IBM Sterling Connect:Enterprise Command Line Client

Implementation Guide

Version 1.3

This edition applies to the 1.3 Version of IBM® Sterling Connect:Enterprise® Command Line Client and to all subsequent releases and modifications until otherwise indicated in new editions.

Before using this information and the product it supports, read the information in *Notices*, on page 67.

Contents

Preface

Chapter 1 About Sterling Connect:Enterprise Command Line Client

Chapter 2 Installing Sterling Connect:Enterprise Command Line Client

Chapter 3 Configuring and Using Sterling Connect:Enterprise Command Line Client with SSH

Glossary

Index

Notices

Preface

The *IBM Sterling Connect:Enterprise Command Line Client Implementation Guide* document is for staff who install and maintain the IBM® Sterling Connect:Enterprise® Command Line Client version 1.3 product.

Read the first two chapters in the book to gain the general knowledge required to install the Sterling Connect:Enterprise Command Line Client product. These chapters introduce you to the basic components, general concepts, and summarize the preinstallation and installation procedures.

This guide assumes knowledge of the UNIX or Microsoft Windows operating system.

Task Overview

The following table guides you to the information required to perform Sterling Connect:Enterprise Command Line Clienttasks:

Task	Reference
Understanding Sterling Connect:Enterprise Command Line Clientsecurity features and operation	*Chapter 1, About Sterling Connect:Enterprise Command Line Client*
Installing Sterling Connect:Enterprise Command Line Clienton a UNIX OS	*Chapter 2, Installing Sterling Connect:Enterprise Command Line Client*
Installing Sterling Connect:Enterprise Command Line Clienton Windows operating systems	*Chapter 2, Installing Sterling Connect:Enterprise Command Line Client*
Configuring SSH-2 connections, connecting to remote hosts, issuing commands, and using automation scripts.	*Chapter 3, Configuring and Using Sterling Connect:Enterprise Command Line Client with SSH*
Configuring SSL connections, connecting to remote hosts, issuing commands, and using automation scripts.	Chapter 4, *Configuring and Using Sterling Connect:Enterprise Command Line Client with SSL Secure FTP*
Configuring unsecure connections, connecting to remote hosts, issuing commands, and using automation scripts.	Chapter 5, *Configuring and Using Sterling Connect:Enterprise Command Line Client for FTP*

Chapter 1

About Sterling Connect:Enterprise Command Line Client

Sterling Connect:Enterprise Command Line Client provides a means for manually and automatically exchanging files with the Sterling Connect:Enterprise server during secure and unsecure connections. With Sterling Connect:Enterprise Command Line Client, you can use standard FTP commands for all connections. During secure connections, you can use the Secure Shell version 2 (SSH-2) or the Secure Sockets Layer (SSL) protocol to exchange files with the Sterling Connect:Enterprise server, which provides complete local and remote security over the Internet.

Security Essentials

Sterling Connect:Enterprise Command Line Client provides the following security essentials:

- Secrecy—You can encrypt messages, ensuring that they can only be read using the encryption key that you provide. You can also decrypt messages sent to you.
- Authentication—You can authenticate the party that you are exchanging data with to ensure they are who they say they are.
- 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.

Selecting Your Protocol

When using Sterling Connect:Enterprise Command Line Client you have a choice of three protocols: FTP, Secure FTP using SSL, and the SSH-2 SFTP protocol. An FTP connection is unsecure. Both SSL and SSH connections provide the essential security features described in *Security Essentials* on page 9. If the server you are connecting to requires either SSL or SSH, you will need to connect based on their preference. If it is your choice to use SSL or SSH, consider the following:

- SSH requires a single dedicated socket.
- SSH uses public and private keys generated using a utility included with the product. Therefore, no certificates or maintenance of expired certificates is required.
- SSH allows you to select a prioritized list of authentication mechanisms and encryption algorithms.
- SSL requires at least two dedicated sockets, possibly a range of sockets.
- SSL requires certificates from a Certificate Authority (CA). This requires additional cost and maintenance because CA certificates expire.

Chapter 2

Installing Sterling Connect:Enterprise Command Line Client

This chapter describes installing Sterling Connect:Enterprise Command Line Client on a computer running the UNIX or Microsoft Windows operating systems. For a list of supported platforms, refer to *IBM Sterling Connect:Enterprise Command Line Client Release Note*s.

Before You Begin

Before you install Sterling Connect:Enterprise Command Line Client:

- Read *IBM Sterling Connect:Enterprise Command Line Client Release Note*s and verify that your system meets the installation requirements and that you have the appropriate Java components installed on your computer.
- If you previously installed a demonstration version, you must uninstall before proceeding with a new installation.
- If you are using SSH or SSL protocol, verify that the remote server supports the protocol.

Installing Sterling Connect:Enterprise Command Line Client on a UNIX System

Automated installation scripts control installation of Sterling Connect:Enterprise Command Line Client. 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 Sterling Connect:Enterprise Command Line Client 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. If you are installing from a CD-ROM, load the Sterling Connect:Enterprise Command Line Client CD-ROM in the CD-ROM drive and record its mount point.
3. Navigate to the directory where the application files are located. If you are installing from a CD-ROM, this is the root directory of the CD-ROM.
4. Type the following:

```
CECLC.Version.Platform.bin
```

The following screens are displayed:

```
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

================================================================================
IBM Sterling Connect:Enterprise Command Line Client V1.3.00(created with
InstallAnywhere)
----------------------------------------------------------------------------------
--------

Preparing CONSOLE Mode Installation...
```

```
Introduction
------------
InstallAnywhere will guide you through the installation of
IBM Sterling Connect:Enterprise Command Line Client V1.3.00
Respond to each prompt to proceed to the next step in the installation.
If you want to change something on a previous step, type 'back'.
You may cancel this installation at any time by typing 'quit'.
PRESS <ENTER> TO CONTINUE:
```

5. Read the information on the screen, and then press **Enter** to begin the installation.

   The following screen is displayed:

```
Choose Installation Folder
--------------------------

Where would you like to install?

  Default Install Folder: /data/CLC/clc1300

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
      : /data/CLC/clc1300
```

6. Type the full destination directory path for the Sterling Connect:Enterprise Command Line Client 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:

```
INSTALL FOLDER IS: /data/CLC/clcftp
   IS THIS CORRECT? (Y/N): y
```

7. Press **Enter** if the destination directory is correct. If it is incorrect, type **n** and repeat step 6 on page 13 to revise the path.

   ❖ If the destination directory does not exist, it is created.

