CONNECT:ENTERPRISE

CONNECT:Enterprise Command Line Client (Secure FTP)

Implementation Guide

STERLING COMMERCE

PN 00425

CONNECT:Enterprise™ Command Line Client (Secure FTP)

Implementation Guide

Version 1.1

CONNECT:Enterprise Command Line Client (Secure FTP) Implementation Guide
Version 1.1
First Edition

Contents

Chapter 4 Establishing Connections

Chapter 5 Exchanging Files Using Automation Scripts

Glossary

Index

Preface

CONNECT:Enterprise Command Line Client (Secure FTP) is the Sterling Commerce product for organizations that want to use the Secure Sockets Layer (SSL) protocol to transfer data securely from their client to any FTP server. The *CONNECT:Enterprise Command Line Client (Secure FTP) Implementation Guide* is intended to help you install, configure, and operate CONNECT:Enterprise Command Line Client (Secure FTP) in UNIX and Windows environments.

Task Overview

Refer to the following table for a list of tasks you can perform with CONNECT:Enterprise Command Line Client (Secure FTP) and where to find instructions for those tasks in this guide:

Task	Reference
Understanding CONNECT:Enterprise Command Line Client (Secure FTP) security features and operation	Chapter 1, *About CONNECT:Enterprise Command Line Client (Secure FTP)*
Installing CONNECT:Enterprise Command Line Client (Secure FTP) on a UNIX OS	Chapter 2, *Installing CONNECT:Enterprise Command Line Client (Secure FTP)*
Installing CONNECT:Enterprise Command Line Client (Secure FTP) on Windows 98, NT, and 2000 operating systems	Chapter 2, *Installing CONNECT:Enterprise Command Line Client (Secure FTP)*
Configuring security, navigating firewalls, and implementing Clear Channel Control	Chapter 3, *Configuring CONNECT:Enterprise Command Line Client (Secure FTP)*
Establishing secure and unsecure connections	Chapter 4, *Establishing Connections*
Automating file exchanges using scripts	Chapter 5, *Exchanging Files Using Automation Scripts*

Getting Support for Sterling Commerce Products

Sterling Commerce provides intuitive technical products and superior Help and documentation to enable you to work independently. However, if you have a technical question regarding a Sterling Commerce product, use the Sterling Commerce Customer Support Web site.

The Sterling Commerce Customer Support Web site at *www.sterlingcommerce.com* is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information. Refer to the Customer Support Reference Guide at *www.sterlingcommerce.com/customer/tech_support.html* for specific information on getting support for Sterling Commerce products.

Notational Conventions

The *CONNECT:Enterprise Command Line Client (Secure FTP) Implementation Guide* uses certain notational conventions to denote special circumstances. The following table describes the conventions used in this guide:

Convention	Description
bold	Bold letters represent screen names, key names, commands, and values that you must type as written.
italic	Italic letters are placeholders for information you must provide. For example, a reference to *filename* requires that you type the actual name of a file instead of the word in italic type. Italic letters also show emphasis and denote titles of publications.
underlined letters	Underlining indicates default values for fields. For example, Yes | No means that if a value is not specified, No is used.
`Monospaced characters` (characters of equal width)	Represent information for screens, commands, processes, and reports.
| (vertical bar)	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example, Yes | No means either Yes or No is valid, but not both.
[ ] (brackets)	Brackets indicate that information is optional. For example, -d[1|2|3[filename]] indicates that you can specify a debug number or debug number and file.
commas (,) parentheses ( )	Code all commas and parentheses as they appear.

Chapter 1

About CONNECT:Enterprise Command Line Client (Secure FTP)

CONNECT:Enterprise Command Line Client (Secure FTP) provides a means for manually and automatically exchanging files with the CONNECT:Enterprise server during secure and unsecure connections. CONNECT:Enterprise Command Line Client (Secure FTP) uses standard FTP commands for all connections. During secure connections, it also uses the Secure Sockets Layer (SSL) protocol to exchange files with the CONNECT:Enterprise server, which provides complete local and remote security over the Internet.

Security Essentials

CONNECT:Enterprise Command Line Client (Secure FTP) provides the following security essentials:

- Secrecy—The message cannot be read by anyone but the person it is addressed to because it is encrypted.
- Authentication—The person on the other end of a transmission is who he says he is.
- Data integrity—The message cannot be tampered with during transmission without you knowing it.
- Non-repudiation—The person who sent you the message cannot deny sending it.
- Data confidentiality—The message remains private during transmission.

CONNECT:Enterprise Command Line Client (Secure FTP) relies on cryptography, the science of keeping messages private, to achieve these security essentials. Through the use of public and private key pairs, certificates, and digital signatures, CONNECT:Enterprise Command Line Client (Secure FTP) ensures data remains private and unaffected during transmissions.

CONNECT:Enterprise Command Line Client (Secure FTP) Operation

In some situations within your business environment, access to information does not require strong security. Other situations require verification that a client is a legitimate and trusted partner. CONNECT:Enterprise Command Line Client (Secure FTP) provides the flexibility for you to establish the appropriate connection for your security need.

During secure FTP sessions, the server must always identify itself to the client. The identification process is called server authentication. In environments that demand strong security, it is necessary for both client and server to identify each other during FTP sessions. This second type of identification is known as client-server

authentication. CONNECT:Enterprise Command Line Client (Secure FTP) supports both types of authentication through the use of X.509 certificates.

In server-only authentication, the server sends a certificate to the client during the initial handshake process. The client compares the certificate to a list of trusted root certificates stored in a local directory. If the certificate sent by the server has been signed by a trusted source, the client establishes the connection.

With client-server authentication, both the client and the server send certificates to each other during the initial handshake process. Both client and server compare the certificates to a list of trusted root certificates stored in a local directory. If both certificates have been signed by a trusted source, a secure connection is established.

The certificates used during these sessions can be generated using Sterling Commerce Certificate Wizard. For more information about Certificate Wizard, see the Sterling Commerce Certificate Wizard Installation Card and Help.

Chapter 2

Installing CONNECT:Enterprise Command Line Client (Secure FTP)

This chapter describes installing CONNECT:Enterprise Command Line Client (Secure FTP) on a computer running the UNIX operating system and on a computer running the Windows 98, Windows NT, or Windows 2000 operating system.

Before You Begin

Before you install CONNECT:Enterprise Command Line Client (Secure FTP):

- Read *CONNECT:Enterprise Command Line Client (Secure FTP) Release Note*s and verify that your system meets the installation requirements.
- Verify that the CONNECT:Enterprise FTP server supports the SSL protocol.
- Verify that the appropriate Java components are installed.

Installing Command Line Client on a UNIX System

Automated installation scripts control installation of CONNECT:Enterprise Command Line Client (Secure FTP). The installation scripts use the following conventions:

Convention	Description
[y|n]	Specifies acceptable responses to prompts, where Y or y = yes and n or N = no.
[Y|n]	Identifies the default response with a capital letter.
Enter	Press to accept the default value.
Ctrl+C	Press to stop the executing script.

