The z/OS Problem Documentation Upload Utility

The IBM® z/OS® Problem Documentation Upload Utility is a parallel File Transfer Protocol (FTP) utility that is designed to send documentation in a more efficient manner to IBM FTP sites. This utility sections the input file into smaller files that are sent in parallel resulting in shorter transmission time for very large data sets (such as stand-alone dump). You can optionally encrypt the data sets.

There are two work files for each parallel FTP session (the "A" file and the "B" file). Each "A" work file is filled by copying records from the input file. When the "A" file is full, the FTP sessions are started in parallel. At the same time, each "B" work file is filled by copying records from the input file. When the "B" file is full and the transfer of the "A" file is complete, transfer of the next "B" file starts. This process continues between the "A" and the "B" files, until everything in the input file is sent.

You can have up to 20 parallel FTP sessions running simultaneously. The work data sets are dynamically allocated and can range in size from 1 MB to 9,999 MB. You can experiment to see what works best in your environment, but here are some guidelines:

- Start with three or four parallel FTP sessions. Too many parallel FTP sessions can saturate the network link.
- Use medium size work data sets.

If the work data sets are very small in relationship to the input data set, you can end up with too many files on the IBM FTP sites. For example, if you are sending a 100 GB z/OS stand alone dump and make the work data set size 1 MB, you create 100,000 files on the IBM FTP site, which exceeds the IBM limit of 999 files. This also causes a lot of delay by starting and stopping the FTP sessions for each file.

If the work data sets are very large in relationship to the input file size, the amount of overlap time is decreased. When the program first starts, it must fill the "A" work files before it starts transmitting any data, which means the copy time is not overlapping with data that needs to be sent through FTP. For example, if you were sending a 1 GB dump and you set the work data set size to 1 GB (1,000 MB), there is no overlap between copying the records and sending the work files.

The parallel FTP program always compresses the input data before it is written to the work data sets, therefore it is not necessary to use a tool such as AMATERSE or TRSMAIN to compress the input data set before using the parallel FTP program to send it to the IBM FTP site. In addition, 192-bit triple Data Encryption Standard (DES) can be requested by using the **CIPHER_KEY** keyword. Without the keyword, the data is just compressed. With the keyword, the data is compressed, and then encrypted.

Encryption is provided by the CP Assist for Cryptographic Functions (CPACF), DES/TDES Enablement (feature 3863) and is available on all processors starting with the z990 (2084) and z890 (2086). CPACF, feature 3863 enables clear key DES and TDES instructions on all supported CPs. The encryption algorithm used for 3DES is documented in *z/Architecture Principles of Operation*, SA22-7832 (www.ibm.com/systems/z/os/zos/bkserv/r10pdf/#zarchpops).

JCL statements for z/OS Problem Documentation Upload Utility

The JCL statements for the z/OS Problem Documentation Upload Utility are:

SYSPRINT

The data set can be either SYSOUT or a sequential data set. The data set must be RECFM=FB, LRECL=134.

SYSUT1

The sequential data set to transfer to IBM. The record formats currently supported are fixed, fixed blocked, variable, and variable blocked. For example, the dump data set that you are sending to IBM. The input has to be either a sequential data set, a member of a PDS, or a member of a PDSE.

SYSIN

A sequential data set that uses the following control statements. The data set must be RECFM=FB, LRECL=80.

The parallel FTP utility is managed through the following JCL control statements with these guidelines:

- Use an asterisk (*) in the first column to indicate comments.
- Keywords must start in column one.
- Use control statements that are in form VERB=OPERAND.
- Mixed case verbs and operands are allowed.
- The operand starts in the column after the equal sign and goes to the first blank column except `TARGET_SYS`, `DIRECTORY`, `CIPHER_KEY`, `ACCOUNT`, `USERID`, and `PASSWORD`, which can contain blanks.
- Anything after the first blank is ignored except for any operands that can contain blanks. In those cases, do not use blanks from column one to the end of the operand.

TARGET_SYS

The name of the TCP/IP system to transfer the files to using FTP. One through 64 characters, dotted decimal format is allowed, no default value, can contain blanks, and it must be specified.

If using a proxy server, this should be the name of the proxy server.

You can include additional FTP command parameters on the TARGET_SYS parameter by using the z/OS UNIX specifications as shown in the topic FTP command -- Entering the FTP environment in *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780. For example, to trace output (`-d`) and use a specific ftpdata_filename (-f"//'WES.MYFTP.DATA'"):

```
TARGET_SYS=-d -f"//'WES.MYFTP.DATA'" testcase.boulder.ibm.com
```

Use the `-p` parameter to specify an alternate IP stack.