   ❖ If the destination directory exists, the following screen is displayed:

```
IBM Sterling Connect:Enterprise CLC Installation Detected
---------------------------------------------------------

An installation of IBM Sterling Connect:Enterprise CLC already exists in

   /data/CLC/Manual_CLC1300_build25

You may either overwrite the existing installation, or specify
an alternate directory to perform a new installation.  If you
choose to overwrite the existing installation, all files in the
installation will be deleted.

    1- Overwrite existing installation
    2- Create new installation

ENTER THE NUMBER OF THE DESIRED CHOICE: 1
```

   Type 1 to overwrite the existing installation or 2 to create a new installation.

A pre-instalation summary screen is displayed:

```
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Product Name:
    IBM Sterling Connect:Enterprise Command Line Client V1.3.00

Install Folder:
    /data/CLC/clc1300

Disk Space Information (for Installation Target):
    Required:  221,325,357 bytes
    Available: 40,478,111,744 bytes

PRESS <ENTER> TO CONTINUE:
```

8. Press Enter to continue with the installation. When the installatuion is complete, the following screen is displayed:

```
Installation Complete
---------------------

Congratulations. IBM Sterling Connect:Enterprise Command Line Client V1.3.00
has been successfully installed to:

   /data/CLC/clc1300

PRESS <ENTER> TO EXIT THE INSTALLER:
```

9. Press **Enter** to exit the installer.

Sterling Connect:Enterprise Command Line Client is now fully operational.

Setting Environment Variables

To run Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client on a Microsoft Windows Operating System

If you previously installed a demonstration version of Sterling Connect:Enterprise Command Line Client, uninstall the demo version before completing this procedure. Refer to *Uninstalling Sterling Connect:Enterprise Command Line Client from a Microsoft Windows Operating System on page 15*.

Install Sterling Connect:Enterprise Command Line Client on a Microsoft Windows operating system, using the following steps:

1. Exit all Microsoft Windows programs that are running.
2. Insert the Sterling Connect:Enterprise Command Line Client CD-ROM into the CD-ROM drive.

   If the Autorun option is enabled for the CD-ROM drive, the Sterling Connect:Enterprise Command Line Client installation setup detects the type of Microsoft 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 Yes to **Accept**.
6. In the **Select Directory** dialog box, click **Browse** to select a directory or accept the default and click **Next**.
7. In the **Start Copying Files** dialog box, click **Next** to start copying files.
8. After copying the files into the program folder, the installation program displays the **InstallShield Wizard Complete** dialog box. Click **Finish** to complete the installation.

Uninstalling Sterling Connect:Enterprise Command Line Client from a Microsoft Windows Operating System

The Microsoft Windows uninstall program removes the Sterling Connect:Enterprise Command Line Client application, its components, program folder, program items, and all other settings. To uninstall Sterling Connect:Enterprise Command Line Client and all of its components, follow these steps:

1. From your Microsoft Windows desktop, click **Start**.
2. Click **Settings**.
3. Click **Add/Remove Programs**.
4. Select Sterling Connect:Enterprise Command Line Client from the list of available programs and click **Add/Remove**.
5. Click **OK** to complete the uninstall procedure.

Chapter 3

Configuring and Using Sterling Connect:Enterprise Command Line Client with SSH

The SSH functionality of Sterling Connect:Enterprise Command Line Client allows you to connect to SSH-2 SFTP servers. This product does not connect using Telnet, SCP, or any other connection protocol. Use the information in this chapter to configure Sterling Connect:Enterprise Command Line Client for SSH-2, connect to SSH-2 servers, issue commands, and write automation scripts.

Before You Begin

Before you start Sterling Connect:Enterprise Command Line Client, ensure that the remote server has SSH enabled. Gather the following information from your host site administrator to access the server:

- User ID or Mailbox ID
- Password
- IP address or host name of the server
- SSH port number
- Host public key file from server you are trading with (not required if you automatically trust)
- Authentication methods available: password and/or public key. If public key, key format that is required: IETF SECH or OpenSSH.

Identify your security preferences from the following options:

- Authentication method: password and/or public key
- Order of preferred ciphers to use for data encryption
- Order of preferred Message Authentication Algorithms (MACs) to use to verify data integrity
- Network path and firewall navigation information

Sterling Connect:Enterprise Command Line Client does not perform host-based authentication.

Configuring SSH

To configure SSH, you must complete the following:

1. Create the SSH authentication key file. If you or the server you are connected to requires public key authentication, you will need an authentication key pair. If you do not already have an authentication key, create one using *Creating the SSH Authentication Key* on page 18.
2. Update the SSH known hosts file. This file contains the public host key for all servers you connect to. Sterling Connect:Enterprise Command Line Client uses this file to verify that you are connecting to a trusted host. Refer to *Updating the SSH Known Hosts File* on page 19.
3. Configure the SSH parameters. You can use the default system parameters or customize them for your environment. Refer to *Configuring SSH Parameters* on page 21.
4. Configure the SSH logging parameters. You can use the default system parameters for logging or change the level of logging details. Refer to *Configure SSH Logging* on page 22.

Creating the SSH Authentication Key

The SSH authentication key file contains the private key file used for key authentication. Use the following procedure to create the key. After the key is created, you will need to provide the public key to the administrator of the server you are connecting to.

1. Open command prompt and navigate to the Sterling Connect:Enterprise Command Line Client installation directory.
2. Type the following command and press **Enter**:

   ```
   ssh-keygen -b bits -t type -f name
   ```

   The following table describes the parameters:

Parameter	Description
bits	Number of bits to use in the key. Valid values are 768–1024 (must be a multiple of 64). Larger keys are stronger. Default is 1024.
type	The type of key to create. Valid values are DSA and RSA. Default is DSA.
name	Path and file name to where you want to create the file. If you provide only a file name, it will be stored in the Sterling Connect:Enterprise Command Line Client installation directory.

3. Type the passphrase to use for the key and press **Enter**.
4. Confirm the passphrase and press **Enter**.

   Two files are created: *name*.pub and *name,* where *name* is the name you gave the key in step 1. These keys are in OpenSSH format. The *name*.pub file is the file you provide to the server.

   Many servers, including Sterling Connect:Enterprise require public keys to be in OpenSSH format. Refer to *Converting a Public Key to OpenSSH Format* on page 19 to convert to OpenSSH format if you have a IETF SECSH format key.

Converting a Public Key to OpenSSH Format

To communicate with an OpenSSH server (including Sterling Connect:Enterprise), the public key must be in OpenSSH format. Use the following procedure to convert an IETF SECSH key to OpenSSH:

1. Open command prompt and navigate to the Sterling Connect:Enterprise Command Line Client installation directory.
2. Type the following command and press **Enter**:

```
ssh-keygen -i -f name.pub > name.open
```

where *name* is the name of the public key you want to convert. The resulting file, *name*.open is the public key in OpenSSH format. This is the file you provide to the OpenSSH server.