Note: Do not use colons (:) for values supplied at the prompts.

To install CONNECT:Enterprise Command Line Client (Secure FTP) and set up your Java environment, use the following steps:

1. Log on to the UNIX system with sufficient privileges, as defined by your company standards, to install the software. You may want to create an account specifically for this purpose.
2. Load the CONNECT:Enterprise Command Line Client (Secure FTP) CD-ROM in the CD-ROM drive and record its mount point.

   If you are using the HP-UX operating system, use the PFS utility so that the full file names on the CD-ROM are displayed. After you start PFS on the HP-UX system, use a command similar to the following to mount the CD-ROM:

   ```
   pfs_mount –t rrip /dev/dsk/c12d0 /cdrom
   ```

   For more information on starting and configuring the PFS utility, see pfs_mountd(1M) and pfsd(1M) in the MAN pages. To view an online man page topic, type the **man** command using the following example syntax:

   ```
   man pfs
   ```

3. From the root directory of your CD-ROM, type:

   ```
   $cesftp_inst
   ```

   The following screen is displayed:

   ```
                            Sterling Commerce, Inc., (TM)
                  CONNECT:Enterprise Command Line Client (Secure FTP)
   Installation Procedure:

   You are beginning the CONNECT:Enterprise Command Line Client (Secure FTP)
   Installation Procedure. You will be asked to specify a directory (called the
   destination directory) where the CONNECT:Enterprise Command Line Client (Secure
   FTP) files are stored.

   Sterling Commerce, Inc (TM) and CONNECT:Enterprise are trademarks of Sterling
   Commerce, Inc. in the U.S.A and other countries.

   UNIX is a trademark of UNIX Systems Laboratories, Inc.

   Press Enter to Continue:
   ```

4. Read the information on the screen, and then press **Enter** to begin the installation. The following prompt is displayed:

   ```
   Enter the path where the CD-ROM drive is mounted (e.g., /cdrom/ceftp1101):
   ```

5. Type the location of the CD-ROM root directory and press **Enter**. The following prompt is displayed:

   ```
   Enter the destination directory path where the CONNECT:Enterprise Command Line
   Client (Secure FTP) is to be installed [$HOME/clcftp]:
   ```

6. Type the full destination directory path for the CONNECT:Enterprise Command Line Client (Secure FTP) installation, and press **Enter**.

   Note: All files are installed in this directory, so you must have write permission for this directory.

   A prompt is displayed showing the directory you specified, as in the following example:

```
You have selected /ceftp_dir
for installation. Do you want to continue?: [Y|n]
```

7. Press **Enter** if the destination directory is correct. If it is incorrect, type **n** and repeat step 6 on page 2-3 to revise the path.

   ❖ If the destination directory does not exist, press **Enter** at the following prompt to create it:

```
Destination directory "/ceftp_dir" does not exist.
Do you want to create it?:[Y|n]
```

   ❖ If the destination directory exists, press **Enter** at the following prompt to overwrite it:

```
Files in /ceftp_dir could be overwritten by the installation. Do you want to
continue?:[Y|n]
```

   Note: You can type **n** at either of these prompts and repeat step 6 on page 2-3 to create a different destination directory.

   The following prompt is displayed:

```
Please specify whether you are a U.S.(domestic) or International user. Type
"US" if you are located within the U.S. or "INT" if you are located outside of
the U.S.:
```

8. Type your license type, **US** or **INT**. Depending on your response, the domestic or international license agreement is displayed.

```
License agreement:
….
….

Do you accept the above license agreement?:[y|n]
```

   To view the entire agreement, press the **Space Bar** to scroll or press **Enter** to view line by line.

9. Type **y** to accept the agreement or **n** to decline it. There is no default value. You must accept the license agreement to continue the installation.

After you accept the license agreement, the following screen is displayed:

```
6000 kbytes of disk space are required to install
CONNECT:Enterprise Command Line Client (Secure FTP).

Please wait while the installation determines if there is sufficient disk space to
complete this operation.

Sufficient free disk space for install is available.
6000 kbytes needed, 2005885 kbytes available
```

Note: If you are prompted that the disk space is insufficient, exit the installation. Do not continue the installation with insufficient space because CONNECT:Enterprise Command Line Client (Secure FTP) will not be installed properly.

10. Press **Enter** to continue. The following screen is displayed:

```
Installing CONNECT:Enterprise Command Line Client (Secure FTP)

...Extracting CONNECT:Enterprise Command Line Client (Secure FTP) files from the
media. Actual time taken varies depending on the system configuration.

cesftp.jar
README
trusted.txt
ceftpver
ceuftpcust
sslj.jar
jsafe.jar
PwdTerm.so

...Done

Verify CONNECT:Enterprise Command Line Client (Secure FTP) files are extracted.

...Done.
```

After CONNECT:Enterprise Command Line Client (Secure FTP) is installed, you are prompted to set up your Java environment:

```
Do you want to set up the Java environment for the CONNECT:Enterprise Command Line
Client (Secure FTP)? [Y|n]
```

11. Press **Enter** to continue.

Note: Although you can configure the Java environment later by typing **ceuftpcust** at the installation directory prompt, you must set up your Java environment before you can operate CONNECT:Enterprise Command Line Client (Secure FTP).

The following screen is displayed:

```
Sterling Commerce, Inc.,(TM)
CONNECT:Enterprise Command Line Client (Secure FTP) Customization.

To abort the process, enter Control-C.

"Please press ENTER to continue..."
```

12. Press **Enter** to continue. The system searches for supported Java programs on your computer, assigns a number to all versions, and lists the full path for each version, as in the following example:

```
**********************************************
Select the appropriate Java interpreter program.

Contact your System Administrator if you are unsure
about the appropriate selection.
**********************************************
0 = To enter the absolute path of the Java program (Java):
1 = /usr/java1.2/jre/bin/java
2 = /usr/java1.2/bin/java
3 = /usr/java1.2.beta4/jre/bin/java
4 = /usr/java1.2.beta4/bin/java
Select [0-4]:
```

Note: If the system does not find a supported Java program, you are prompted to enter the absolute path of the Java program. Contact your system administrator if you are unsure about the appropriate entry.

13. Select a Java program from the list, type its corresponding number, and press **Enter.** To use a different Java program, type **0** to display a prompt where you can type the absolute (full) path of another Java program (for example, /usr/java_dev/bin/java).

    If you type the absolute path for an unsupported Java program, you are prompted to try again. Repeat this step until you type the path for a supported Java program. Refer to *CONNECT:Enterprise Command Line Client (Secure FTP) Release Notes* for more information about supported Java programs.