USERID

The user ID on the target system that is used to send the files. One through 64 characters, no default value, does not have to be specified, and can contain imbedded blanks. If USERID and PASSWORD are not supplied and NETRCLEVEL=2, the values from the NETRC data set is used for the FTP sessions.

If using a proxy server, this can be the full login to the remote system in the format userid@remote.system.name.

PASSWORD

The password for the USERID on the target system. One through 64 characters and the default value is blanks.

If using a proxy server, this can be the USERID and PASSWORD for the proxy server in the format userid@password.

ACCOUNT

The account data that is sent when an FTP session is started.

TARGET_DSN

The variable (descriptive) portion for the file names on the target system. One through 50 characters, no default value, and it must be specified.

WORK_DSN

The prefix for the data set names of work files on the sending system. One through 40 characters, no default value, and it must be specified.

WORK_DSN_SIZE

The size of the work files in megabytes. One through four decimal digits and the default is 100 MB.

DATACLAS

The data class to use when allocating the work files on the sending system. One through eight characters with no default value.

STORCLAS

The storage class to use when allocating the work files on the sending system. One through eight characters with no default value.

CC_FTP

The number of parallel FTP sessions to use when transmitting the files. One or two decimal digits, the value must be between one and 20, and the default is two.

DIRECTORY

The directory on the target system where the files will be sent with FTP. One through 64 characters, with no default, can contain blanks, and it must be specified.

PMR The PMR number that this file is to be associated with. This field must be 13 characters long and specified in the form `xxxxx.yyy.zzz` with the variables defined as:

Field	Explanation	Sample
xxxxx	PMR Number	34143
yyy	Branch Office	055
zzz	IBM Country Code	724

CIPHER_KEY

The encryption key to use for 192-bit triple DES encryption. The 24 characters following CIPHER_KEY= are used as the key. The key can include imbedded and/or trailing blanks. For example, `CIPHER_KEY=HERE IS CIPHER KEY IN 24` or `CIPHER_KEY=Shortkey`. Note that if `CIPHER_KEY=` is followed by 24 blanks, the file will be encrypted with a key of 24 blanks. If CIPHER_KEY= is not specified, no encryption is performed.

FTPCMDS

An optional DD statement that provides additional flexibility for traversing firewall or proxy servers. When this DD statement is provided, after the initial USERID and PASSWORD are sent, the specified sequential data set is read by the application and the commands contained in the data set are included as FTP commands. The data set must be RECFM=FB and LRECL=80.

JCL examples

Tip: You might want to place your user ID and password into a separate catenated data set. This provides added security because the user ID and password are not directly in the JCL. It is also easier to change the user ID and password across multiple jobs.

The section lists the following examples:

- "Example: Simple FTP connection"
- "Example: FTP connection using a proxy server "
- "Example: FTP connection using a proxy server with proxy user ID" on page 5
- "Example: Using a proxy server with the FTPCMDS DD statement" on page 6
- "Example: Using a proxy server with the FTPCMDS DD statement and a port specification on the TARGET_SYS parameter" on page 6
- "Example: Forcing PASSIVE mode using FTPCMDS inline DD statement" on page 7
- "Example: Using a surrogate userid and NETRC dataset" on page 7

Example: Simple FTP connection

```
//FTP     EXEC PGM=MTFTPS
//STEPLIB  DD  DISP=SHR,DSN=SHANNON.MTFTP.LOAD
//SYSUDUMP DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DISP=SHR,DSN=H44IPCS.WESSAMP.TRKS055K
//SYSIN    DD  *
USERID=anonymous
PASSWORD=anonymous
TARGET_SYS=testcase.boulder.ibm.com
TARGET_DSN=wessamp.bigfile
WORK_DSN=wes.ftpout
CC_FTP=03
WORK_DSN_SIZE=500
DIRECTORY=/toibm/mvs/
PMR=12345.123.123
//
```

Traversing the firewall: You can use the following examples as a starting point to traverse a firewall or proxy server. There are very few common characteristics for firewall or proxy servers, and local customization. If you are able to traverse the firewall or proxy server with a plain FTP statement, modifications to the parameters USERID, PASSWORD, ACCOUNT, and TARGET_SYS, in conjunction with commands in the FTPCMDS data set, the ftp_data file, or both can permit the z/OS Problem Documentation Upload Utility to traverse your firewall or proxy server.