Note: The ssh-keygen utility does not convert private keys between IETF SESC and OpenSSH formats.

Updating the SSH Known Hosts File

The known hosts file contains the public host key for all servers you are connecting to. Sterling Connect:Enterprise Command Line Client uses this file to verify that you are connecting to a trusted host. To connect to a server, you must have the server's public host key in this file. Use the following procedures to create the known hosts file, add a host key to the known hosts file, or to change the name and location of the known hosts file.

Creating the Known Hosts File

If you do not use the automatic trust feature of Sterling Connect:Enterprise Command Line Client, you must create the known hosts file manually. Use the following procedure. For more information on the automatic trust feature, refer to *Connecting to an SSH Server from a UNIX Operating System* on page 23 or *Connecting to an SSH Server from a Microsoft Windows Operating System* on page 25.

1. Navigate to the Sterling Connect:Enterprise Command Line Client installation directory.
2. Create a directory named **.ssh**.
3. Using a text editor, create a file in the .ssh directory called **known_hosts**. Refer to *Manually Add a Public Host Key* on page 19 to add a public host key.

Manually Add a Public Host Key

Use the following procedure to manually add the public host key:

1. Obtain the public host key from the administrator of the server you are connecting to.
2. Open the **known_hosts** file. The default location is <*install*>/.ssh/**known_hosts**, where <install> is the Sterling Connect:Enterprise Command Line Client installation directory.
3. If there is already a public host key in the file, insert a carriage return at the end of the file.

4. Add the host names (comma separated) and IP addresses. These items are in bold in the following example:

```
server01,1.11.1.11
ssh-rsaAAAAB3NzaC1yc2EAAAABIwAAAIEAy/G47p7RXsR+1/DfbvqmokVZKUDXfCQt1VILoVkmBTdoqH
t0Z/2OSfFEdyRFw28ikyHody2GnoItSuwERcJZaei9GsCaM7EgDKdjf0adjfoakdXQm5PB9H7cOhQURL6
9zlQPuaOrj14xcoLRCRCRdTI3MMdidlrTndiDDDL:ddl=
server02,11.1.11.1
ssh-dssAAAAB3NzaC1kc3MAAACBAMRptv+Ixu+X4CIKu8YieRT6vVYrFkOGFVQsfl2k1hZkjPTGCuYBWP
ZZmb7haAn3alB6wZTogpRg8t1DSveUJ4FV/H53mjXvXmIwTJvEjz0weytif4xiBwQGZxlYfSYvfl/JGHZ
sJAIYaL1x2WR/7Uz6lYYEaZL+S4+2+xpJriIDOdiadDIDDoasddasdhn44qSFmjgGzPd1ZnX3kVJK1dHF
B1RJ/rtYPMiRf4UX5FG/X4BH+fJFvlMQ0u4TqCgQv?>?<DKDIODOSDID(0-eda-dfkad-adadoXq5lgzu
mTHgYDm6A2XXhkETMOzhtr4GA5Zkzti4nXEjDQPexp3APeUDJXIi7N50oB8TIQUWIfuLRDZnKGKEc4jwt
x0iON/XJioYVPIgnTEpLJwlzuTJKL/DGF9pqUuuoZkR0t8AClY3jvunADhXXzNg0Q4tFTUwEp30e8XQ+L
Q=
```

5. Add the public host key immediately after the host name and IP address. These items are in bold in the following example:

```
server01,1.11.1.11
ssh-rsaAAAAB3NzaC1yc2EAAAABIwAAAIEAy/G47p7RXsR+1/DfbvqmokVZKUDXfCQt1VILoVkmBTdoqH
t0Z/2OSfFEdyRFw28ikyHody2GnoItSuwERcJZaei9GsCaM7EgDKdjf0adjfoakdXQm5PB9H7cOhQURL6
9zlQPuaOrj14xcoLRCRCRdTI3MMdidlrTndiDDDL:ddl=
server02,11.1.11.1
ssh-dssAAAAB3NzaC1kc3MAAACBAMRptv+Ixu+X4CIKu8YieRT6vVYrFkOGFVQsfl2k1hZkjPTGCuYBWP
ZZmb7haAn3alB6wZTogpRg8t1DSveUJ4FV/H53mjXvXmIwTJvEjz0weytif4xiBwQGZxlYfSYvfl/JGHZ
sJAIYaL1x2WR/7Uz6lYYEaZL+S4+2+xpJriIDOdiadDIDDoasddasdhn44qSFmjgGzPd1ZnX3kVJK1dHF
B1RJ/rtYPMiRf4UX5FG/X4BH+fJFvlMQ0u4TqCgQv?>?<DKDIODOSDID(0-eda-dfkad-adadoXq5lgzu
mTHgYDm6A2XXhkETMOzhtr4GA5Zkzti4nXEjDQPexp3APeUDJXIi7N50oB8TIQUWIfuLRDZnKGKEc4jwt
x0iON/XJioYVPIgnTEpLJwlzuTJKL/DGF9pqUuuoZkR0t8AClY3jvunADhXXzNg0Q4tFTUwEp30e8XQ+L
Q=
```

6. Save the known hosts file.

You can add an unlimited number of keys in the known hosts file. You can add up to two entries for a single host but they must be different types (one DSA and one RSA).

Changing the Name or Location of the Known Hosts File

When you install Sterling Connect:Enterprise Command Line Client, the default location for the known_hosts file is <*install*>/.ssh/**known_hosts** where <*install*> is the Sterling Connect:Enterprise Command Line Client installation directory. You can change the name or location of the known hosts file using the following procedure:

1. Create the file you want to use for the known hosts file.
2. Open the **sshclc.properties** file located in the Sterling Connect:Enterprise Command Line Client installation directory.
3. Uncomment the **knownhostkeyfile** parameter (if necessary) and specify the path and file name of the file created in step 1. For example:

```
knownhostkeyfile=c:/hosts/myknownhostsfile
```

Use only the file name if it is located in the <*install*>/.ssh directory. Use the full path and file name if it is not.