    After you select a supported Java program, the installation script completes the Java setup, creates the script file, and displays the following prompt:

```
The script file /ceftp_dir/ceftp has been created!

To bring up the CONNECT:Enterprise Command Line Client, type ceftp from the
installation destination directory.
```

Note: If the class file (rt.jar or classes.zip) is missing from the Java program directory, the system searches your computer for class files, assigns a number to each one, and lists its full path. You must select a class file, type its corresponding number, and press **Enter**. If the system does not find a class file, you are prompted to enter the absolute path of the class file. Contact your system administrator if you are unsure about the appropriate entry.

CONNECT:Enterprise Command Line Client (Secure FTP) is now fully operational.

Setting Environment Variables

To run CONNECT:Enterprise Command Line Client (Secure FTP) from a different directory, update your PATH environment variable, as illustrated the following examples:

```
(csh)

setenv PATH "ceftp_dir:$PATH"
```

```
(sh or ksh)

PATH="ceftp_dir:$PATH";export PATH
```

Installing Command Line Client on a Windows System

Install CONNECT:Enterprise Command Line Client (Secure FTP) in the Windows 98, Windows NT, or Windows 2000 environment using the following steps:

1. Exit all Windows programs that are running.
2. Insert the CONNECT:Enterprise Command Line Client (Secure FTP) CD-ROM into the CD-ROM drive.

   If the Autorun option is enabled for the CD-ROM drive, the CONNECT:Enterprise Command Line Client (Secure FTP) installation setup detects the type of Windows operating system installed on your computer and automatically starts.

   If the Autorun option is disabled on your computer:

   a. Click the **Start** button, and click **Run**.
   b. In the **Run** dialog box, click the **Browse** button.
   c. In the **Browse** dialog box, select the drive mapped to your CD-ROM drive from the **Look in:** field drop-down box.
   d. Select the \Win32\CONNECT Enterprise Command Line Client (Secure FTP) folder.
   e. Double-click **CommandLineClient(SecureFTP).exe**. The program returns to the **Run** dialog box.
   f. Click **OK**.
3. At the **Welcome** screen, click **Next** to begin the installation.
4. Specify whether you are a **US** or **International** user by clicking the option button next to the correct selection. Click **Next**.
5. In the **Software License Agreement** dialog box, read the license agreement and click the box that specifies **I have read the license and agree**. Click **Accept**.
6. In the **Choose Destination Location** dialog box, click **Next** to accept the default installation folder.
7. In the **Select Program Folder** dialog box, select **Enterprise Command Line Client (Secure FTP)** from the scroll list and click **Next**.
8. After copying the files into the program folder, the installation program displays the **Setup Complete** dialog box. Click **Finish** to complete the installation.

Chapter 3

Configuring CONNECT:Enterprise Command Line Client (Secure FTP)

CONNECT:Enterprise Command Line Client (Secure FTP) supports secure and unsecure FTP connections. Before you can use CONNECT:Enterprise Command Line Client (Secure FTP) to establish secure connections, you must configure the system to support server or client-server authentication. You achieve this authentication using certificates.

Configuring the Security Environment

The use of certificates is an important component of CONNECT:Enterprise Command Line Client (Secure FTP) functions.Certificates are issued by a trusted, well-known entity called a certificate authority (CA). A certificate authority is responsible for verifying and processing certificate requests, and issuing and managing certificates. You should choose a certificate authority that your trading partners trust. You must meet the requirements of the certificate authority you choose.

Note: A fatal error is returned and the CONNECT:Enterprise Command Line Client stops if the server public certificate is not signed by a trusted CA, or if the trusted root file is corrupted or unreadable.

Certificates typically contain:

- Distinguished name and public key of the server or client
- Distinguished name and digital signature of the certificate authority
- Period of validity (certificates expire and must be renewed)
- Administrative and extended information

CONNECT:Enterprise Command Line Client (Secure FTP) uses certificates in two files that are integral to its operation:

- Trusted root certificate file—enables the server to identify itself and be identified by the client during FTP sessions
- Key certificate file—enables the client to identify itself through the use of an encrypted message and be identified by the server during secure FTP sessions

Installing Trusted Root Certificate Files

The trusted root certificate file, trusted.txt, is included with CONNECT:Enterprise Command Line Client (Secure FTP) and located in your installation directory. As installed, it includes trusted root certificates from two reputable public certificate authorities.

If the server uses certificates from other public certificate authorities or utilities, you must install the corresponding trusted root certificates on the client workstation by manually pasting them into the trusted root certificate file (trusted.txt) after the last END CERTIFICATE line. These other certificates must be compatible with the required format (ASCII, base64 encoded text).

To install a trusted root certificate file, complete the following steps:

1. After you obtain the certificate for your certificate authority, select and copy the file contents.
2. Start a text editor.
3. Open the trusted.txt file located in the CONNECT:Enterprise Command Line Client (Secure FTP) installation directory.
4. When the file opens, scroll to the bottom of the page and locate the END CERTIFICATE line.
5. Place your cursor on the following line and press **Enter** to add a blank line to the file.
6. Paste the contents from step 1.
7. Save the file.
8. Make a backup copy of the trusted.txt file.

WARNING: Automated checking against certificate revocation lists (CRLs) is not implemented in CONNECT:Enterprise Command Line Client (Secure FTP). If the server certificate is compromised, the administrator of the FTP server must notify all trading partners.

Example: Trusted Root Certificate File (trusted.txt)

```
RSA Commercial CA / Verisign - exp.Jan 7, 2010
-----BEGIN CERTIFICATE-----
MIICNDCCAaECEAKtZn5ORf5eV288mBle3cAwDQYJKoZIhvcNAQECBQAwXzELMAkG
A1UEBhMCVVMxIDAeBgNVBAoTF1JTQSBEYXRhIFNlY3VyaXR5LCBJbmMuMS4wLAYD
….
….
-----END CERTIFICATE-----

Thawte Server CA, 1996.07.31 - 2020.12.31
-----BEGIN CERTIFICATE-----
MIIDEzCCAnygAwIBAgIBATANBgkqhkiG9w0BAQQFADCBxDELMAkGA1UEBhMCWkEx
FTATBgNVBAgTDFdlc3Rlcm4gQ2FwZTESMBAGA1UEBxMJQ2FwZSBUb3duMR0wGwYD
….
….
-----END CERTIFICATE-----
```

Creating Key Certificate Files

A key certificate file is necessary when you want to establish a secure FTP connection using client-server authentication. The key certificate file contains a private key and an X.509 certificate, which is provided by a certificate authority. You can use Certificate Wizard to create key certificate files. See the Sterling Commerce Certificate Wizard Installation Card for instructions on installing Certificate Wizard.

