

APPENDIX O

Netview FTP Data Transfer

IBM's System Network Architecture (SNA) is a set of software and hardware products which are integrated in such a way as to provide a variety of capabilities previously unavailable in an IBM environment. Among these is the ability of two computer programs executing at different sites to be in direct communication with one another. The two programs are said to be "in session" with one another in much the same manner as an SNA 3270 terminal is "in session" with a computer program such as CICS.

This capability allows the two programs to exchange data without requiring an intermediate program such as JES2 (although the programs may still interface with JES2 in order to utilize JES2's spooling function). Currently there are two SNA data transfer utilities:

IBM's Netview File Transfer Program for MVS (Netview FTP MVS);

IBM's TSO/E, an interactive program which can transmit files to other TSO/E sites (discussed in part 3 of this section).

Use of NETVIEW File Transfer Program MVS

Netview File Transfer Program for MVS (Netview FTP MVS) is an application program that uses the rules of IBM's System Network Architecture (SNA) to send and receive datasets. For MVS nodes it is the successor of IBM's File Transfer Program Version 2 Release 2 (FTP VS.2) and is available with or without the Advanced Function Feature (AFF). NWRDC operates Netview FTP MVS AFF.

Netview FTP MVS nodes can transfer to and from other nodes running Netview FTP MVS AFF, or Netview FTP MVS with the compatibility PTF installed. (Contact the IBM Support Center for information regarding the PTF(s) applicable to your operating system.) The level of communication can be only on the level of the less sophisticated of the two transfer programs. This means that only functions available to the less sophisticated of the two programs may be utilized.

Netview FTP MVS is designed to give high performance by transferring datasets from the sending to the receiving volumes without intermediate spooling, and by compressing data records, thereby reducing the amount of data sent. Also Netview FTP MVS incorporates a recovery feature which negates the need to retransmit an entire file should the communication link fail.

In Netview FTP MVS the submission of a file transfer request is a process that is separate and independent of the actual processing of the file transfer. The segment which handles the file transfer request (the queue handler) must be started before any other Netview FTP

MVS component can work. The component which transfers the file (a server) works independently of the queue handler.

Netview FTP MVS functions as follows:

- 1) A user, batch job or application program submits a file transfer request to the queue handler. The queue handler puts the request onto a request queue with a status of **waiting**.
- 2) An active server issues an OBTAIN command to the queue handler, who selects a file-transfer request and passes it to the server.
- 3) The local server attempts to start a VTAM session with an active server at the remote node that was specified in the request. The local and the remote servers then determine which is the sending server and which is the receiving server. If requested security information is checked and required data sets are dynamically allocated.
- 4) The sending server then opens the sending data set.
- 5) If requested, the receiving server establishes security and if required dynamically allocates required datasets.
- 6) The receiving server opens the receiving dataset.
- 7) The local server updates the request queue status to **active**.
- 8) The sending server reads records from the input dataset (or from the dataset-handler-module, if appropriate), compresses the data in the manner requested, and packs it into data requested units (RU's). When the request unit is full, the sending server passes it to VTAM which directs its transport to the receiving server.
- 9) The receiving server decompresses the data back to its original form, writes the data to the output dataset (or requests your dataset-handler-routine to handle the data) and updates the checkpoint/restart dataset.
- 10) Steps h and i are repeated until end-of-file is reached on the input dataset.
- 11) When the transfer of data is complete, the sending server closes the input dataset and the receiving server closes the output dataset. Both servers deallocate their datasets, and write out summary information. One of the servers terminates the VTAM session.
- 12) The local server updates the queue request status to **finished**. The local server may now issue another OBTAIN queue handler command.

For ease of operation and maintenance, naming conventions have been established for the definition of the sending and receiving logical unit names in the VTAM cross domain application tables. In order to send or receive multiple datasets at the same time, more than one queue handler and more than 1 server are defined at NWRDC. Use the following when executing Netview FTP MVS AFF at NWRDC:

NWRFPUB1 is the remote logical unit when the district is sending data and is the local logical unit when the district is receiving data from NWRDC.

Copies of FTP installed in the districts bear different names. (See V.6.). Note that the district SENDING logical unit (SLU) name is placed opposite the name of its RECEIVING logical unit (RLU) at NWRDC and vice versa.