Configuring SSH Parameters

The Sterling Connect:Enterprise Command Line Client installation creates an SSH configuration file, **sshclc.properties**, in the installation directory. This file includes the SSH parameters that Sterling Connect:Enterprise Command Line Client accesses to establish secure FTP SSH connections. Use this file to override system defaults (system defaults are indicated in the table for step 2). Use the following procedure:

1. From the Sterling Connect:Enterprise Command Line Client installation directory, open the **sshclc.properties** file.
2. Remove the # from the beginning of each line you want to use and define the parameters as follows:

Parameter	Description
cipherlist	Indicates the ciphers to use for data encryption in order of preference. You can remove ciphers from the list and you can reorder the list. You cannot add ciphers to the list. Available ciphers (in default order of preference): 3des-cbc, blowfish-cbc, aes256-cbc, aes192-cbc, aes128-cbc. If this field is left blank, the following list applies: blowfish-cbc, aes192-cbc, aes128-cbc, aes256-cbc, 3des-cbc.
compression	Indicates whether to compress outbound data. Yes = Compress data. No = Do not compress data. **Default is Yes.**
keyfile	Indicates the location of the SSH client key file. Specify absolute path and file name. If you do not have an SSH client key file, create one using the ssh-keygen utility that is included with all SSH applications.
knownhostkeyfile	Indicates the location of the known hosts file. Specify absolute path and file name. Note the following: • If you specify a file name only, Sterling Connect:Enterprise Command Line Client searches in the <*install*>/.ssh directory, where <*install*> is the installation directory. • If you do not specify this parameter, Sterling Connect:Enterprise Command Line Client searches in the <*install*>/.ssh directory for known_hosts. **Default is known_hosts (located in <*install*>/.ssh).**
maclist	Specifies the Message Authentication Algorithms (MACs) to use to verify data integrity in order of preference. You can remove MACs from the list and you can reorder the list. You cannot add MACs to the list. Available MACs (in default order of preference): hmac-sha1, hmac-md5-96, hmac-md5, hmac-sha1-96 If this field is left blank, the following list applies: hmac-sha1, hmac-md5, hmac-sha1-96, hmac-md5-96
passwordauth	Indicates that the client can request password authentication. Yes = Sterling Connect:Enterprise Command Line Client will perform password authentication if supported by the server. No = Sterling Connect:Enterprise Command Line Client will not perform password authentication if supported by the server. If both password authentication and public key authentication are set to Yes, public key authentication is attempted first. **Default is Yes.**
publickeyauth	Indicates that the client can request public key authentication. Yes = Sterling Connect:Enterprise Command Line Client will perform public key authentication if supported by the server. No = Sterling Connect:Enterprise Command Line Client will not perform public key password authentication if supported by the server. If both password authentication and public key authentication are set to Yes, public key authentication is attempted first. **Default is No.**
sockettimeout	Time to wait for activity before disconnect, in seconds. **Default is 1800.**

3. Save the **sshclc.properties** file. The following example illustrates the contents of a configuration file:

```
#
# Licensed Materials - Property of IBM
# IBM Sterling Connect:Enterprise(R) Command Line Client
# (C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Sample ResourceBundle properties file
#sockettimeout time=seconds, 1800=30 minutes
sockettimeout=1800
compression=no
passwordauth=yes
publickeyauth=no
#cipherlist=aes128-ctr,aes128-cbc,3des-ctr,3des-cbc,blowfish-cbc,aes192-cbc,aes2
56-cbc
#maclist=hmac-sha1,hmac-md5-96,hmac-md5,hmac-sha1-96
#keyfile=c:/usr/newkey
#knownhostkeyfile=myknown_host
language=en
country=us
```

Configure SSH Logging

The Sterling Connect:Enterprise Command Line Client installation creates an SSH log file, **sshlog.properties**, in the installation directory. This file includes the parameters to turn on logging and to set the logging level. Use the following procedure to update this file:

1. From the Sterling Connect:Enterprise Command Line Client installation directory, open the **sshlog.properties** file.
2. Update the following parameters:

Parameter	Description

com.sterlingcommerce.sshclc.SftpLogger.defaultlog	Indicates the level of logging. Valid values are:
	FATAL = Logs only fatal messages.
	ERROR = Logs all fatal and nonfatal error messages.
	WARN = Logs all ERROR plus additional warnings.
	INFO = General information.
	DEBUG = Debug information.
	TRACE = Trace information.
	Default is INFO.

com.sterlingcommerce.sshclc.SftpLogger.writelogfile	Enables or disables logging. True = Messages are written to the log file. False = Messages are not written to the log file. **Default is True.**

3. The following example illustrates the contents of the **sshlog.properties** file:

```
#
# Licensed Materials - Property of IBM
# IBM Sterling Connect:Enterprise(R) Command Line Client
# (C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.
#
############################################################
#       Default Logging Configuration File
#
############################################################



############################################################
# Handler specific properties.
# Describes specific configuration info for Handlers.
############################################################

# default file output is in user's runtime directory.
# Limit the message that are printed on the console to INFO and above.
# the valid values are DEBUG,INFO,WARN,ERROR,FATAL
# the writelogfile value=true will write to log file
com.sterlingcommerce.sshclc.SftpLogger.defaultlog = INFO
com.sterlingcommerce.sshclc.SftpLogger.writelogfile = true
```

By default, log messages are written to the **clcsshlog.trc** file. This file is located in the directory where **cesshsftp** is run. The log file is overwritten each time **clcsshsftp** is run.

Connecting to an SSH Server from a UNIX Operating System

After you configure SSH or use the default configuration, use the following procedure to connect to an SSH server if you are using Sterling Connect:Enterprise Command Line Client on a UNIX operating system.

1. For UNIX operating systems, navigate to the Sterling Connect:Enterprise Command Line Client installation directory and type the following:

```
./cesshsftp -a autoscript -d newloglevel logfilename -h -L newkeyfile -P
passphrase -r -s propertyfile -x
```

The following table describes the optional parameters:

Parameter	Description
-a *autoscript*	Specifies the location and file name of the automation script file. Refer to *Writing Automation Scripts* on page 27 for more information on creating automation scripts.
-d *newloglevel newlogfilename*	Indicate *newloglevel* to override the default log level. Valid values are: 0 = No logging. 1 = General information. 2 = Debug information. Indicate *newlogfilename* to override the default log file.

Parameter	Description
-h	Returns the command line syntax
-L *newkeyfile*	Indicates to override the default public authentication key file with *newkeyfile*. If you use this parameter, you can use the -P parameter to specify the passphrase. If you do not include the -P parameter in the command, you will be prompted for the passphrase.
-P *passphrase*	Specifies the passphrase associated with the *newkeyfile* specified with the -L parameter.
-r	Returns the product name, release, and build.
-s *newpropertyfile*	Indicates to override the default property file with *newpropertyfile.*
-x	Turns on result code exiting for the entire instance of the client. Used with automation scripts to exit if a command fails.