With Sterling Commerce Certificate Wizard, you can:

- Generate the private key necessary for the key certificate file
- Generate a Certificate Signing Request (CSR) to request the X.509 certificate
- Submit the CSR to the certificate authority

After the certificate authority validates the information in the CSR, you receive a certificate that you can use to create a key certificate file. To create the key certificate file, you must manually attach the X.509 certificate to the private key. For more information on using Certificate Wizard to perform these tasks, see the Sterling Commerce Certificate Wizard Help.

Note: A fatal error is returned and the CONNECT:Enterprise Command Line Client stops if the server public certificate is not signed by a trusted certificate authority, or if the key certificate file is corrupted or unreadable.

Navigating Firewalls

CONNECT:Enterprise Command Line Client (Secure FTP) uses two features that enable you to control firewall navigation when you connect from the client to the server.

- Setting port range limits
- Implementing the Clear Control Channel (CCC) feature

Setting Port Range Limits

Setting port range limits enables you to restrict the TCP/IP ports used for FTP transactions between CONNECT:Enterprise Command Line Client and CONNECT:Enterprise for UNIX, providing a more secure environment. You control the order in which port numbers are assigned by the system and specify which port ranges are available for transactions. Assign a specific TCP/IP source port number or a range of port numbers with a particular TCP/IP address (or addresses) for incoming CONNECT:Enterprise sessions.

Note: Because these ports must also be available at the server end of the connection, you need to coordinate with system personnel at your trading partner site. The server must be running CONNECT:Enterprise for UNIX 1.2.02 or later to use this feature.

Specify the TCP/IP ports in a port-range list using the following syntax:

```
[retries/retrywait/]nnnnn-nnnnn*
```

Parameter	Definition	Valid Values
retries	Optional. The number of times the system will attempt to reestablish a connection if the original connection fails.	The numeric values 0 to 99. The default is 0
retrywait	Optional. The number of seconds between each attempt to establish a connection.	The numeric values 0 to 180 Default is 0
range	A range of port numbers using the format nnnnn-nnnnn. Separate multiple port ranges with commas.	A numeric value where nnnnn-nnnnn represents the beginning and end of each range.

Sample Command Lines

Port ranges can be specified using the –R command line parameter or in a script file called by the –a command line parameter. See Chapter 4, *Establishing Connections*, for command line parameter definitions and usage.

The following sample command line specifies two port ranges, the first from forty to fifty thousand inclusive, and the second from fifty-five to sixty thousand inclusive. If the original connection attempt fails, there will be one retry with a delay of ninety seconds between connection attempts:

```
-R 1/90/40000-50000,55000-60000
```

The same format applies when specifying port ranges in a script file. The following sample command line illustrates the port_range command in a script:

```
port_range 1/90/40000-50000,55000-60000
```

Implementing the Clear Control Channel (CCC) Feature

Using the CCC policy, you can request that the FTP command socket revert to clear text after user authentication has been performed. The CCC policy setting is only applicable to Secure FTP, and must be enabled at both the client end and server end of the connection.

Note: The server must be running CONNECT:Enterprise for UNIX 1.2.02 or later to use this feature.

The following table provides the definitions of the valid CCC policy values.

Value	Description
Required	The client transmits the CCC command to the CONNECT:Enterprise server. - If the server returns a positive response, all subsequent transmissions on the control socket are in clear text. - If the server returns a negative response, the command line client stops.

Value	Description
Optional	The client transmits the CCC command to the CONNECT:Enterprise server. - If the server returns a positive response, all subsequent transmissions on the control socket are in clear text. - If the server returns a negative response, the connection remains open but all subsequent transmissions on the control socket remain encrypted.
Disallowed (default)	The client does not send the CCC command to the CONNECT:Enterprise server. *This is consistent with not specifying the parameter.*

Setting the CCC Policy

The CCC policy can be set in three ways using the command line parameters. See Chapter 4, *Establishing Connections*, for command line parameter definitions and usage:

- From the command line using the –C option, type **–C r|o|d**. Only the first letter, of each argument to the –C option, is necessary.
- In the security configuration file called by the –a command line parameter, use the keyword cccpolicy=, for example, cccpolicy=required|optional|disallowed. The full keyword must be used.
- With a locsite command in a script file called by the –a command line parameter, type the following: locsite cccpolicy=required|optional|disallowed. The full keyword must be used.

Chapter 4

Establishing Connections

CONNECT:Enterprise Command Line Client (Secure FTP) enables you to establish the type of connection appropriate for your business and your trading partners. This chapter describes how to establish secure and unsecure connections and the commands that you can use during those sessions.

Before You Begin

Before you start CONNECT:Enterprise Command Line Client (Secure FTP), ensure that the CONNECT:Enterprise server has SSL FTP enabled and gather the following information from your host site administrator:

- Mailbox ID
- Mailbox password
- IP address or host name of the CONNECT:Enterprise server and the FTP listening port
- Authentication level (server only or client-server)
- Encryption strength
- Network path and firewall information

Note: Secure FTP works only with firewalls that do not inspect data that passes through them (such as normal packet filtering, Static Network Address Translation, or Static NAT, and SOCKS). Proxy account firewalls do not work with secure FTP.

You use this information to access the CONNECT:Enterprise server.

Establishing Secure Connections

Before you can establish a connection that requires client-server authentication, you must define the **keycert**, **trusted**, and **strength** parameters. The **keycert** parameter sets the key certificate file name and location (path); the **trusted** parameter sets the trusted root certificate file name and location (path); and the **strength** parameter sets the encryption strength used during the SSL session.

Note: The **keycert** parameter must be set for sessions that require client-server authentication. If client-server authentication is not necessary, only the **trusted** and **strength** parameters are required.

Three options are available for defining security parameters:

- Configuration file—By default, if the configuration file exists, the system uses the security parameter settings in that file.
- Command line—If the configuration file does not exist, the system refers to the command line for security parameter settings. Defining security parameters on the command line also overrides security settings in the configuration file.
- **locsite** subcommand—If the security parameters are not defined, you can define them using the **locsite** subcommand. You can also use the **locsite** subcommand to override all other parameter settings in the configuration file or on the command line, and to specify a different configuration file. For more information about using the **locsite** subcommand to define security parameter values, see the *Overriding Configuration File Security Parameters to Establish a Connection* on page 4-3.

Defining Default Security Parameters in the Configuration File

The CONNECT:Enterprise Command Line Client (Secure FTP) installation creates two configuration files, sample_secureftp.cfg and secureftp.cfg, in the installation directory. Both files include the SSL parameters that CONNECT:Enterprise Command Line Client (Secure FTP) accesses to establish secure FTP connections, but only the secureftp.cfg file is used for this purpose. The sample_secureftp.cfg is included only as a backup template.

Define the following SSL parameters in the secureftp.cfg file:

Parameter	Description
keycert=*keycert filename*†	Specifies the location (path) and file name of the key certificate file.
strength=strong|weak|all	Specifies the Encryption strength used during the SSL session.
trusted=*trusted filename*	Specifies the location (path) and file name of the trusted root certificate.
cccpolicy=required|optional|disallowed	Specifies whether a clear control channel is used.

