

# Programmable Batch Interface for FTP and FTP2

---

This chapter describes the programmable batch interface for FTP and FTP2. It includes these sections:

- **What Is the Programmable Batch Interface?**  
Describes how to use a programmable batch interface to FTP/FTP2.
- **FTP/BATCH Interfaces**  
Describes the FTP/BATCH module and the control fields.
- **Coding The Batch Application**  
Describes how to code the application.
- **Execution Samples**  
Describes batch application examples using FTP and FTP2.
- **Debugging Information**  
Describes information useful for debugging application programs.
- **Sample Batch Programs**  
Describes and lists several sample batch programs.

## What Is the Programmable Batch Interface?

Using FTP/FTP2 in batch mode automates the process of transferring files, which is particularly useful if your site performs FTPs to/from particular servers repeatedly. Executing the FTP/FTP2 operations via a batch program can eliminate mistakes caused by typing errors and users only need to know how to start the batch program and don't need to learn or understand the FTP/FTP2 transfer commands.

For example, if you need to transfer a specific file from one host to another on a regular basis a batch program can be written to execute the commands automatically instead of having to repeat the same FTP/FTP2 commands from a terminal each time the file transfer is necessary.

Although both FTP and FTP2 can be invoked in batch mode existing batch methods cannot do much to respond to error conditions that may occur during a file transfer. A programmable batch interface has been added to Cisco IOS for S/390 that lets you write a program using a language of your choice (as long as the language supports an assembler language CALL interface, such as C, PL/I, COBOL, etc.) to issue FTP and FTP2 commands and to respond to conditions that may develop during the course of those FTP/FTP2 sessions.

- The batch program you write will run in a batch region on an MVS host.
- Your application will simulate input from a user's terminal.
- Multiple batch routines can be executed concurrently.
- You are limited only by the number of LUs defined to Cisco IOS for S/390. In general, each FTP/FTP2 user ("user" includes a batch application) requires a minimum of two LUs for execution.

For details on invoking FTP and FTP2, and FTP3, read Invoking Client FTP2 in Client FTP2, Invoking Client FTP3 in Client FTP3, and Invoking Client FTP in Client FTP.

## FTPBatch Interfaces

The FTPBatch module within Cisco IOS for S/390 is used as an interface between your batch program and Cisco IOS for S/390, FTP, and FTP2 programs.

---

**Note** Cisco IOS for S/390 has an alias of HB0007 for backwards compatibility. Batch programs using previous versions of this module will have to assemble/link the new module for successful execution.

---

When writing your batch program, you must set register R1 to point to the address of a parameter list. The first 4 bytes of the parameter list must point to an FTPBatch batch control area and the last 4 bytes must point to an FTPBatch data control area.

The batch control area is a 10-field control block that stores the Cisco IOS for S/390 commands you want executed. It also holds the responses that are returned from the Cisco IOS for S/390 application.

The data control area consists of a single 79-character field. This area is where your batch program must write the FTP/FTP2 commands you would ordinarily enter on a terminal while interactively operating FTP/FTP2 (for example, OPEN, GET, PUT, or APPEND).

## Batch Control Fields

This section describes each field in the batch control area.

### Command Field

The command field indicates the batch command that Cisco IOS for S/390 will execute.

---

Table 11-1 shows the alphabetic characters that are valid values for this field:

**Table 11-1 Valid Values for the Batch Command Field**

<b>Value</b>	<b>Description</b>
<b>O</b>	<b>OPEN</b> command  Establishes (opens) a connection between the batch application and an FTP or FTP2 application. Which application (FTP/FTP2) you establish a connection with is determined by what is specified on the SYSOPT input record in the startup JCL.
<b>S</b>	<b>SEND</b> command  Indicates you are sending a command to the application (either FTP or FTP2). The actual command to be sent must be stored in the 79-character data control area.
<b>R</b>	<b>RECEIVE</b> command  Receives a response from the FTP/FTP2 application. The response received will be the response from the previous command sent (via the <b>SEND</b> command). You are not required to issue a <b>RECEIVE</b> request for every message generated by the previous <b>SEND</b> . All response messages not retrieved via a <b>RECEIVE</b> will be ignored. (This allows the use of a batch program that only executes <b>SEND</b> requests.) All commands and responses will appear in the SYSPUT log file.
<b>C</b>	<b>CLOSE</b> command  Closes the connection between your batch application and the FTP/FTP2 application.