The following prompt is displayed:

```
======================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
======================================================================

cesshsftp> Enter username@hostname:
```

2. Type the user, host name, and port (if other than default port 22) you want to connect to and press **Enter**.

   - If the public host key of the server is not in your known hosts file, the following prompt is displayed:

     ```
     The authenticity of host 'hostname' can't be established.
     RSA key fingerprint is b6:9b:b3:70:cd:5f:f9:cf:45:32:96:17:6d:83:d9:b5.
     Are you sure you want to continue connecting?
     cesshsftp> Please enter Yes|No
     ```

     Do one of the following:

     a.Type **Yes** and press **Enter** to trust the server for the current session. Sterling Connect:Enterprise Command Line Client will trust the server until you log off. The server's public host key is not added to the known hosts file.

     b.Type **Always** and press **Enter** to trust the server permanently. Sterling Connect:Enterprise Command Line Client adds the server's public host key to the known hosts file.

     c.Type **No** and press **Enter** to not trust the server.

3. If the public host key of the server is in your known hosts file, the following prompt is displayed::

   ```
   cesshsftp> Enter Password:
   ```

4. Type the password associated with the user name and press **Enter**. The following prompt is displayed:

   ```
   cesshsftp>
   ```

   You can begin issuing commands. Refer to *Issuing Subcommands* on page 26 for more information.

Connecting to an SSH Server from a Microsoft Windows Operating System

After you configure SSH or use the default configuration, use the following procedure to connect to an SSH server if you are using Sterling Connect:Enterprise Command Line Client on a Microsoft Windows operating system.

1. Click the start menu and select **Programs**> **Sterling Commerce**> **Command Line Client**> **Run Command Line Client(SSH)**. The following prompt is displayed:

```
======================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
======================================================================

cesshsftp> Enter username@hostname:
```

2. Type the user, host name, and port you want to connect to and press **Enter**.

   - If the public host key of the server is not in your known hosts file, the following prompt is displayed:

```
The authenticity of host 'hostname' can't be established.
RSA key fingerprint is b6:9b:b3:70:cd:5f:f9:cf:45:32:96:17:6d:83:d9:b5.
Are you sure you want to continue connecting?
cesshsftp> Please enter Yes|No
```

   Do one of the following:

   a. Type **Yes** and press **Enter** to trust the server for the current session. Sterling Connect:Enterprise Command Line Client will trust the server until you log off. The server’s public host key is not added to the known hosts file.

   b. Type **Always** and press **Enter** to trust the server permanently. The server’s public host key is added to the known hosts file.

   c. Type **No** and press **Enter** to not trust the server. If you select this option, refer to *Updating the SSH Known Hosts File* on page 19 before attempting to connect to the server again.

3. If the public host key of the server is in your known hosts file, the following prompt is displayed:

```
cesshsftp> Enter Password:
```

4. Type the password associated with the user name and press **Enter**. The following prompt is displayed:

```
cesshsftp>
```

You can begin issuing commands. Refer to *Issuing Subcommands* on page 26 for more information.

Overriding the SSH Configuration File When Connecting

You can override SSH configuration parameters when connecting to a server. Use the following procedure:

1. From a command line, navigate to the Sterling Connect:Enterprise Command Line Client installation directory.
2. Type the following command:

```
cesshsftp -a autoscript -d newloglevel logfilename -h -L newkeyfile -P passphrase
-r -s propertyfile -x
```

The following table describes the parameters. No parameters are required.

Parameter	Description
-a *autoscript*	Specifies the location and file name of the automation script file. Refer to *Writing Automation Scripts* on page 27 for more information on creating automation scripts.
-d *newloglevel newlogfilename*	Indicate *newloglevel* to override the current log level. Valid values are: 0 = No logging. 1 = General information. 2 = Debug information. Indicate *newlogfilename* to override the current log file.
-h	Returns the command line syntax.
-L *newkeyfile*	Indicates to override the current public authentication key file with *newkeyfile*. If you use this parameter, you can use the -P parameter to specify the passphrase. If you do not include the -P parameter in the command, you will be prompted for the passphrase.
-P *passphrase*	Specifies the passphrase associated with the *newkeyfile* specified with the -L parameter.
-r	Returns the product name, release, and build.
-s *newpropertyfile*	Indicates to override the current property file with *newpropertyfile.*
-x	Turns on result code exiting for the entire instance of the client. Used with automation scripts to exit if a command fails.

3. Continue with step 2 on page 25.

Issuing Subcommands

After you connect to the SSH server, you can use the following subcommands. File names with spaces must be enclosed with double quotes (" "). If you are connecting to a Sterling Connect:Enterprise server, refer to the *IBM Sterling Connect:Enterprise for UNIX Remote User's Guide* for a detailed description of and examples for sending standard SSH syntax and $$ commands.

Subcommand	Description
cd	Changes the working directory.

Subcommand	Description
delete/rm/del	Flags a document of data as deleted.
dir	Requests a formatted listing of documents from the host site.
get	Requests a 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.
lpwd	Displays the current local directory.
ls	Displays the current remote directory for an OpenSSH server. Displays current mailbox and batch for a Sterling Connect:Enterprise SSH server.
put	Sends a document of data to the host site.
pwd	Prints the working directory.
quit|bye	Closes all connections and exits the client.
rename	Allows you to rename a remote file or batch. If you are renaming a batch number to a batch name, the batch number must start with a #. Examples: rename oldfile newfile changes oldfile to newfile rename #9999 newfile changes batch number 9999 to newfile

Writing Automation Scripts

Sterling Connect:Enterprise Command Line Client provides automated scripting capabilities for file exchanges. This scripting capability eliminates the need for you to run Sterling Connect:Enterprise Command Line Client manually. This feature works on any platform that supports Java. You invoke automation scripts using the -a parameter when starting Sterling Connect:Enterprise Command Line Client.

Note: You cannot use the automatic trust feature in a script. You must have the host key of the server you are connecting to in your known hosts file.

Writing Basic Automation Scripts

Following are three scripts that demonstrate different authentication routines:

- The following script connects to server01 port 22 using public key authentication, downloads file01 in mailbox01, and quits:

```
open server01 23
userID
keyfilepath
passphrase
cd mailbox01
get file01
quit
```

Note: It is not necessary to provide the keyfile or the passphrase if they are configured in the property file or if they were included in the command line parameter.