† Keycert value is only necessary if client-server authentication is required.

The following example illustrates the contents of a configuration file:

```
trusted=trusted.txt
keycert=keycert.txt
strength=strong
cccpolicy=disallowed
```

To define your SSL parameters in the secureftp.cfg file, complete the following steps:

1. Record the location (path) and name of the key certificate file.
2. Start a text editor.
3. Open the secureftp.cfg file located in the CONNECT:Enterprise Command Line Client (Secure FTP) installation directory.
4. Replace the **keycert=** entry with the key certificate file path and file name from step 1.
5. Make any other changes for the **trusted** and **strength** parameters.
6. Save the file.
7. Make a backup copy of the secureftp.cfg file.

If you do not define the security parameters, CONNECT:Enterprise Command Line Client (Secure FTP) uses the default entries in the secureftp.cfg file.

Establishing a Connection Using the Configuration File Security Parameters

To establish a secure connection using the values specified for the security parameters in the configuration file, complete the following steps:

1. At the command line prompt, type **ceftp** and the *host name* and *port number* of the CONNECT:Enterprise server to which you want to connect, as in the following example, and press **Enter**:

```
ceftp CEServer 10021
```

2. Type your *userid* and *Password* at the appropriate prompts, and press **Enter**.

WARNING: HP-UX 10.20 does not hide the *Password* when typed. This is an issue between HP-UX 10.20 and Java 1.1.8 that cannot be resolved with CONNECT:Enterprise Command Line Client (Secure FTP).

When the secure connection is established, the following prompt is displayed:

```
ceftp-s>
```

Overriding Configuration File Security Parameters to Establish a Connection

You can define and override the **keycert**, **trusted**, and **strength** parameters specified in the configuration file from the command line and by using the **locsite** subcommand.

To define the security settings from the command line, type the following information at the command line prompt:

- **–c** *keycert filename*—key certificate file for client-server authentication (optional)
- **–t** *trusted filename*—trusted root certificate file for server authentication
- **–e s**trong|**w**eak|**a**ll—encryption strength

To define the security parameters and establish a secure connection from the command line, complete the following steps:

1. At the command line prompt, type **ceftp** and the *host name* and *port number* of the CONNECT:Enterprise server to which you want to connect, and the **keycert**, **trusted**, and **strength** entries, similar to the following example:

```
$ceftp CEServer 10021 -c keycert.txt -t trusted.txt -e s
```

2. Press **Enter**. The following prompt is displayed if a key certificate file is present:

```
Certificate Passphrase:
```

3. At the Certificate Passphrase prompt, type the passphrase you specified for the key certificate file in Sterling Commerce Certificate Wizard.

WARNING: HP-UX 10.20 does not hide the Certificate Passphrase when typed. This is an issue between HP-UX 10.20 and Java 1.1.8 that cannot be resolved with CONNECT:Enterprise Command Line Client (Secure FTP).

When the secure connection is established, the following prompt is displayed:

```
ceftp-s>
```

You can use the **locsite** subcommand with the **keycert**, **trusted**, **strength,** and, **cccpolicy** parameters to define the security settings. Type the following information after the **locsite** subcommand:

- **keycert**—location and name of the key certificate file for client-server authentication (optional)
- **trusted**—location and name of the trusted root certificate file for server authentication
- **strength**—encryption strength
- **cccpolicy**—specifies whether clear control channel is used

To define the security parameters and establish a secure connection using the **locsite** subcommand, complete the following steps:

1. Type the **keycert**, **trusted**, **strength**, and **cccpolicy** entries with the **locsite** subcommand, similar to the following example, and press **Enter** after each entry:

```
>locsite keycert=/ceftp_dir/keycert.txt
>locsite trusted=/ceftp_dir/trusted.txt
>locsite strength=strong
>locsite cccpolicy=disallowed
```

The following prompt is displayed if a key certificate file is present:

```
Certificate Passphrase:
```

2. Type the Certificate Passphrase for the key certificate file that you specified in Sterling Commerce Certificate Wizard. When the secure connection is established, the following prompt is displayed:

```
ceftp-s>
```

See *Supported Subcommands* on page 4-7 for other ways to use the **locsite** command.

Establishing Unsecure Connections

To establish a connection that does not use SSL security, you can use the unsecure command line parameter (**–u**) or the **locsite unsecure** subcommand. Using the unsecure command line parameter (**–u**) makes the entire session unsecure. Using the **locsite unsecure** subcommand makes one connection unsecure. For more information about the **locsite** subcommand, see the **locsite** entry in *Supported Subcommands* on page 4-7.

Establish an unsecure connection in one of the following ways:

- At the command line prompt, type **ceftp**, the *host name* and *port number* of the CONNECT:Enterprise server to which you want to establish a connection, and the **–u** parameter, as shown in the following example, and press **Enter**:

```
$ceftp host_name port_number –u
```

  The following prompt is displayed:

```
All connections will be unsecure (for every connection).
ceftp>
```

- At the command line prompt, type **ceftp** to start CONNECT:Enterprise Command Line Client (Secure FTP). At the ceftp prompt, type the **locsite unsecure** subcommand, as in the following example:

```
ceftp>locsite unsecure
```

  The following prompt is displayed:

```
An unsecure connection will be attempted.
ceftp>
```

Command Line Parameters

After you establish a connection, you can use the supported command line parameters and supported subcommands. In addition, refer to the *CONNECT:Enterprise for UNIX Remote User's Guide* and the *CONNECT:Enterprise for OS/390 User's Guide* for a detailed description of and examples for sending standard FTP syntax commands.

The following command line parameters are supported by CONNECT:Enterprise Command Line Client (Secure FTP). File names with spaces must be enclosed with double quotes (" ").

Command Line Parameter	Description
–a *automation script filename*	Specifies the location and file name of the automation script file; see Chapter 5, *Exchanging Files Using Automation Scripts*, for more information
–c *key certificate filename*	Specifies the location and file name of the key certificate file

