20:13:44
>>%tempFile%
  echo stssiglocalkey=gen
>>%tempFile%
echo stssigkeyseedType=dynamic
>>%tempFile%
  echo STSPrevSigKeyExpDateTime=2005:01:01-20:13:45
  >>%tempFile%
echo ;
>>%tempFile%
  echo display localnode;
>>%tempFile%
  echo ; -- Validate parmfile
  >>%tempFile%
echo ;
>>%tempFile%
  echo validate parmfile
  >>%tempFile%
echo ;
```


(Continued)

```
>>%tempFile%
echo quit;
>>%tempFile%
call "%spDir%\spcli.cmd" -e 8 -li y < %tempFile%
set RC=%ERRORLEVEL%
del %tempFile%
REM End of invokeCLI
goto :EOF
REM Custom termination logic written by customer.
REM :terminateCustom
REM Customer adds custom termination code here.
REM For example, E-mail standard out log for review.
REM Send error messages to system monitoring facility.
echo.
echo Custom Terminating with errorlevel of %RC%
echo.
REM End of terminateCustom
goto :EOF
popd
```

Related tasks:

“Automatically Create STS Keys for Remote Node Records” on page 56

Automatically Add a Remote Node to a Sterling Connect:Direct Network

The spcust_sample2 script illustrates how to automate adding a remote node to a Sterling Connect:Direct network.

```
@echo off
REM spcust_sample2.cmd contains an example of configuring
REM Secure+ with the Secure+ CLI. The example demonstrates
REM the use of statically generated public keys to
REM facilitate the automated addition of a new node to a
REM Connect:Direct network. In this example, the name of
REM the new node is cd-ultra3600.
REM A script based on this sample is used to
REM configure the existing nodes in the new node's Netmap.
REM The new node itself is configured with a script based REM
REM on spcust_sample1.cmd. In order for the addition to be
REM successful, the two scripts must use the same seeds
REM to generate keys.
REM
```

(Continued)

```
REM Variables
REM The return code.
REM spcli.cmd returns the highest return code of the commands
REM it executed. Possible return codes and their meanings are
REM 0 success
REM 4 warning
REM 8 error
REM 16 fatal error
set cdInstallDir=
set spDir=%cdInstallDir%\Server\Secure+
pushd "%spDir%"
REM Main script
echo This script has been prevented from running because it will alter
echo The configuration of Secure+. Before removing this warning and its
echo exit call, please modify the script so that it carries out only
echo desired modifications to the configuration of Secure+.
goto :EOF
call :initCustom
call :invokeCLI
call :terminateCustom
REM End of main script
goto :EOF
REM Functions
REM Custom initialization logic written by customer.
:initCustom
REM Customer adds custom initialization code here.
echo Init custom...
echo.
REM End of initCustom
goto :EOF
REM Invoke CLI to configure Secure+.
invokeCLI
set tempFile=clicmds.txt
echo ;
>%tempFile%
echo ;
>>%tempFile%
echo ; -- Synch with netmap
>>%tempFile%
echo ;
>>%tempFile%
echo sync netmap
>>%tempFile%
echo path=v4.6.00\KCHEN-TW-4600
>>%tempFile%
echo name=*
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
```

(Continued)