- The following script connects to server01 port 22 using password authentication, downloads file01 in mailbox01, and quits:

```
open server01 22
userID
password
cd mailbox01
get file01
quit
```

- The following script connects to server01 port 22 using public key and password authentication, downloads file01 in mailbox01, and quits:

```
open server01 22
userID
keyfilename
passphrase
password
cd mailbox01
get file01
quit
```

Chapter 4

Configuring and Using Sterling Connect:Enterprise Command Line Client with SSL Secure FTP

The SSL functionality of Sterling Connect:Enterprise Command Line Client allows you to connect to SSL servers over Secure FTP. Use the information in this chapter to configure Sterling Connect:Enterprise Command Line Client for SSL, connect to SSL servers, issue commands, and write automation scripts.

Before You Begin

Before you start Sterling Connect:Enterprise Command Line Client, ensure that the remote server has SSL Secure FTP enabled. Gather the following information from your host site administrator to access the server:

- Mailbox ID
- Mailbox password
- IP address or host name of the Sterling Connect:Enterprise server
- SSL Secure FTP listening port number
- Authentication level
- Encryption strength
- Network path and firewall navigation information
- Trusted root entry for the server certificate

  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 firewalls do not work with secure FTP.

Configuring for SSL

Before you can use Sterling Connect:Enterprise Command Line Client to establish SSL connections, you must configure the system to support server or client-server authentication.

The use of certificates is an important component of Sterling Connect:Enterprise Command Line Client 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 Sterling 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

Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client. If the server certificate is compromised, the administrator of the FTP server must notify all trading partners.

Example: Trusted Root Certificate File (trusted.txt)

```
Thawte Server CA - exp. Dec 31, 2020
-----BEGIN CERTIFICATE-----
MIIDEzCCAnygAwIBAgIBATANBgkqhkiG9w0BAQQFADCBxDELMAkGA1UEBhMCWkEx
FTATBgNVBAgTDFdlc3Rlcm4gQ2FwZTESMBAGA1UEBxMJQ2FwZSBUb3duMR0wGwYD
VQQKExRUaGF3dGUgQ29uc3VsdGluZyBjYzEoMCYGA1UECxMfQ2VydGlmaWNhdGlv
biBTZXJ2aWNlcyBEaXZpc2lvbjEZMBcGA1UEAxMQVGhhd3RlIFNlcnZlciBDQTEm
MCQGCSqGSIb3DQEJARYXc2VydmVyLWNlcnRzQHRoYXd0ZS5jb20wHhcNOTYwODAx
MDAwMDAwWhcNMjAxMjMxMjM1OTU5WjCBxDELMAkGA1UEBhMCWkExFTATBgNVBAgT
DFdlc3Rlcm4gQ2FwZTESMBAGA1UEBxMJQ2FwZSBUb3duMR0wGwYDVQQKExRUaGF3
dGUgQ29uc3VsdGluZyBjYzEoMCYGA1UECxMfQ2VydGlmaWNhdGlvbiBTZXJ2aWNl
cyBEaXZpc2lvbjEZMBcGA1UEAxMQVGhhd3RlIFNlcnZlciBDQTEmMCQGCSqGSIb3
DQEJARYXc2VydmVyLWNlcnRzQHRoYXd0ZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBANOkUG7I/1Zr5s9dtuoMaHVHoqrC2oQl/Kj0R1HahbUgdJSGHg91
yekIYfUGbTBuFRkC6VLAYttNmZ7iagxEOM3+vuNkCXDF/rFrKbYvScg71CcEJRCX
L+eQbcAoQpnXTEPew/UhbVSfXcNY4cDk2VuwuNy0e982OsK1ZiIS1ocNAgMBAAGj
EzARMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEEBQADgYEAB/pMaVz7lcxG
7oWDTSEwjsrZqG9JGubaUeNgcGyEYRGhGshIPllDfU+VPaGLtwtimHp1it2ITk6e
QNuozDJ0uW8NxuOzRAvZim+aKZuZGCg70eNAKJpaPNW15yAbi8qkq43pUdniTCxZ
qdq5snUb9kLy78fyGPmJvKP/iiMucEc=
-----END CERTIFICATE-----
Verisign Class 3 Public Primary CA Expires Aug 1 2028
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCEHC65B0Q2Sk0tjjKewPMur8wDQYJKoZIhvcNAQECBQAwXzELMAkG
A1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFz
cyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTk2
MDEyOTAwMDAwMFoXDTI4MDgwMTIzNTk1OVowXzELMAkGA1UEBhMCVVMxFzAVBgNV
BAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAzIFB1YmxpYyBQcmlt
YXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GN
ADCBiQKBgQDJXFme8huKARS0EN8EQNvjV69qRUCPhAwL0TPZ2RHP7gJYHyX3KqhE
BarsAx94f56TuZoAqiN91qyFomNFx3InzPRMxnVx0jnvT0Lwdd8KkMaOIG+YD/is
I19wKTakyYbnsZogy1Olhec9vn2a/iRFM9x2Fe0PonFkTGUugWhFpwIDAQABMA0G
CSqGSIb3DQEBAgUAA4GBALtMEivPLCYATxQT3ab7/AoRhIzzKBxnki98tsX63/Do
lbwdj2wsqFHMc9ikwFPwTtYmwHYBV4GSXiHx0bH/59AhWM1pF+NEHJwZRDmJXNyc
AA9WjQKZ7aKQRUzkuxCkPfAyAw7xzvjoyVGM5mKf5p/AfbdynMk2OmufTqj/ZA1k
-----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 IBM® Sterling Certificate Wizard to create key certificate files. See the IBM Sterling Certificate Wizard Readme file for instructions on installing Certificate Wizard.

With Sterling 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 IBM Sterling Certificate Wizard Help.

Note: A fatal error is returned and the Sterling 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

Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client and Sterling 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 Sterling 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 Sterling Connect:Enterprise for UNIX version 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 *Issuing Commands* on page 4-36, 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. This allows Statefull Packet Inspection (SPI) firewalls to correctly handle the FTP session. 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 Sterling Connect:Enterprise for UNIX version 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.
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 *Issuing Commands* on page 4-36, 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.

Connecting to an SSL Server

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:

- SSL 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 SSL Configuration File Security Parameters from the Command Line to Establish a Connection* on page 4-35.

Defining Default Security Parameters in the SSL Configuration File

The Sterling Connect:Enterprise Command Line Client installation creates two SSL configuration files, sample_secureftp.cfg and secureftp.cfg, in the installation directory. Both files include the SSL parameters that Sterling Connect:Enterprise Command Line Client accesses to establish secure FTP SSL 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.

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

Parameter	Description
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 Sterling Connect:Enterprise Command Line Client 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, Sterling Connect:Enterprise Command Line Client uses the default entries in the secureftp.cfg file.

Overriding SSL Configuration File Security Parameters from the Command Line to Establish a Connection

You can define and override the following parameters in the configuration file from the command line:

- keycert filename—key certificate file for client-server authentication (optional)
- trusted filename—trusted root certificate file for server authentication
- strong|weak|all—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 Sterling 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.

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

```
ceftp-s>
```

Overriding the SSL Configuration File Security Parameters Using the locsite Command

You can use the locsite subcommand to override the the following parameters:

- 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

Use the following procedure:

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

```
ceftp>locsite keycert=/ceftp_dir/keycert.txt
ceftp>locsite trusted=/ceftp_dir/trusted.txt
ceftp>locsite strength=strong
ceftp>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 4-38 for other ways to use the locsite command.