Command Line Parameter	Description
–d [level [*filename*|*ceftp.trc*]]	Specifies the level of debug and/or debug file; overwritten at each CONNECT:Enterprise Command Line Client (Secure FTP) startup: 0 = Disable debug output 1 = Connection status, send/receiving a file, security channel requested 2= FTP commands, SSL FTP responses, and level 1 logs 3 = IPC connections (ipaddr, port #), accepts, rejects, authentication status (pass or failed) and level 2 logs Note: If only the debug level is specified, the default file name (ceftp.trc) is used and the file is placed in the product runtime directory.
–e encryption_strength	Specifies the encryption strength (cipher strength) to use with the SSL connection: s = **strong** uses strongest encryption level possible w = **weak** uses weak encryption a = **all** uses available encryption algorithms. The following ciphers are supported: Strong RSA_WITH_RC4_128_SHA RSA_WITH_RC4_128_MD5 RSA_WITH_3DES_EDE_CBC_SHA RSA_WITH_DES_CBC_SHA Weak RSA_EXPORT_WITH_DES_40_CBC_SHA RSA_EXPORT_WITH_RC4_40_MD5 RSA_WITH_NULL_MD5 RSA_WITH_MULL_SHA
–h	Returns the command line syntax
host_name	Specifies the name of the system running CONNECT:Enterprise server; you can enter the IP address of the host instead of the host name.
–i	Turns off interactive prompting during multiple document transfers (prompting on by default)
port_number	Specifies the CONNECT:Enterprise Server FTP port listener number; see Chapter 2 of the *CONNECT:Enterprise for UNIX System Administrator's Guide* for a description of the CPD file, which defines the FTP port listener number
–R	Enables you to control firewall navigation when connecting from client to server by specifying up to five ports or ranges of ports used to establish connections.
–r	Returns the product name, release, and build.
–s *configuration filename*	Specifies the location and file name of the client configuration file, which is a user-defined configuration file
–t *trusted root certificate filename*	Specifies the location and file name of the trusted root certificate file
–u	Specifies to CONNECT:Enterprise Command Line Client (Secure FTP) to ignore all security parameters and establish an unsecure connection; generates a message saying that the connection is not secure for every connection
–v	Turns off verbose (verbose on by default)
–x	Turns on result code exiting for the entire instance of the client

Supported Subcommands

The following standard FTP syntax subcommands are supported by CONNECT:Enterprise Command Line Client (Secure FTP). The subcommands can be entered at the ceftp> prompt. File names with spaces must be enclosed with double quotes (" ").

Note: CONNECT:Enterprise Command Line Client (Secure FTP) supports only the subcommands listed in the following table.

Subcommand	Description
ascii|asc|a	Sets ASCII transfer type
binary|bin|b	Sets binary transfer type
cd	Changes the working mailbox ID
close	Closes all client-to-server connections
delete	Flags a document of data as deleted
dir	Requests a formatted listing of documents from the host site
get	Requests a formatted document of data from the host site
help|? [command]	Returns help information; type **help *command*** at the command prompt to receive help information for a particular command
lcd	Changes the local working directory
locsite	Sets the security configuration parameters locally; overrides all other parameters; valid parameters are: keycert — specifies the location and file name of the key certificate file trusted — specifies the location and file name of the trusted root certificate file strength — specifies what encryption strength should be used with the SSL connection; options are strong, weak, all securecfg — specifies the location and file name of the client security configuration file unsecure — specifies an unsecure connection; valid for only one connection; you must type another **locsite unsecure** subcommand (before the open subcommand) for another unsecure connection cccpolicy — specifies whether a clear control channel can be used; default is disallowed; this feature must be enabled on both ends of the connection **Note:** By typing **locsite** at the CONNECT:Enterprise Command Line Client (Secure FTP) command line prompt, you can validate current security settings before you make a secure connection.
mdelete (mdel)	Flags multiple documents of data as deleted
mget	Receives multiple documents of data from the host site
mput	Sends multiple documents of data from the host site
open	Notifies the remote FTP server with a PORT command

Subcommand	Description
passive	Notifies the server of a passive mode connection
prompt	Forces interactive prompting on multiple commands
put	Sends a document of data to the host site
pwd	Prints the working mailbox ID
quit|bye	Closes all connections and exits the client
site	Commands that are used to give specific configuration options to the host site and are only good for the ceftp session
user	Sends new user information to the host site
verbose	Toggles verbose mode (default on)
!command [parameters]	Sends the command to the operating system

Additional Examples for Establishing Connections

The following examples show different entries you can make in CONNECT:Enterprise Command Line Client (Secure FTP) to establish secure or unsecure connections.

Example 1—Secure connection with client-server authentication using a default configuration file:

```
#ceftp CEserver 10021
Certificate Passphrase:
Name (CEserver:myid):myid
Password:
ceftp-s>
```

-or-

```
#ceftp
Certificate Passphrase:
ceftp>open CEserver 10021
Name (CEserver:myid):myid
Password:
ceftp-s>
```

Example 2—Secure connection using server-only authentication by setting the configuration file at the prompt (if the key certificate file is not defined in the configuration file):

```
#ceftp -s /user/jeff/myconfig
ceftp>
```

Example 3—Secure connection using client-server authentication by setting the configuration file through the **locsite** subcommand:

```
#ceftp
ceftp> locsite securecfg=/home/jeff/myconfig
Certificate Passphrase:
ceftp>
```

Example 4—Secure connection using client-server authentication by entering security settings at the command line prompt:

```
#ceftp CEserver 10021 -c /home/sam/keycert.txt -t /opt/ceftp/trusted.txt -e s
Certificate Passphrase:
Name (CEserver:myid):myid
Password:
ceftp-s>
```

Example 5—Secure connection using server-only authentication that specifies security settings through the **locsite** subcommand, which overrides any settings specified at the command line prompt or in the configuration file:

```
#ceftp CEserver 10021 -e w
Name (CEserver:myid):myid
Password
ceftp-s> get newsfile
Transferred 3038 bytes in 0.0 seconds (233.0 bytes/sec)
226 Transfer complete (Batch Number = 25).
200 PORT command successful.
ceftp-s> close
ceftp> locsite strength=strong
ceftp>open RealSecureServer 20021
Name (CEserver:myid):myid
Password
ceftp-s>
```

Example 6—Current security settings, which are viewed by typing the **locsite** subcommand on the command line at any time during the connection:

```
#ceftp locsite
Local Site Status:
Client Certificate File="null"
Server Trusted Root File="/home/temp2/kmoor1/build08/hp_ceftp/trusted.txt"
Encryption Strength="strong"
Security flag="true"
Configuration File="/home/temp2/kmoor1/build08/hp_ceftp/secureftp.cfg"
```

Example 7—All connections are unsecure:

```
#ceftp -u
All connections will be unsecure (for every connection).
ceftp> open myhost myport
```

Example 8—A single connection is unsecure:

```
#ceftp
ceftp> locsite unsecure
An unsecure connection will be attempted.
```

