Gentran:Director®
for Windows®

Version 5.4

FTP Option

User's Guide

Sterling Commerce
An IBM Company

Copyright Notice

April 2005

Contents

Preface

The *Gentran:Director – FTP Option User Guide* is for staff who install and maintain the Gentran:Director 5.4 product.

This guide assumes knowledge of the Windows operating system.

Task Overview

The following table lists the information required to perform FTP Option tasks:

Task	Reference
Installing the FTP Option scripts	Chapter 1, *Installing the FTP Option Scripts*
Configuring unsecure connections, connecting to remote hosts, issuing commands, and using automation scripts.	Chapter 2, *Configuring and Using the FTP Option*
Integrating the FTP Option into Gentran:Director	Chapter 3, *Integrating the FTP Option into Gentran:Director*

Getting Support for Sterling Commerce Products

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

The Sterling Commerce Customer Support Web site at www.sterlingcommerce.com is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information.

In addition, you can also reach Customer Support at 1-800-GENTRAN (option 7) or by e-mail at pcbasedsupport@stercomm.com.

Installing the FTP Option Scripts

This chapter describes installing the FTP Option scripts from the Comm Scripts CD-ROM or ESD Portal download on a computer running Gentran:Director.

Installing the FTP Option Scripts

To install the FTP Option communications scripts, see "Installing the FTP Option Communication Scripts" in Chapter 1 of the *Gentran:Director Getting Started Guide.*

Configuring and Using the FTP Option

The FTP Option allows you to connect to FTP servers. Use the information in this chapter to configure the FTP Option to connect to an FTP server, issue commands, and write automation scripts.

Before You Begin

Before you start using the FTP Option, gather the following information from your host site administrator to access the FTP server:

✦ IP address or host name

✦ 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, 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 sessions.

> **Note:** Because these ports must also be available at the server end of the connection, you need to coordinate with your FTP server's personnel.

1. Open an FTP Option command line.

---

2.   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.	Valid values: 0 to 99 The default is 0.
retrywait	Optional. The number of seconds between each attempt to establish a connection.	Valid 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 8 for command line parameter definitions and usage.

The following sample command line specifies two port ranges, the first from 40,000 to 50,000, inclusive, and the second from 55,000 to 60,000, inclusive. If the original connection attempt fails, there will be one retry with a delay of 90 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 an FTP connection, you must use one of the following:

✦ The unsecure command line parameter (*-u*) makes the entire session unsecure.

✦ The *locsite unsecure* subcommand makes one connection unsecure.

For more information about the *locsite* subcommand, see the *locsite* entry in Issuing Commands on page 8.

Establish a connection in one of the following ways:

✦ At the command line prompt, type **gdftp**, the host name and port number of the FTP server, and the **-u** parameter, as shown in the following example. Press **Enter**:

```
$gdftp host_name port_number -u
```

The following prompt is displayed:

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

✦ At the command line prompt, type **gdftp** to start the FTP Option. At the gdftp prompt, type the **locsite** unsecure subcommand, as in the following example:

```
gdftp>locsite unsecure
```

The following prompt is displayed:

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

Issuing Commands

After you establish a connection, you can use the supported command line parameters and subcommands.

The following command line parameters are supported by the FTP Option. 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 Automation Scripts on page 12 for more information.
`-d [level [filename]]`	Specifies the level of debug and/or the debug file; overwritten at each FTP Option 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 FTP server; you can enter the IP address of the host instead of the host name.
`-i`	Turns off interactive prompting during multiple document transfers. The default is to prompt.
`port_number`	Specifies 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.

Parameter	Description
`-s configuration filename`	Specifies the location and file name of the client configuration file, which is a user-defined configuration file.
`-u`	Specifies to the FTP Option 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. The default is on.
`-x`	Turns on result code exiting for the entire instance of the client.

Supported Subcommands

The following standard FTP syntax subcommands are supported by the FTP Option. The subcommands can be entered at the `gdftp>` prompt. File names with spaces must be enclosed with double quotes (" ").

Note: The FTP Option 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 FTP Option 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 as deleted.

Subcommand	Description
dir	Requests a formatted listing of documents from the host site.
get	Requests a formatted document 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 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 as deleted.
mget	Receives multiple documents from the host site.
mput	Sends multiple documents to the host site.
open	Notifies the remote FTP server with a PORT command.
passive	Notifies the server of a passive mode connection.

Subcommand	Description
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 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 gdftp 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).

Automation Scripts

> **Note:** Gentran:Director uses FTP scripts, unattended session files, and batch files to integrate the Gentran:Director FTP Option.