Issuing Commands

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

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

Parameter	Description
–a *automation script filename*	Specifies the location and file name of the automation script file; see *Microsoft Windows and UNIX Automation Scripts* on page 4-43 for more information.

Parameter	Description
–c *key certificate filename*	Specifies the location and file name of the key certificate file.
–d [level [*filename*]	Specifies the level of debug and/or debug file; overwritten at each Sterling Connect:Enterprise Command Line Client startup: 0 = Lowest debug level 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 debug information is displayed on the screen.
–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
–h	Returns the command line syntax.
host_name	Specifies the name of the system running Sterling 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 Sterling Connect:Enterprise Server FTP port listener number; see the *IBM Sterling Connect:Enterprise for UNIX Configuration Files Reference Guide* for a description of the CPD file, which defines the FTP port listener number.
–R [retries/retrywait/]nnnnn-nnnnn	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. retries = Optional. The number of times the system will attempt to reestablish a connection if the original connection fails. retrywait = Optional. The number of seconds between each attempt to establish a connection. nnnnn-nnnnn = A range of port numbers using the format nnnnn-nnnnn. Separate multiple port ranges with commas.
–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 Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client. The subcommands can be entered at the ceftp> prompt. File names with spaces must be enclosed with double quotes (" ").

Note: Sterling Connect:Enterprise Command Line Client 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.
debug [level [*filename*]]	Specifies the level of debug and/or debug file; overwritten at each Sterling Connect:Enterprise Command Line Client startup: 0 = Lowest debug level 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 debug information is displayed on the screen.
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 Sterling Connect:Enterprise Command Line Client command line prompt, you can validate current security settings before you make a secure connection.
ls	Displays a list of documents from the host site.
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.

Subcommand	Description
open	Notifies the remote FTP server with a PORT command.
passive	Notifies the server of a passive mode connection.
portrange [retries/retrywait/]nnnnn-nnnnn	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. retries = Optional. The number of times the system will attempt to reestablish a connection if the original connection fails. retrywait = Optional. The number of seconds between each attempt to establish a connection. nnnnn-nnnnn = A range of port numbers using the format nnnnn-nnnnn. Separate multiple port ranges with commas.
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.
rename	Allows you to rename a remote file or batch. If you are renaming a batch number to a batch name, the batch number must start with a #. Examples: rename oldfile newfile changes oldfile to newfile rename #9999 newfile changes batch number 9999 to newfile
site	Commands that are used to give specific configuration options to the host site and are only good for the ceftp session.
status	Displays the status of the client for the session.
type	Displays the current transfer type: ascii or binary. To change the transfer type, use the ascii or binary subcommand.
user	Sends new user information to the host site.
verbose	Toggles verbose mode (default on).

Additional Examples for Establishing Connections

The following examples show different entries you can make in Sterling Connect:Enterprise Command Line Client to establish secure or unsecure connections.

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

```
#ceftp CEserver 10021
===============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) v1.3.00,
Build25
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
===============================================================================

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

-or-

```
#ceftp
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================
Certificate Passphrase:
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/user01/myconfig
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

ceftp>
```

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

```
#ceftp
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

ceftp> locsite securecfg=/home/user01/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/user01/keycert.txt -t /opt/ceftp/trusted.txt -e s
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

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
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

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
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

Local Site Status:
Configuration File="/home/clcftp/secureftp.cfg"
Client Key-Certificate File="/home/clcftp/keycert.txt"
Server Trusted Root File="/home/clcftp/trusted.txt"
Encryption Strength="strong"
CCC
Security flag="true"
Configuration File="/home/user01/ceftp/secureftp.cfg"
```

Example 7—All connections are unsecure:

```
#ceftp -u
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

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

Example 8—A single connection is unsecure:

```
#ceftp
==============================================================================
IBM(R) Sterling Connect:Enterprise(R) Command Line Client (Secure FTP) 1.3.00
(C) Copyright IBM Corp.  2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
==============================================================================

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

Example 9—Site commands used with Sterling Connect:Enterprise for z/OS:

```
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- 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 *IBM Sterling Connect:Enterprise for z/OS Remote User's Guide*.

Example 10—FTP $$ Commands used with Sterling 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 *IBM Sterling Connect:Enterprise for UNIX Remote User's Guide*.

Microsoft Windows and UNIX Automation Scripts

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

If you want to use the automated scripting capabilities of Sterling Connect:Enterprise Command Line Client, 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 Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client.

Return Codes

The return code from a Sterling Connect:Enterprise Command Line Client invocation can help you determine whether to restart Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client:

Return Code	Category
0	FTP commands were successful.
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.

Return Code	Category
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 Microsoft 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 Sterling Connect:Enterprise Command Line Client with the automation script file.

```
#!/bin/sh
#
retc=0#Set user return value to 0.
#Invoke the Sterling Connect: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 "Sterling Connect: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 Sterling Connect:Enterprise Command Line Client 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 as illustrated in the following:

```
#!/bin/sh
#
retc=0#Set user return value to 0.
#Invoke the Sterling Connect:Enterprise Command Line Client with the Return Code
Checking On (—x).
#
ceftp —a auto_file.txt
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 "Sterling Connect:Enterprise Command Line Client experienced a failure."
else
echo "The Account Log was sent successfully"
fi
exit $retc
```

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

Microsoft Windows Scripting

For return code checking in Microsoft 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 Sterling Connect:Enterprise Command Line Client 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 Microsoft Windows batch file that checks the Sterling Connect:Enterprise Command Line Client return code using the **–x** command line parameter.