Table 11-2 describes how to terminate FTP and FTP2:

**Table 11-2 Terminating FTP and FTP2**

<b>Reason For Termination</b>	<b>FTP</b>	<b>FTP2</b>
User program issues a <b>CLOSE</b> .	yes	yes
User program issues a <b>SEND</b> request that executes an <b>END</b> command.	yes	yes
User program issues a <b>SEND</b> request that executes a <b>BYE</b> or a <b>QUIT</b> .	no	yes

Although a **CLOSE** command is not always required after a **BYE**, **END**, or **QUIT** command, it is recommended that you issue a **CLOSE** request before exiting your program. If a user program **CLOSE** request is issued after a **BYE**, **END**, or **QUIT**, a return code of 16 (Close error - not open) will be returned to your program.

## Length

The length field identifies the number of characters allowed as input from the data control area. Valid selections are 1 to 79. Your program must pad this field with spaces (X'40') before writing to it, to clean out remnants of previous commands.

## Return Code

The return code field contains the return code received from Cisco IOS for S/390 indicating if a command was returned in error.

Table 11-3 lists the possible return codes:

**Table 11-3 Return Codes**

Return Code	Description
0	Command completed successfully
3	Invalid command received on input
4	Invalid message length specified
5	Application no longer active
7	Invalid parameter input
8	Send command execution failed. Check SYSPUT output. See the note that follows this table
14	Logic error - next command must be open or close
15	Open error - already open
16	Close error - not open
17	Send error - not open
18	Receive error - not open

---

**Note** When a user program issues a SEND request to execute an FTP command, there may be several response messages associated with a single SEND command. A return code of 08 will be returned to the user program in two ways:

---

- When control is returned to the user program following a user SEND request. This indicates the command did not complete successfully.
- When control is returned to the user program following a user RECEIVE request. Normally the 08 return code will be set on the message that FTP/FTP2 encountered the error. But in some cases (due to processing logic), the 08 return code will be returned to the user on a following message.

### Control1

The control flag set by Cisco IOS for S/390 in this field indicates the relative position of the data block in a data stream. This flag is set to one of the values shown in Table 11-4:

**Table 11-4 Control1 Flags**

Flag	Description
<b>F</b>	Indicates the first block in the data stream
<b>M</b>	Indicates the middle block in the data stream
<b>L</b>	Indicates the last block in the data stream

### Control2

This field contains a control flag set by Cisco IOS for S/390 that indicates the state of session activities. This flag will be set to one of the values shown in Table 11-5:

---

**Table 11-5 Control2 Flags**

<b>Flag</b>	<b>Description</b>
<b>S</b>	Indicates a session with Cisco IOS for S/390 is being started
<b>R</b>	Indicates the RESPONSE field contains a reply to a command (O/S/R/C)
<b>Q</b>	Indicates the session is closing (quit)

When FTP or FTP2 terminates with a BYE, END, or QUIT command, this field is set to Q when control is passed back to your program.

### Secondary LU (Send)

This field is not used.

### Secondary LU (Receive)

This field is not used.

### Response

This field contains textual data returned from Cisco IOS for S/390 FTP or FTP2 application. This area must be defined as a 132-character area. It contains an image of a buffer sent to the SYSPUT output file upon returning to the user program after a RECEIVE request.

### Data Control Field

When FTP/FTP2 commands are issued from a user's terminal, all data is on one line (the screen has only one line from which to enter single line commands). Therefore, when FTP/FTP2 commands are issued by a batch application, the data is restricted to one line containing up to 79 characters. The FTP/FTP2 commands you want issued must be written into the data control field.

## Coding The Batch Application

Batch applications can be coded in any language capable of making assembly language calls. This includes languages such as COBOL, C, and PL/I. Your batch application must submit FTP/FTP2 commands according to the specifications described in Batch Invocation in Client FTP2. Your batch application calls the Cisco IOS for S/390 FTPBATCH module, which passes the information to Cisco IOS for S/390 as if it came from the user's terminal. Your application should also check the error return codes after issuing each batch command. (Read FTPBATCH Interfaces.)

After your batch application is written, it must be compiled, cataloged, and link-edited. The Cisco IOS for S/390 load library must be included with the link-edit job as module FTPBATCH is located in this library.

Your batch application (user program) issues calls to the FTPBATCH module which, in turn, gets input (in other words, the run time options) from the SYSOPT data set and passes that input to the FTP/FTP2 programs. Output from the FTP/FTP2 commands is directed to the SYSPUT file. Cisco IOS for S/390 controls the FTP/FTP2 programs, although this occurs transparently to your application.