```
echo ; -- Update remotenode records
>>%tempFile% echo ;
>>%tempFile%
echo update remotenode
  >>%tempFile%
  echo name=salmon3600
>>%tempFile%
echo stsAuthLocalkey=set
  >>%tempFile%
echo StsAuthKeyPairFile=keypairfile
  >>%tempFile%
echo StsAuthKeyPairFilePassphrase=secret
  >>%tempFile%
echo stsSigLocalkey=set
>>%tempFile%
echo StsSigKeyPairFile=keypairfile
>>%tempFile%
  echo StsSigKeyPairFilePassphrase=secret
>>%tempFile%
echo STSAuthRemoteKey=0200.5F8E.F736.1E38.85C7.2BD0.E312.BD4D.FBC5.A2D3.8E1F
>>%tempFile%
echo STSSigRemoteKey=0200.5F8E.F736.1E38.85C7.2BD0.E312.BD4D.FBC5.A2D3.8E1F
>>%tempFile%
echo ;
  >>%tempFile%
echo display remotenode name=salmon3600
>>%tempFile%
  echo ;
>>%tempFile%
  echo validate parmfile
  >>%tempFile%
  echo ;
>>%tempFile%
echo quit;
>>%tempFile%
  call "%spDir%\spcli.cmd" -p "%spDir%\Nodes" -e 8 -li y < %tempFile%
set RC=%ERRORLEVEL%
del %tempFile%
REM End of invokeCLI
goto :EOF
REM Custom termination logic written by customer.
:terminateCustom
  REM Customer adds custom termination code here.
  REM For example, E-mail standard out log for review.
  REM Send error messages to system monitoring facility.
  echo Custom Terminating with errorlevel of %RC%
REM End of terminateCustom
goto :EOF
popd
```

Configure Sterling Connect:Direct Secure Plus to Use the SSL or TLS Protocol

The `spcust_sample3` script demonstrates using the CLI to configure Sterling Connect:Direct Secure Plus to use the SSL or TLS protocol with the trusted root file, key certificates, and ciphers.

```
@echo off
REM spcust_sample3.cmd contains an example of configuring
REM Secure+ to use SSL or TLS protocols with the Secure+ CLI.
REM The example demonstrates the configuration of Secure+
REM with the trusted root and key certificates and ciphers
REM Variables
REM The return code.
REM spcli.cmd returns the highest return code of the commands
REM it executed. Possible return codes and their meanings are
REM 0 success
REM 4 warning
REM 8 error
REM 16 fatal error
echo ; -- Create a new parmfile
>>%tempFile%
echo ;
>>%tempFile%
echo init parmfile
>>%tempFile%
echo localnode=KCHEN-TW
>>%tempFile%
echo path="%spDir%\Nodes"
>>%tempFile%
echo passphrase=12345678901234567890123456789012
>>%tempFile%
echo ;
>>%tempFile%
echo display info
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Synch with netmap
>>%tempFile%
set cdInstallDir= set spDir=%cdInstallDir%\Server\Secure+
pushd "%spDir%"
REM Main script This script has been prevented from running because it will alter
REM The configuration of Secure+. Before removing this warning and its
REM exit call, please modify the script so that it carries out only
REM desired modifications to the configuration of Secure+.
goto :EOF
call :initCustom
call :invokeCLI
call :terminateCustom
REM End of main script
```

(Continued)

```
goto :EOF
REM Functions
REM Custom initialization logic written by customer.
:initCustom
REM Customer adds custom initialization code here.
echo Init custom...
echo.
REM del /F "%spDir%\Nodes"
REM End of initCustom
goto :EOF
REM Invoke CLI to configure Secure+.
REM :invokeCLI
set tempFile=clicmds.txt
echo ;
>%tempFile%
echo ;
>>%tempFile%
echo sync netmap
>>%tempFile%
echo path=v4.5.00\KCHEN-TW-4500
>>%tempFile%
echo name=*
>>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Update localnode, enable protocol and add keys.
>>%tempFile%
echo ;
>>%tempFile%
echo update localnode
>>%tempFile%
echo override=n
>>%tempFile%
echo protocol=tls
>>%tempFile%
echo SSLTLSTRUSTEDROOTCERTFILE="%spDir%\Certificates\trusted.txt"
>>%tempFile%
echo SSLTLSCERTFILE="%spDir%\Certificates\keycert.txt"
>>%tempFile%
echo SSLTLSCERTPASSPHRASE=athena
>>%tempFile%
echo SSLTLENABLECIPHER=all >>%tempFile%
echo ;
>>%tempFile%
echo ;
>>%tempFile%
echo ; -- Display localnode
>>%tempFile%
echo ;
```

(Continued)