Example: FTP connection using a proxy server

```
//FTP      EXEC PGM=MTFTPS
//STEPLIB   DD DISP=SHR,DSN=SHANNON.MTFTP.LOAD
//SYSUDUMP  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD DISP=SHR,DSN=H44IPCS.WESSAMP.TRKS055K
//SYSIN     DD *
USERID=anonymous@testcase.boulder.ibm.com
PASSWORD=proxyid@proxypw
TARGET_SYS=your.proxy.server.name
TARGET_DSN=wessamp.bigfile
WORK_DSN=wes.ftpout
CC_FTP=03 WORK_DSN_SIZE=500
DIRECTORY=/toibm/mvs/
PMR=12345.123.123
//
```

Example: FTP connection using a proxy server with proxy user ID

```
//FTP     EXEC PGM=MTFTPS
//STEPLIB  DD DISP=SHR,DSN=SHANNON.MTFTP.LOAD
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=H44IPCS.WESSAMP.TRKS055K
//SYSIN DD *
USERID=anonymous@proxyid@testcase.boulder.ibm.com
PASSWORD=proxyid@proxypw
TARGET_SYS=proxy.server.name
TARGET_DSN=wessamp.bigfile
WORK_DSN=wes.ftpout
CC_FTP=03
WORK_DSN_SIZE=500
DIRECTORY=/toibm/mvs/
PMR=99999.000.000
//
```

Example: Using a proxy server with the FTPCMDS DD statement

```
//FTP     EXEC PGM=MTFTPS
//STEPLIB  DD DISP=SHR,DSN=SHANNON.MTFTP.LOAD
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=H44IPCS.WESSAMP.TRKS055K
//FTPCMDS  DD DISP=SHR,DSN=WES.FTPCMDS.DATA
//SYSIN    DD *
USERID=proxyid@testcase.boulder.ibm.com
PASSWORD=proxypw
TARGET_SYS=proxy.server.name 2121
TARGET_DSN=SVCD
WORK_DSN=HLQ.FTPOUT
CC_FTP=03
WORK_DSN_SIZE=500
DIRECTORY=/toibm/mvs/
PMR=99999.000.000
//


*******************************************
* The WES.FTPCMDS.DATA data set contains  *
* user anonymous pw userid@company.com    *
*******************************************
```

Example: Using a proxy server with the FTPCMDS DD statement and a port specification on the TARGET_SYS parameter

```
//FTP     EXEC PGM=MTFTPS
//STEPLIB  DD DISP=SHR,DSN=SHANNON.MTFTP.LOAD
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=H44IPCS.WESSAMP.TRKS055K
//FTPCMDS  DD *
user anonymous pw userid@company.com
//SYSIN    DD *
USERID=proxyid@testcase.boulder.ibm.com
PASSWORD=proxypw
TARGET_SYS=proxy.server.name
TARGET_DSN=SVCD
WORK_DSN=HLQ.FTPOUT
CC_FTP=03
WORK_DSN_SIZE=500
DIRECTORY=/toibm/mvs/
PMR=99999.000.000
CIPHER_KEY=PMR99999sad
//
```

Example: Forcing PASSIVE mode using FTPCMDS inline DD statement

```
//FTP EXEC PGM=MTFTPS
//STEPLIB DD DISP=SHR,DSN=SHANNON.MTFTP.LOAD
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=H44IPCS.WESSAMP.TRKS055K
//FTPCMDS DD *
locsite fwfriendly
//SYSIN DD *
USERID=proxyid@testcase.boulder.ibm.com
PASSWORD=proxypw
TARGET_SYS=proxy.server.name
TARGET_DSN=SVCD
WORK_DSN=HLQ.FTPOUT
CC_FTP=03
WORK_DSN_SIZE=500
DIRECTORY=/toibm/mvs/
PMR=99999.000.000
CIPHER_KEY=PMR99999sad
//
```

Example: Using a userid.NETRC data set

Sample JCL that is storing the proxy login and password in the userid.NETRC data set. (This can be submitted as a surrogate job where the userid.NETRC is not visible to the job originator.) Use of the userid.NETRC data set requires NETRCLEVEL=2, which is set in the FTP.DATA data set.

- Find information about the use of the NETRC data set in *z/OS Communications Server: IP User's Guide and Commands*.
- Find information about the use of the FTP.DATA data set in *z/OS Communications Server: IP Configuration Reference*.

```
//FTP EXEC PGM=MTFTPS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=H44IPCS.WESSAMP.GB01
//SYSIN DD *
TARGET_SYS=-f"//'WES.MYFTP.DATA'" testcase.boulder.ibm.com
PMR=99999.999.000
DIRECTORY=/toibm/mvs/
TARGET_DSN=wes.gb01tst
work_dsn=wes.ftpout
cc_ftp=03 WORK_DSN_SIZE=500
//

For this instance, the userid.NETRC dataset consists of one line:
machine testcase.boulder.ibm.com login anonymous password ibmusr@ibm.com

The FTP.DATA dataset contains:
;*************************************************************************
;*
NETRCLEVEL 2
;*
```