## Authorized Program Facility (APF)

If your batch program is to communicate with the FTP2 application, your program must reside in an authorized program library. To authorize a new LOAD file, you must add a new statement to the IEAAPFxx member in your SYS1.PARMLIB. After adding a new LOAD file to IEAAPFxx, the system must be IPLed to enable the new authorization. Also, your program must be link edited with the authorization parameter of SETCODE AC(1).

---

**Note** Batch programs that work with the FTP application are not required to be APF authorized.

---

## Application Selection (SYSOPT) and Invocation Options

User batch programs must select either the FTP or FTP2 application for processing commands. The selection is made by coding FTP or FTP2 beginning in column 1 of the SYSOPT input record (see the execution examples below).

The SYSOPT input record can be used to pass invocation options to the FTP or FTP2 application. Following the FTP or FTP2 selection characters, include a slash (/) followed by the options you want.

### Examples

- Sample invocation with options:  
**FTP2 /TEST (this syntax passes the TEST option to FTP2)**
- Sample invocation without options:  
**FTP2**

You can also enter PL/I options along with FTP/FTP2 options. The PL/I options must be coded following the FTP or FTP2 selection characters, but before the separator (/).

### Examples

The following are PL/I invocation examples:

**FTP2 NOSPIE NOSTAE /TEST**

(Note that the TEST option follows the PL/I options.)

**FTP2 NOSPIE NOSTAE**

---

**Note** The FIOS option can not be used with the programmable batch interface.

---

## User Program Batch Invocation

Your user program can run in batch like any other batch program, or as a TSO command processor by running under a batch Terminal Monitor Program (TMP). FTP2 commands DO, DIR, and LS, and the FTP DO command require a TSO environment to execute, thus they will only execute in batch under the TMP.

With batch, it is important to specify the FTP commands carefully, because the slightest error might cause all subsequent commands to fail and force you to rerun the batch job. Your program should test the return code after each request to ensure that correct processing is achieved.

---

## Execution Samples

These examples are provided to help you code your applications.

### Sample Using FTP

Your batch application can be invoked like any other batch program. This sample shows how to invoke your program to work with FTP.

```
//jobname      JOB      job-stmt-parms
//FTPSTEP      EXEC     PGM=user-prog
//STEPLIB      DD       DISP=SHR,DSN=user.linklib
//              DD       DSN=SNSTCP.Vxxx.LINK, /*IF FTP NOT LINKLISTED*/
//              DD       DISP=SHR
//SYSPUT       DD       SYSOUT=*,DCB=BLKSIZE=133
//SYSUDUMP     DD       SYSOUT=*
//SYSOPT      DD       *,DCB=BLKSIZE=80
FTP / option1 option2 ...
/*
```

### Sample Using FTP2

You can set up your FTP2 application for either batch or TMP execution.

#### Batch Program

The following is an example of how to set up your FTP2 application for batch execution.

```
//jobname      JOB      job-stmt-parms
//FTPSTEP      EXEC     PGM=user-prog
//STEPLIB      DD       DISP=SHR,DSN=user.linklib
//              DD       DSN=SNSTCP.Vxxx.LINK, /*IF FTP2 NOT LINKLISTED*/
//              DD       DISP=SHR
//SYSPUT       DD       SYSOUT=*,DCB=BLKSIZE=133
//SYSUDUMP     DD       SYSOUT=*
//NETRC        DD       DISP=SHR,DSN=userid.FTP.NETRC
//SYSOPT      DD       *,DCB=BLKSIZE=80
FTP2 / option1 option2 ...
/*
```

---

**Note** When you invoke FTP or FTP2 via a batch program, the TSO environment required for the **DO**, **DIR**, and **LS** commands is not available. Consequently, these commands will be rejected when FTP or FTP2 executes via a batch program. The **NETRC** statement is required if the automatic login feature is being used.

---

### TMP Execution

Your user program can be invoked in batch as a TSO command processor by running it under a batch TMP. This mode of operation allows your program to use **DO**, **DIR**, and **LS** commands (which are not supported in ordinary batch mode). TMP execution also causes output from TSO commands to be directed to the **SYSTSPRT** file; they are not returned to your batch program.

The following example shows how to initiate your batch program under TMP:

```
//jobname      JOB      job-statement-parameters
//TPMSTEP      EXEC     PGM=IKJEFT01
//STEPLIB      DD      DISP=SHR,DSN=user.load.lib
//             DD      DSN=SNSTCP.Vxxx.LINK, /*IF NOT LINKLISTED*/
//             DD      DISP=SHR
//SYSPRINT     DD      SYSOUT=*
//SYSPUT       DD      SYSOUT=*,DCB=BLKSIZE=133
//SYSTSPRT     DD      SYSOUT=*
//SYSUDUMP     DD      SYSOUT=*
//SYSTSIN      DD      *
user_program_name
/*
//SYSOPT       DD      *
[FTP | FTP2] / option1 option2 ...
/*
```

## Debugging Information

This section describes information to help you debug your application programs.