```
>>%tempFile%
echo display localnode
>>%tempFile%
echo ;
>>%tempFile%
echo validate parmfile
>>%tempFile%
echo ;
>>%tempFile%
echo quit;
>>%tempFile%
call "%spDir%\spcli.cmd" -e 8 -li y < %tempFile%
set RC=%ERRORLEVEL%
del %tempFile%
    REM End of invokeCLI
goto :EOF
REM Custom termination logic written by customer.
:terminateCustom
REM Customer adds custom termination code here.
REM For example, E-mail standard out log for review.
REM Send error messages to system monitoring facility.
echo.
echo Custom Terminating with errorlevel of %RC%
echo.
    REM End of terminateCustom
goto :EOF
popd
```

Chapter 14. Use LCU to Configure Encrypted Passwords

Configure Encrypted Passwords Using the LCU

The Sterling Connect:Direct Secure Plus CLI displays passwords in plain text. If you need to encrypt passwords for use with the Sterling Connect:Direct Secure Plus CLI, use the Local Connection Utility (LCU) to create an LCU file that contains non-encrypted information used to encrypt the password and the encrypted password, such as a keycert passphrase. You can then refer to this file when prompted for passwords.

LCU Files

The following example shows how to specify when an LCU file is used in place of a plain-text password:

```
C:\...\..\Connect Direct v4.6.00\Common Utilities>lcu -f C:\SomeDir\MyLCU.dat
*****
*          Connect:Direct Java Client Connection Utility          *
*                               Version 4.6.00 *
*-----*
* Copyright (c) 1983, 2011 *
* All Rights Reserved. *
*****
Node:
>JLYON-XP.4600
API Address: <Enter> = 'JLYON-XP'
>
API Port: <Enter> = '1363'
>
User Name:
>SomeValue
Password:
> Confirm Password:
>
Saving file: C:\SomeDir\MyLCU.dat
C:\...\..\Connect Direct v4.6.00\Server\Secure+>SPCLI
...
SPCLI> Create STSKeyPair
KeyPairFile=C:\SomeDir\StsKeyPairFile.dat
Passphrase=LCU:C:\SomeDir\MyLCU.dat;
SPCG670I rc=0 Create stskeypair command successful.
SPCLI> Update RemoteNode
Name=JLYON-XP.4600
StsAuthLocalKey=set
StsAuthKeyPairFile=C:\SomeDir\StsKeyPairFile.dat
StsAuthKeyPairFilePassphrase=LCU:C:\SomeDir\MyLCU.dat
SPCG470I rc=0 Update remote node "JLYON-XP.4600" command successful.
```

The use of the LCU syntax “LCU:” indicates that what follows is an LCU filename and not a passphrase. The pathname of the LCU file can be a relative path, a relative path to the bin directory, or a full path. If LCU:filename contains spaces, it must be enclosed in quotation marks: “LCU:filename”. The default name of the LCU file is cddef.bin. After the cddef.bin file is created, you can rename it as needed.

LCU files can be used to provide encrypted passwords for the following commands and parameters:

Command	Parameter
Update LocalNode	StsAuthKeyPairFilePassphrase StsSigKeyPairFilePassphrase SslTlsCertPassphrase
Create RemoteNode	StsAuthKeyPairFilePassphrase StsSigKeyPairFilePassphrase SslTlsCertPassphrase
Update RemoteNode	StsAuthKeyPairFilePassphrase StsSigKeyPairFilePassphrase SslTlsCertPassphrase
Create STSKeyPair	Passphrase
Update Client	SslTlsCertPassphrase
Update SEAServer	SslTlsCertPassphrase

Create an LCU File

About this task

To create an LCU file:

Procedure

1. Type the following command to run the LCU utility:

```
lcu.bat
```

2. As you are prompted, enter values for the following parameters:
 - Node
 - API Address
 - API Port
 - User Name
 - Password
 - Confirm Password
3. The cd.def file is created.

Notices

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

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

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

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Trademarks

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

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

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

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

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

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

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

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

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

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

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

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

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



Product Number:

Printed in USA