```
@echo off
:
:Invoke the Sterling Connect: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 "Sterling Connect:Enterprise Command Line Client experienced a failure."
goto END

:XPASSED
echo "Sterling Connect: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 Microsoft Windows batch file checks the Sterling Connect:Enterprise Command Line Client return code without using the **–x** command line parameter.

```
@echo off
:
:Invoke the Sterling Connect: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 "Sterling Connect:Enterprise Command Line Client experienced a failure."
goto END

:XPASSED
echo "Sterling Connect: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 Sterling Connect:Enterprise Command Line Client.

Chapter 5

Configuring and Using Sterling Connect:Enterprise Command Line Client for FTP

Sterling Connect:Enterprise Command Line Client allows you to connect to FTP servers. Use the information in this chapter to configure Sterling Connect:Enterprise Command Line Client to connect through FTP, connect FTP servers, issue commands, and write automation scripts.

Before You Begin

Before you start Sterling Connect:Enterprise Command Line Client, ensure that the Sterling Connect:Enterprise server has FTP enabled and gather the following information from your host site administrator to access the Sterling Connect:Enterprise server:

- Mailbox ID
- Mailbox password
- IP address or host name of the Sterling Connect:Enterprise server
- FTP listening port number
- Network path and firewall navigation information

Setting Port Range Limits

Setting port range limits enables you to restrict the TCP/IP ports used for FTP transactions between Sterling Connect:Enterprise Command Line Client and Sterling 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 Sterling 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 Sterling Connect:Enterprise for UNIX version 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 *Issuing Commands* on page 51, 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
```

Connecting to an FTP Server

To establish a connection that does not use SSL or SSH 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 *Issuing Commands* on page 51.

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 Sterling 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 Sterling Connect:Enterprise Command Line Client. 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>
  ```

Issuing Commands

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

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

Parameter	**Description**
–a *automation script filename*	Specifies the location and file name of the automation script file; see *Microsoft Windows and UNIX Automation Scripts* on page 54, for more information.
–d [level [*filename*]]	Specifies the level of debug and/or debug file; overwritten at each Sterling Connect:Enterprise Command Line Client startup: 0 = Lowest debug level 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 debug information is displayed on the screen.
–h	Returns the command line syntax.
host_name	Specifies the name of the system running Sterling 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).

Parameter	Description
port_number	Specifies the Sterling Connect:Enterprise Server FTP port listener number; see the *Sterling Connect:Enterprise for UNIX Configuration Files Reference Guide* for a description of the CPD file, which defines the FTP port listener number.
–R [retries/retrywait/]nnnnn-nnnnn	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. retries = Optional. The number of times the system will attempt to reestablish a connection if the original connection fails. retrywait = Optional. The number of seconds between each attempt to establish a connection. nnnnn-nnnnn = A range of port numbers using the format nnnnn-nnnnn. Separate multiple port ranges with commas.
–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.
–u	Specifies to Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client. The subcommands can be entered at the ceftp> prompt. File names with spaces must be enclosed with double quotes (" ").

Note: Sterling Connect:Enterprise Command Line Client supports only the subcommands listed in the following table.

Subcommand	Description
!command [parameters]	Sends the command to the operating system.
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.
debug [level [*filename*]]	Specifies the level of debug and/or debug file; overwritten at each Sterling Connect:Enterprise Command Line Client startup: 0 = Lowest debug level. 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 debug information is displayed on the screen.

Subcommand	Description
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 Sterling Connect:Enterprise Command Line Client command line prompt, you can validate current security settings before you make a secure connection.
ls	Displays a list of documents from the host site.
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.
passive	Notifies the server of a passive mode connection.
portrange [retries/retrywait/]nnnnn-nnnnn	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. retries = Optional. The number of times the system will attempt to reestablish a connection if the original connection fails. retrywait = Optional. The number of seconds between each attempt to establish a connection. nnnnn-nnnnn = A range of port numbers using the format nnnnn-nnnnn. Separate multiple port ranges with commas.
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.
rename	Allows you to rename a remote file or batch. If you are renaming a batch number to a batch name, the batch number must start with a #. Examples: rename oldfile newfile changes oldfile to newfile rename #9999 newfile changes batch number 9999 to newfile
site	Commands that are used to give specific configuration options to the host site and are only good for the ceftp session.
status	Displays the status of the client for the session.

Subcommand	Description
type	Displays the current transfer type: ascii or binary. To change the transfer type, use the ascii or binary subcommand.
user	Sends new user information to the host site.
verbose	Toggles verbose mode (default on).

Microsoft Windows and UNIX Automation Scripts

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

If you want to use the automated scripting capabilities of Sterling Connect:Enterprise Command Line Client, you must create an automated script file that contains subcommands.

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 Sterling Connect:Enterprise Command Line Client.

Return Codes

The return code from a Sterling Connect:Enterprise Command Line Client invocation can help you determine whether to restart Sterling Connect:Enterprise Command Line Client 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 Sterling Connect:Enterprise Command Line Client:

Return Code	Category
0	FTP commands were successful.
1	Session establishment failure occurred.
2	Authentication or login failure occurred.
3	Client subcommand (none-copy) failure occurred.
4	Subcommand put(STOR) failure occurred.

Return Code	Category
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 Microsoft 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 Sterling Connect:Enterprise Command Line Client with the automation script file.

```
#!/bin/sh
#
retc=0#Set user return value to 0.
#Invoke the Sterling Connect: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 "Sterling Connect: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 Sterling Connect:Enterprise Command Line Client 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 as illustrated in the following example:

```
#!/bin/sh
#
retc=0#Set user return value to 0.
#Invoke the Sterling Connect:Enterprise Command Line Client with the Return Code
Checking On (—x).
#
ceftp —a auto_file.txt
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 "Sterling Connect:Enterprise Command Line Client experienced a failure."
else
echo "The Account Log was sent successfully"
fi
exit $retc
```

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

Microsoft Windows Scripting

For return code checking in Microsoft 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 Sterling Connect:Enterprise Command Line Client 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 Microsoft Windows batch file that checks the Sterling Connect:Enterprise Command Line Client return code using the **–x** command line parameter.

```
@echo off
:
:Invoke the Sterling Connect: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 "Sterling Connect:Enterprise Command Line Client experience a failure."
goto END

:XPASSED
echo "Sterling Connect: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 Sterling Connect:Enterprise Command Line Client return code without using the **–x** command line parameter.

```
@echo off
:
:Invoke the Sterling Connect: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 "Sterling Connect:Enterprise Command Line Client experience a failure."
goto END

:XPASSED
echo "Sterling Connect: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 Sterling Connect:Enterprise Command Line Client.

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.

SSH

Secure Shell (SSH) is a method of TCP/IP secure communications between a client and a host computer on a network. Security features included end-to-end encryption, password and public key authentication, and data integrity. Sterling Connect:Enterprise Command Line Client supports SSH-2 SFTP protocol.

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

Notices

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

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

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

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual

Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA__95141-1003

U.S.A.

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

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

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

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

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

This information is for planning purposes only. The information herein is subject to change before the products described become available. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are ficticious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

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

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

Trademarks

IBM, the IBM logo, and 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 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.