The FTP Option provides automated scripting capabilities for file exchanges during unsecure FTP connections. This scripting capability eliminates the need for you to run the FTP Option manually. This feature works on any platform that supports Java.

If you want to use the automated scripting capabilities of the FTP Option, 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:

```
$gdftp -a uauto_sc_file -u
```

> **Note:** The subcommands in the *uauto_sc_file* are standard FTP syntax commands supported by the FTP Option.

Return Codes

The return code from an FTP Option invocation can help you determine whether to restart the FTP Option 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:

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.

Return Code	Category
4	Subcommand put(STOR) failure occurred.
5	Subcommand get(RETR) failure occurred.
6	SSL configuration file parameter failure occurred.
7	Command line parameter failure occurred.
8	**Locsite** command parameter failure occurred.
9	Reserved.
10	Catastrophic failure occurred.

You can perform the return code checks in two ways:

✦ Using the $-x$ command line parameter

✦ Typing the @ symbol next to the subcommand for which you would like the return code checked

The $-x$ command line parameter checks return codes for all commands. The @ symbol only checks return codes for the subcommand (s) with which it is associated.

Writing FTP Scripts

For return code checking, you must create an automation script file and a batch file. The batch file must contain the FTP Option subcommands.

The batch file 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 batch file that checks the return code using the *-x* command line parameter.

```
@echo off
:
:Invoke the FTP Option with the Return Code Checking On (-x).
:
CALL gdftp -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 "FTP Option experience a failure."
goto END

:XPASSED
echo "FTP Option subcommand was successful"
goto END

:END
```

In the preceding example, the *CALL gdftp* 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 batch file checks the return code without using the *-x* command line parameter.

```
@echo off
:
:Invoke the FTP Option with the Return Code Checking On.
:
CALL gdftp -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 "FTP Option experience a failure."
goto END

:XPASSED
echo "FTP Option subcommand was successful"
goto END

:END
```

For the preceding example, the *CALL gdftp* 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 the FTP Option.

Integrating the FTP Option into Gentran:Director

This chapter describes integrating the FTP Option into Gentran:Director.

Setting up Your Scripts

Three types of files are installed from the Comm Scripts CD-ROM (or ESD Portal download):

✦ FTP scripts, which are the scripts for communicating with the VANs

✦ Batch files, which invoke the FTP Option

✦ Unattended session files

FTP Scripts

The FTP scripts are installed to the COMMERCE Connection32\bin\Commscr\GENFTP directory. The files are named *<network name>***FTPSnd.txt** and *<network name>***FTPRcv.txt** (where *<network name>* is the name of the VAN).

These files contain variables that you need to modify before you can use them.

1. Use a text editor to open the files you will be using.

2. Search for the variables listed below.

Variable	Description
## IP Address ## (space before and after "IP Address")	Your IP address

Variable	Description
`## ID ##` (space before and after "ID")	Your user ID
`## Password ##` (space before and after "password")	Your user password
`## homedir ##` (space before and after "homedir")	The directory where Gentran:Director is installed **Note:** The default directory is C:\Program Files\COMMERCE Connection32.

3.   Replace the variables (from the first # to the last #) with your information.

4.   Save and close the file.

Batch Files

The batch files are installed to the COMMERCE Connection32\FTP directory. The files are named **backup.bat** and **delete.bat**.

> **Note:**   If you installed scripts for more than VAN, you will need to make copies of the .bat files and rename them.

These files contain a variable that you need to modify before you can use them.

1.   Use a text editor to open the files you will be using.

2.   Search for the variable listed below.

Variable	Description
`##VAN##` (no spaces)	The name of the VAN you are connecting to and installed the scripts for

3.   Replace the variable (from the first # to the last #) with your information.

4.   Save and close the file.

Session Files

The session files are installed to the COMMERCE Connection32\Unattend directory. The files are named *<VAN>***FTP.ses** (where *<VAN>* is the name of the VAN).

These files contain a variable that you need to modify before you can use them.

1.   Use a text editor to open the files you will be using.

2. Search for the variable listed below.

Variable	Description
## homedir ## (space before and after "homedir")	The directory where Gentran:Director is installed **Note:** The default directory is C:\Program Files\COMMERCE Connection32.

3. Replace the variable (from the first # to the last #) with your information.

4. Save and close the file.

Setting up Send/Receive

Before setting up Send/Receive, you must have updated the scripts (as described above) and set up a comm profile to point to the FTP folder (for **sendfile.edi**).