Example 9—Site commands used with CONNECT:Enterprise Repository Data Exchange for OS/390:

```
ceftp>site dir_filter=DIT KEEP
200 The value of the DIR_FILTER is DIT
ceftp-s>dir
QATEMP   #0000054 CT=000019968 BID=junk                    1546-00132  C R  MU
QATEMP   #0000096 CT=000285966 BID=client.bmp              1304-00140  C R  M
QATEMP   #0000097 CT=000019968 BID=junk.doc                1307-00140  C R  M
QATEMP   #0000098 CT=000019968 BID=junk.doc                1307-00140  C R  M
QATEMP   #0000099 CT=000019968 BID=junk.doc                1307-00140  C R  M
ceftp-s>quote stat
211 211-CONNECT:Enterprise RDX at 17:03:17 on 2000.189 host time.
211-Session started at 16:57:06 on 2000/189 host time.
211-User: QATEMP     Current working Mailbox ID: QATEMP
211-TYPE: A       MODE: S           STRUcture: F
211-Local SITE option values:
211- Allocation type=NONE      BCHSEP=NONE  BLKSIZE=0
211- DIR_FILTER=DIT                     DIRECTORY=0         DIRFORM=MBOX_CLIENT
211- EO=NO   FTIME=1980001:0000   LRECL=0      LS_FILTER=!M
211- MULTXMIT=YES  ONEBATCH=YES  ORIGIN=             PRIMARY=0
211- RECFM=        REMOTE_FILENAME_LENGTH=LONG     SECONDARY=0
211- TO=NO   TTIME=               XMIT=YES
211-           0 Kbytes received for         0 batches during this session
          78 Kbytes sent from            4 batches during this session
ceftp>site blksize=32760
200 SITE command was accepted.
ceftp>site onebatch=no
200 SITE command was accepted.
ceftp>site xmit=no
200 SITE command was accepted.
```

For additional instructions on using site commands, see the *CONNECT:Enterprise Repository Data Exchange for OS/390 Remote User's Guide*.

Example 10—FTP $$ Commands used with CONNECT:Enterprise for UNIX:

```
ceftp> dir "$$ ID=ACCTPAY BID='invoices' PASSWORD=letmein"

ceftp-s>put sales.dat "$$ ID=ACCTG BID='sales' XMIT=Y"

ceftp-s> get "$$ ID=MyMBX BID='my payroll' CONV=A" mytax.file
```

For additional instructions on using the FTP $$ commands, see the *CONNECT:Enterprise for UNIX Remote User's Guide*.

Chapter 5

Exchanging Files Using Automation Scripts

CONNECT:Enterprise Command Line Client (Secure FTP) provides automated scripting capabilities for file exchanges during secure and unsecure FTP connections. This scripting capability eliminates the need for you to run CONNECT:Enterprise Command Line Client (Secure FTP) manually. This feature works on any platform that supports Java.

Windows and UNIX Automation Scripts

If you want to use the automated scripting capabilities of CONNECT:Enterprise Command Line Client (Secure FTP), you must create an automated script file that contains subcommands.

The following is an example of a secure automation script file called auto_sc_file:

```
mypassphrase
open myhost myportnum
myid
mypassword
get file1
quit
```

To run the script, type:

```
$ceftp –a auto_sc_file
```

Note: If the key certificate file is defined in the CONNECT:Enterprise Command Line Client (Secure FTP) configuration files, the first line that appears when the script runs is the response to the passphrase prompt. All other lines are subcommands that run automatically. The subcommands are standard FTP syntax commands supported by CONNECT:Enterprise Command Line Client (Secure FTP).

The following is an example of an unsecure automation script file called uauto_sc_file:

```
open myhost myportnum
myid
mypassword
get file1
quit
```

To run the script, type:

```
$ceftp –a uauto_sc_file –u
```

Note: The subcommands in the uauto_sc_file are standard FTP syntax commands supported by CONNECT:Enterprise Command Line Client (Secure FTP).

Return Codes

The return code from a CONNECT:Enterprise Command Line Client (Secure FTP) invocation can help you determine whether to restart CONNECT:Enterprise Command Line Client (Secure FTP) to resend data or to send a subcommand. The only way to check the return code is within a script. The following table lists possible return codes for CONNECT:Enterprise Command Line Client (Secure FTP):

Return Code	Category
0	FTP commands were OK
1	Session establishment failure occurred
2	Authentication or login failure occurred
3	Client subcommand (none-copy) failure occurred
4	Subcommand put(STOR) failure occurred
5	Subcommand get(RETR) failure occurred
6	SSL configuration file parameter failure occurred
7	Command line parameter failure occurred
8	**Locsite** command parameter failure occurred
9	Reserved
10	Catastrophic failure occurred

You can perform the return code checks in two ways:

- Using the **–x** command line parameter
- Typing the @ symbol next to the subcommand for which you would like the return code checked

The **–x** command line parameter checks return codes for all commands. The @ symbol only checks return codes for the subcommand (s) with which it is associated. You can use both types of return code checking on UNIX or Windows platforms.

UNIX Scripting

You can check return codes on a UNIX platform with automation script files. The following example shows a script that invokes CONNECT:Enterprise Command Line Client (Secure FTP) with the automation script file.

```
#!/bin/sh
#
retc=0#Set user return value to 0.
#Invoke the Enterprise Command Line Client with the Return Code Checking On (—x).
#
ceftp —a auto_file.txt —x
retc=`echo $?`

#
# Check the Return Code
#
if [ $retc -eq 4 ]; then # PUT Command Failed
echo "The Account Log did not transfer"
elif [ $retc -ne 0 ]; then
echo "Enterprise Command Line Client experienced a failure."
else
echo "The Account Log was sent successfully"
fi
exit $retc
```

The preceding example references the automation script file auto_file.txt, with the **–x** command line parameter to initiate return code checking. The auto_file.txt file has the following contents:

```
mypassphrase
open myhost myportnum
mymboxid
mypassword
put /sql/repository/accounts.long "$$ID=bankone BID='Weekly accounts log'"
quit
```

To initiate CONNECT:Enterprise Command Line Client (Secure FTP) without return code checking, place the @ symbol next to the **put** subcommand in the auto_file.txt file and remove the **–x** command line parameter from the ceftp –a auto_file.txt line of the script.

Windows Scripting

For return code checking in Windows, you must create an automation script file and a batch file. The automation script file must have the same content as the UNIX script. The batch file must contain the CONNECT:Enterprise Command Line Client (Secure FTP) subcommands.

The batch file actually performs the return code checks, but it accesses the information in the automation script file. You can configure the two files to use the **–x** command line parameter to check codes for all commands or the @ symbol in association with a subcommand to check codes for only that command.