Before initiating an FTP transmission, it is prudent to insure that the SNA link between the two data centers is operational. If your terminal has full SNA capability (i.e., you can logon to applications at the remote data center) you can check the line's functionality by attempting a CICS logon at the remote data center. Alternatively, another method can be employed to insure the SNA link is active, and at the same time reduce the necessity to ever execute a wait time. This method requires that the transmitting site submits a job to execute at the receiving site (say NWRDC) whose first step is to route JCL on the internal reader that is punched out as a job back at the transmitting site. This job initiates FTP. In this way, no FTP job is begun at either site unless both data centers are active. Appendix H details the JCL needed to transmit data from a district site and have it received and catalogued at NWRDC using this method.

Recovery/Restart

A user-requested restart can be initiated only if the user restarts FTP and specifies the RESTART=YES option. A checkpoint/restart record would have been written during a previous transmission.

A checkpoint/restart record is written to the checkpoint/restart data set (DVGCR) whenever a user-defined number (RUNUMCR) of request units has been received and successfully written to the output data set. Once a RESTART=YES session has been started, the sending FTP requests information to be sent from the receiving FTP. The sending FTP then transmits data from the input dataset for those records whose relative record number is greater than the record number received from the checkpoint/restart data. Note that a checkpoint/restart dataset need not be defined in a Netview FTP MVS job, since it would have been defined in the jobstream of the server that is to process the request.

Common FTP Errors and Resolution

The following are some errors frequently encountered when executing FTP and some guidelines to follow to resolve them.

- 1) "Error on inquire application status: VTAM cross domain connection to other node is not established or logical unit name is not an application program."
 - a. Contact the systems programmer at the other node to insure that the logical unit name is both defined in the cross domain table and is active.
 - b. Resubmit the job.
- 2) "VTAM open error for logical unit: ACB error code = 5A."
 - a. Make certain that the logical unit name has been typed accurately and agrees with the cross domain table entries.
 - b. Resubmit the job.
- 3) "Logical error for VSAM Put: Reg. 15 return code = 08, Function code in RPL = 00, Feedback Field in RPL = 1C, File name = DVGCR, error message is DVG045I."
 - a. List the contents of the C/R data set, to make sure there are no checkpoint/restart records belonging to data sets for which transmission needs to be started again.
 - b. Delete checkpoint/restart data set and re-create.
 - c. Resubmit the job.

FTP Transmission

Transmission of data files (reporting formats) to NWRDC using FTP is a three step process. Each step must be performed for each file transferred:

- 1) **Execute FTP** using JCL and JES2 statements similar to those on page V-8.
- 2) **Verify that the transmission was successful.** You can do this by checking the print output that will be routed back to you. Successful file transfer is indicated by zero condition codes and a statement in system messages that the dataset has been cataloged. FTP will print a report enumerating the number of records successfully transmitted.
- 3) After a file has been successfully transmitted, **execute a procedure at NWRDC named XMITFINI.** This creates the indicator file used to notify the System and the EDS that a given file has been completely transmitted and is ready for edit processing. Appendix J explains how to execute XMITFINI.

The above procedure is used to transmit both original data files and batch update correction files to NWRDC. To retrieve edit reports and error files from NWRDC, this procedure is played back in reverse (once for each file and report to be retrieved), as follows:

- 4) Using the JCL and JES2 example on V-9 that reverses the process, so that the local site is the receiving site and NWRDC is the sending site, submit one execution for each file you wish to retrieve. The reversing process requires changing the following control parameters:
 - a. Change XMODE parameter,
 - b. reverse the SLU and RLU,
 - c. Update SFTYPE and RFTYPE,
 - d. Change SFILEID and RFILEID,
 - e. Update RSECURP and SSECURP, and
 - f. Modify receiving dataset characteristics as necessary: RTDISP, RRECFM, RLRECL, RBLKSIZE, RUNIT, RVOLSER, RSPCUNIT, RSPCPRIM, RSPCSEC.

If you are retrieving edit report files, it may be easier to use a procedure at NWRDC named NWRRJE instead of FTP. Upon the successful execution of this procedure, your edit reports will be available for printing in your local computer's PRINT queue. Detailed instructions on the use of NWRRJE may be found in Appendix G.

Use of TSO/E Interactive Data Transmission Facility

The TRANSMIT/RECEIVE commands in TSO allow data to be transmitted during an interactive session and received on another system through an interactive session.