> **Note:** The session file looks in the Out Box for data when trying to send. If there is nothing in the Out Box, it will perform a receive only.
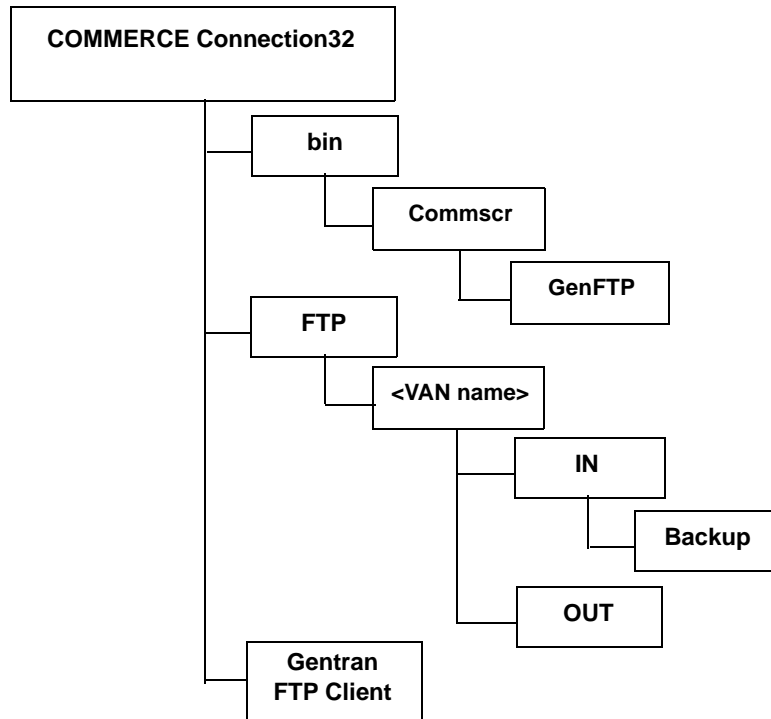
Use the Gentran:Director process control to create an event. Then, schedule, execute, or activate the event. See Chapter 2 in the *Gentran:Director Administrator's Guide* for details.

Directory Structure

The following illustrates the FTP Option directory structure within COMMERCE Connection32:

```
COMMERCE Connection32
    ├── bin
    │      └── Commscr
    │             └── GenFTP
    ├── FTP
    │      └── <VAN name>
    │                ├── IN
    │                │     └── Backup
    │                └── OUT
    └── Gentran
        FTP Client
```

Data Flow Overview

The following describes the FTP Option data flow:

Stage	Description
1	The GDW_SendReceive performs a data interchange.
2	The data interchange runs the file copy comm profile and extracts outgoing data to the FTP folder.
3	The unattended script moves the data to the appropriate VAN OUT folder.
4	The unattended script invokes the appropriate batch file to send the data to the VAN.
5	The unattended script invokes a receive to place the incoming data into the appropriate VAN IN folder.

Stage	Description
6	The unattended script copies the data into the Backup folder.
7	The GDW_Process_File processes data into Gentran:Director.

Periodically, clean out old files from the **COMMERCE Connection32\FTP\<VAN name>\In\Backup** folder.

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 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 (for example, your place of work or your telephone contact details). 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. 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 word or phrase that establishes identity to allow access to a system.

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 the 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 is often used during the period between your request and receipt of a certificate from a public certificate authority. If self-signed certificates are used, the trusted root signing certificate must be installed in the client manually.

Session Key

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

T

Third-party Certificate

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

Trusted Root Certificate File

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

U

Unsecure Connection

An FTP connection that has no security.

X

X.509 Certificate

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

Index

A

Automation Scripts 12

B

Batch file sample
  checking return code with -x 14
  checking return code without -x 15

Batch files 18

Batch scripting 13

C

Command parameters 8

Connecting to an FTP server 7

Customer Support 1

D

Data flow 20

F

FTP scripts 17

FTP servers 7

I

Issuing Commands 8

P

Parameters
  FTP Option commands 8
  TCP/IP ports 6

R

Return codes 12

S

Sample batch file
  checking return code with -x 14
  checking return code without -x 15

Sample commands
  TCP/IP ports 6

Scripts
  directory structure 20
  setting up 17
  supplied by Sterling Commerce 17
  writing 13

Session files 18

Setting port information 5

Setting up Send/Receive 19

Setting up your comm scripts 17

Subcommands 9

Support 1

T

TCP/IP ports 5

W

Writing scripts 13