### Environment

The environment used to execute the programmable batch facility consists of these three parts:

- Your batch application
- The FTPBATCH interface module
- The Cisco IOS for S/390 FTP or FTP2 application

These three parts all reside within the same MVS address space during execution. Therefore, be aware that program errors within your application can overlay code within the FTPBATCH or FTP/FTP2 application causing unpredictable results. Ensure that your applications do not conflict with the FTPBATCH or FTP/FTP2 application processing.

### PL/I Environment

The FTP and FTP2 applications are coded in the PL/I language. The environment setup when PL/I is used can cause some problems when trying to debug other modules of a different language that may be running within the same address space. To help debugging, use PL/I execution parameters to control PL/I debug options. PL/I options can be submitted by entering them on the SYSOPT input record.

#### Example

```
FTP NOSPIE NOSTAE / TEST
```



---

## FTP and FTP2 Debugging Options

Both the FTP and FTP2 applications have debugging options that let you obtain debugging information on the internal operation of the user FTP or FTP2 program and the interactions between the User FTP program and the Server FTPs. When using debugging options, trace information generated by the option will be sent to both the SYSPUT file and to your batch program.

## Message Output

Output messages created from the programmable batch facility can be found at several separate locations within the jobs output stream. The SYSGET output file will contain FTP or FTP2 transfer and trace related information. WTO messages are on the MVS console log and can be directed to the user's JCL message log file. Depending on the language used to write your batch application, the messages output by the language being used may appear in a separate area of your output. In general, be aware that the complete job output must be reviewed to assist you if problems occur.

## Sample Batch Programs

This section contains sample batch applications to illustrate the requirements for coding batch applications. The samples provided are for reference only and may not contain sufficient details for an actual Cisco IOS for S/390 environment.

## COBOL Language Batch Application

See member FTPCOBOL in the SAMP library for a sample of a COBOL language batch application.

## Assembler Language Batch Application

See member FTPASM in the SAMP library for a sample of an Assembler language batch application.

## Sample Batch Application Linkedit

```
//LINK1 DD DISP=SHR,DSN=SNSTCP.Vxxx.LINK
//LINK2 DD DISP=SHR,DSN=user.linklib
//SYSLIN DD *
        ENTRY user_prog
        INCLUDE LINK1(FTPBTACH)
        INCLUDE LINK2(user_prog)
        NAME user_prog(R)
/*
```

## Sample Batch Application SYSPUT Output

```
14:05:00 Cisco IOS for S/390 User FTP - Enter command or '?'
14:05:00 USERFTP: a:conn 123.456.7.8
14:05:09 A: UNIX.COMPANY.COM -- FTP Server, Enter command or HELP
14:05:09 USERFTP: a:log wv85 #####
14:05:11 USERFTP: b:conn 234.567.8.9
14:05:13 B:220 unix FTP server (SunOS 4.1) ready.
14:05:13 USERFTP: b:log unix #####
14:05:14 USERFTP: a:get filea 'TEST.LIB(MEMBER) '
14:05:14 HOST PREFIX IGNORED FOR GET
14:05:16 B:150 ASCII data connection for filea (234.567.8.9,4175) (428 bytes).
14:05:16 B:226 ASCII Transfer complete.
14:06:08 A:150-Dataset open with attributes:
14:06:08 A:Type A N      Tabs      8      Stru F      Mode S      Recall 5
14:06:08 A:Path TEST.LIB(MEMBER)
14:06:08 A:Volser REST00      Unit SYSDA      Dsorg      PO      Recfm FB      Lrecl      80
14:06:08 A:Blksize      3120      Space 10 0 Tracks      Rlse      Dir      5
14:06:08 A:150
14:06:08 A:226-Transfer complete
14:06:08 A:439 bytes received in 0.85 seconds (516 bytes/s)
14:06:08 A:Path TEST.LIB(MEMBER) User TEST Data bytes received 417
14:06:08 A:Disk tracks written 1      Records padded      11
14:06:08 A:226
14:06:09 USERFTP: a:quit
14:06:10 A:221 Session terminated
14:06:11 USERFTP: b:quit
14:06:11 B:221 Goodbye.
14:06:12 USERFTP: END
```