The following example shows a Windows batch file that checks the CONNECT:Enterprise Command Line Client (Secure FTP) return code using the **–x** command line parameter.

```
@echo off
:
:Invoke the Enterprise Command Line Client with the Return Code Checking On (–x).
:
CALL ceftp –a auto_file.txt –x
if errorlevel 4 goto PUTF
if errorlevel 3 goto FAILED
if errorlevel 2 goto FAILED
if errorlevel 1 goto FAILED
if errorlevel 0 goto XPASSED
goto FAILED

:PUTF
echo "Account Log did not transfer"
goto END

:FAILED
echo "Enterprise Command Line Client experience a failure."
goto END

:XPASSED
echo "Enterprise Command Line Client subcommand was successful"
goto END

:END
```

In the preceding example, the **CALL ceftp** command references the automation script file auto_file.txt, adding the **–x** command line parameter to initiate return code checking. The auto_file.txt file has the following contents:

```
mypassphrase
open myhost myportnum
mymboxid
mypassword
put C:\sql\repository\accounts.long "$$ID=banktwo BID='Weekly accounts log'"
quit
```

The following Windows batch file checks the CONNECT:Enterprise Command Line Client (Secure FTP) return code without using the **–x** command line parameter.

```
@echo off
:
:Invoke the Enterprise Command Line Client with the Return Code Checking On.
:
CALL ceftp –a AUTOF.TXT
if errorlevel 4 goto PUTF
if errorlevel 3 goto FAILED
if errorlevel 2 goto FAILED
if errorlevel 1 goto FAILED
if errorlevel 0 goto XPASSED
goto FAILED

:PUTF
echo "Account Log did not transfer"
goto END

:FAILED
echo "Enterprise Command Line Client experience a failure."
goto END

:XPASSED
echo "Enterprise Command Line Client subcommand was successful"
goto END

:END
```

For the preceding example, the **CALL ceftp** command references the automation script file autof.txt. In this case, the autof.txt file contains the instruction that initiates return code checking. The autof.txt file has the following contents:

```
mypassphrase
open myhost myportnum
mymboxid
mypassword
@put C:\sql\repository\accounts.log "$$ID=banktwo BID='Weekly accounts log'"
quit
```

The autof.txt file contains an @ symbol next to the **put** subcommand, which initiates return code checking for the **put** subcommand only.

Note: Using the **–x** command line parameter with an automation script file overrides any @ symbol + subcommand combination in the file and performs return code checking for the entire content of CONNECT:Enterprise Command Line Client (Secure FTP).

Glossary

A

Authentication

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

C

Certificate

A certificate is obtained from a certificate authority by generating a certificate signing request (CSR) that contains specific information in a specific format about the requester. It typically contains: (1) distinguished name and public key of the server or client; (2) distinguished name and digital signature of the certificate authority; (3) period of validity (certificates expire and must be renewed); and (4) administrative and extended information. The certificate authority analyzes the CSR fields, validates the accuracy of the fields, generates a certificate, and sends it to the requester.

Certificate Authority

A Certificate Authority (CA) is a company that is responsible for verifying and processing certificate requests, and issuing and managing certificates. The CA you choose should be one that your trading partners will trust. You must meet the requirements for the CA you choose.

Certificate Revocation List

A list of certificates that have been revoked.

Certificate Signing Request

An output file sent through E-mail to a Certificate Authority to request an X.509 certificate.

Cipher Suite

A cryptographic algorithm used to encrypt and decrypt files and messages.

Cipher Text

Data that has been encrypted. Cipher text is unreadable until it has been converted into plain text (decrypted) with a key.

Clear Control Channel (CCC)

The CCC command instructs the FTP command socket to revert to clear text after user-authentication has been performed. The CCC command is only applicable to Secure FTP, and must be enabled at both the client end and server end of the connection.

Configuration File

A file that contains instructions and definitions upon which the system bases its processing.

D

Digital Signature

When a message digest is encrypted with a private key, the result is a digital signature. Digital signatures allow a client to authenticate the server, because the client has the server's public key and can use it to decrypt the signature (created with the private key). The client knows the server is the only one who has the private key, so the server must be the one that sent the message.

Decryption

Any process to convert cipher text back into plain text.

Digital Certificate

A digital certificate is a specifically formatted document that allows you to authenticate or identify yourself to a Web browser, E-mail reader, or a secure server. It contains information on who you are, your relevant details, and who issued the certificate. A certificate can be tied to an E-mail address, a Web server, or a company, and in each case the certificate can be used for different things. A basic E-mail certificate allows you to prove that you are who you say you are. It also allows you to store more information about yourself: your place of work, your telephone contact details—anything you want. The certificate also contains your public key. This means that your certificate becomes associated with your key.

E

Encryption

Any process used to convert plain text into cipher text.

F

FTP

Internet application and network protocol for transferring files between host computers.

J

Java

A programming language that allows development of applications that can run from any kind of device or machine—a PC, a Macintosh computer, a network computer, the Internet, or a mobile phone. The Java language makes it possible to develop software that is portable, modular, and secure.

JDK

The Java Development Kit (JDK) contains the software and tools that developers need to compile, debug, and run applets and applications written using the Java programming language.

JRE

The Java Runtime Environment (also known as the Java Runtime or JRE) consists of the Java virtual machine, the Java platform core classes, and supporting files. It is the runtime part of the Java Development Kit and provides no compiler, debugger, or tools. The JRE is the smallest set of executables and files that constitute the standard Java platform.

K

Key Certificate File

A file stored on the client that contains an encrypted message to identify the client and enable client/server authentication during secure FTP connections.

Keys

A collection of bits, usually stored in a file, which is used to encrypt or decrypt a message.

L

locsite

An FTP syntax subcommand that sets the security configuration parameters.

P

Passphrase

Similar to a password but can be made up of any number of characters. A passphrase is generally thought to be stronger than a password, although not many programs support the use of a passphrase.

Password

A character-limited word or phrase that establishes identity to allow access to a system. Generally, a password is composed of letters, numbers, or both.

Private Key

The secret key of a public-private key cryptography system. This key is used to *sign* outgoing messages, and is used to decrypt incoming messages.

Public Key

The public key of a public-private key cryptography system. This key is used to confirm *signatures* on incoming messages or to encrypt a file or message so that only the holder of the private key can decrypt the file or message. A public key is disseminated freely to clients and servers via certificates signed by a certificate authority (CA).

S

Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol that provides secure communications with transport protocols, including FTP, over TCP/IP. It is an open, non-proprietary Internet protocol that has been widely adopted as standard. SSL ensures point-to-point security, meaning that the data is secured as it is transmitted across a single socket.

Self-Signed Certificate

A certificate that identifies your organization rather than a public certificate authority in the file. It's often used during the period between your request and receipt of a certificate from a public certificate authority. If self-signed certificates are used, the trusted root signing certificate must be installed in the client manually.

Session Key

Crypto key intended to encrypt data for a limited period of time, typically only for a single communications session between a pair of entities. When the session is over, the key is discarded and a new one is established for each new session.

T

Third-Party Certificate

A certificate that identifies an organization other than those that are preconfigured for the application. If third-party certificates are used by the server, the corresponding trusted certificate must be installed in the client manually.

Trusted Root Certificate File

A file stored in a local directory on the client that contains a list of trusted sources. During FTP connections, the client compares the server certificate to the trusted root certificate file to determine if the server certificate was signed by a trusted source. The client can establish a secure FTP connection if a trusted source signed the server certificate.

U

Unsecure Connection

An FTP connection that has no security.

X

X.509 Certificate

Public key certificate specification developed as part of the X.500 directory specification, and often used in public key systems.

Index

R

S

T

U

W

X