The TRANSMIT command will transmit sequential or partitioned datasets, (residing on direct access storage,) of F, FB, FBS, V, VB, VBS or U format. It does not support datasets with keys, i.e., ISAM or VSAM. Transmission of a PDS as a sequential dataset can be forced by using the SEQUENTIAL operand. This method however, does not preserve the directory information. If directory information is to be maintained, unload the PDS with IEBCOPY, transmit the dataset, and have the receiver reload with IEBCOPY.

The RECEIVE command is used to retrieve transmitted files and restore them to their original format. Generally, RECEIVE cannot reformat datasets. The record format of the file received must be the same as that transmitted. The record length must be equal to that transmitted for fixed length records and greater than or equal to that transmitted for variable length records. The blocksize may be any valid blocksize. You can choose to accept the default dataset name and space parameters or to override any of the above.

Transmission of data files (reporting formats) to NWRDC using TSO/E is a three step process. Each step must be performed for each file transferred:

- 1) **Execute TSO/E Commands** using statements similar to those in Appendix I.
- 2) **Verify that the transmission was successful as** described in Appendix I.
- 3) After a file has been successfully transmitted, **execute a procedure at NWRDC named XMITFINI**. This creates the indicator file used to notify the System and the EDS that a given file has been completely transmitted and is ready for edit processing. Appendix J explains how to execute XMITFINI.

The above procedure is used to transmit both original data files and batch update correction files to NWRDC. To retrieve edit reports and error files from NWRDC, this procedure is played back in reverse (once for each file and report to be retrieved), by using the same commands but by being signed on to NWRDC for transmission instead of your local computer.

District FTP LU names and corresponding NWRDC FTP LU names.

<u>SBA1CDRM</u> (Alachua)	<u>NWR1CDRM</u> (NWRDC)
SBA1FTPS	NWR1TRAL
SBA1FTPR	NWR1TSAL
<u>BRE1CDRM</u> (Brevard)	
BRE1FTPS	NWR1TRBV
BRE1FTPR	NWR1TSBV
<u>BRO1CDRM</u> (Broward)	
BRO1FTPS	NWR1TRBW
BRO1FTPR	NWR1TSBW
<u>XCOLCDRM</u> (Collier)	
XCOLFTPS	NWR1TRCO
XCOLFTPR	NWR1TSCO
<u>CDRMO1</u> (Hillsborough)	
HIL1FTPS	NWR1TRHL
HIL1FTPR	NWR1TSHL
<u>LAK1CDRM</u> (Lake)	
LAK1FTPS	NWR1TRLA
LAK1FTPR	NWR1TSLA

<u>LEE1CDRM</u>	(Lee)	
LEE1FTPS		NWR1TRLL
LEE1FTPR		NWR1TSLL
<u>LEO1CDRM</u>	(Leon)	
LEO1FTPS		NWR1TRLE
LEO1FTPR		NWR1TSLE
<u>NFC1CDRM</u>	(NEFEC)	
NFC1FTPS		NWR1TRNC
NFC1FTPR		NWR1TSNC
<u>XOKACDRM</u>	(Okaloosa)	
XOKAFTPS		NWR1TROK
XOKAFTPR		NWR1TSOK
<u>PAL1CDRM</u>	(Palm Beach)	
PAL1FTPS		NWR1TRPB
PAL1FTPR		NWR1TSPB
<u>POL1CDRM</u>	(Polk)	
POL1FTPS		NWR1TRPO
POL1FTPR		NWR1TSPO
<u>STL1CDRM</u>	(St. Lucie)	
STL1FTPS		NWR1TRSL
STL1FTPR		NWR1TSSL
<u>VOL1CDRM</u>	(Volusia)	
VOL1FTPS		NWR1TRVO
VOL1FTPR		NWR1TSVO

Netview FTP Sending and Receiving Examples

1. JCL needed to transmit data from a district DOS/VSE site and have it received and cataloged at NWRDC:

```
* $$ JOB JNM=FTPJOB,CLASS=A
// JOB FTPSEND
// DLBL INPUT,'XXXXXXX',,VSAM,CAT=XXXXXX
// DLBL DVGCR,'XXX.CHECKPOINT.RESTART.FILE',,VSAM,CAT=XXXXXX
```

```

// EXEC DVGFTP,SIZE=AUTO
XMOTE=S                               */ transmission mode (send, receive)
RUSIZE=2048                            */ request unit size
SLU=SBA1FTPS                           */ sending logical unit (Alachua-SBA)
RLU=NWRFPUB1                           */ receiving logical unit
RESTART=NO                              */ restart parameter
TIME=30                                 */ min to wait for partner to be active
SFTYPE=VSAM                             */ sending file type
SFNAME=INPUT
SFFILEID='. . .'                         */ sending file reference name (input)
RFTYPE=SAM                               */ receiving file type
RFILEID='DPSTT.GQ.FFFFF.YNNNNS'
RTDISP=CAT
RUNIT=SYSDA
RVOLSER=DPSR83
RBKLSIZE=23360
RLRECL=160
RRECFM=FB
RSPCPRI=100                             */ primary space allocation
RSPCSEC=10                              */ secondary space allocation
RSPCUNIT=TRK
RSECURP=(ACCT,PPPPP)
/*
* $$ EOJ

```

2. JCL needed to transmit data from NWRDC (OS/VS) and have it received into a catalogued dataset at the local site (DOS/VSE):

```

* $$ JOB JNM=FTPJOB,CLASS=A
// JOB FTPRECV
// DLBL OUTPUT,'XXXXXXXX',,VSAM,CAT=XXXXXX
// DLBL DVGCR,'XXX.CHECKPT.RESTART.FILE',,VSAM,CAT=XXXXXX
// EXEC DVGFTP,SIZE=AUTO
XMODE=R                               */ transmission mode (send, receive)
RUSIZE=2048                            */ request unit size
SLU=NWRFPUB1                           */ sending logical unit (Alachua-SBA)
RLU=SBA1FTPR                            */ receiving logical unit
RESTART=NO                              */ restart parameter
TIME=30                                 */ min to wait for partner to be active
SFTYPE=SAM                             */ sending file type
SFILEID='DPSTT.GG.FFFFF.YNNNNS'       */ sending file reference name (input)
RFTYPE=VSAM                             */ receiving file type
RFNAME=OUTPUT
SSECURP=(ACCT,PPPPP)

```

```
RFILEID='. . .' */ receiving file reference name (output)
RESTART=NO
/*
* $$ EOJ
```

When modifying any of the above JCL you need to change the following statements:

```
* $$ JOB...
//NWRFTPXX JOB...
/*ROUTE...
/*PASSWORD...
//OUTPUT DD... or //DLBL OUTPUT
//INPUT DD... or //DLBL INPUT
```

ACCT must be a valid account number. Valid account numbers at both the sending and the receiving sites are prerequisite.

NN is the CPU time in seconds. To estimate the amount of CPU time an FTP job will require use the following:

For FB files: $\#records * lrecl / 300,000 = \text{CPU seconds}$
For VB files: $\#blocks * blksize / 300,000 = \text{CPU seconds}$
For unblocked $\#tracks * \text{bytes-per-track} / 300,000 = \text{CPU seconds}$
(bytes-per-track varies for a disk pack type)

Note: Use a minimum time of 5 seconds to allow for session start-up procedures.

PPPPP

This is the password for ACCT account.

NNN.RYY

This variable allows users to route print to their local printer to view job and/or check the status of the output file after the receive job has terminated.

SLU= and RLU=

Designate the sending and receiving units (see V-6, 7).

//DLBL INPUT or //INPUT DD

This defines the original file at the transmitting data center. If this file is a catalogued dataset, the only required parameters are DSN and DISP=SHR. If the dataset is not catalogued, UNIT and VOL=SER parameters will also be needed. If the dataset is on tape, add a LABEL parameter. At DOS sites this will be the //DLBL Input statement.

//DLBL OUTPUT or //OUTPUT DD

This defines the new file at the receiving data center. DSN, UNIT, DISP, VOL=SER, and DCB are all required parameters. If the OUTPUT file is to reside on disk, a SPACE parameter is also required. At DOS sites this will be the //DLBL Output statement.

Use the following when defining the output statement:

DSN=DPSTT.GG.FFFFFF.YNNNNS

where TT is your district number prefixed by "DPS"

GG is = GQ for Student files or = GU for Staff files or = EU for Finance files

FFFFFF is defined in Appendix S

YNNNN is the Survey Year prefixed by "Y", and

S is the number of the Survey Period being submitted.

See Appendix S for Reporting Format record, blocksize and space information.

If transferring a file from one type of disk to another, remember that the space requirements may change. For example, a file occupying 30 tracks on a 3350 will require 45 tracks on a 3330-1, but only 15 on a 3380. The computations above are correct for the type of disk you will be using (3380).