User’s Guide

Release 2.1

This is a major revision of, and obsoletes, SH12-5656-03.

This edition applies to Version 2 Release 2 Modification Level 1 of NetView File Transfer Program for MVS (5685-108) and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers’ comments may be provided at the back of this publication, or you may address your comments to the following address:

IBM Deutschland Entwicklung GmbH
Information Development, Dept. 0446
Postfach 1380
71003 Boeblingen
Germany

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1988, 1994. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
Chapter 4. Writing NetView FTP V2.2.1 MVS Application Programs

Initializing and Assigning Values to the Fields of the APL and APX

Creating Requests

Transferring Members, Groups of Members, and Entire PDSs

Writing an Application Program to Create a Request to Transfer a Group of Members

Writing an Application Program to Create a Request to Transfer All Members Except Those Selected

Writing an Application Program to Create a Request to Transfer an Entire PDS

Querying Requests in the Queue

Modifying, Holding, and Releasing a Request in the Queue

Restarting Requests in the Queue

Deleting Requests from the Queue

Using the Message Area

NetView FTP V2.2.1 MVS Macros Used by Application Programs

How to Call NetView FTP V2.2.1 MVS

NetView FTP V2.2.1 MVS Mapping Macros

Specifying the APL and APX Communication Fields

APL Control Block Identifier

Return and Reason Codes

Current Level of NetView FTP and Length of the APL

Address and Length of the QRA

Number of Query Records Retrieved and Moved to the QRA

Address and Length of the Message Area

Number of Messages Issued and Moved to the Message Area

Maximum Number of Members that Can Be Selected or Excluded

Specifying the APX Control Block Identifier

Chapter 5. Controlling Requests and Administering the Queue

Controlling Requests

Administering the Request Queue

Chapter 6. Examples of Using the NetView FTP V2.2.1 MVS Panels

Example 1: Sending a PS Data Set

Example 2: Receiving a Labeled Tape File as a PS Data Set

Example 3: Sending Members of a PDS

Example 4: Receiving a Physical Sequential DASD Data Set as a VSAM Cluster

Example 5: Receiving an OS/2 Text File as a PS Data Set

Example 6: Sending a PS Data Set to OS/2

Example 7: Sending a PS Data Set Using OSI/File Services

Chapter 7. Sample NetView FTP V2.2.1 MVS Batch Jobs

Sample Batch Job That Adds a Request and Queries All a User’s Own Requests

Sample Batch Job That Modifies a Request’s Class and Queries All a User’s Own Requests

Sample Batch Job That Releases a Request and Queries All a User’s Own Requests

Sample Batch Job That Restarts a Request
References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM’s product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM’s intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user’s responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood NY 10594, U.S.A.

Programming Interfaces Information

This book is intended to help the customer use NetView FTP V2.2.1 MVS. This book also documents General-Use Programming Interface and Associated Guidance Information.

General-Use programming interfaces allow the customer to write programs that obtain the services of NetView FTP V2.2.1 MVS.

General-Use Programming Interface and Associated Guidance Information is identified where it occurs by an introductory statement to a chapter or section.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

<table>
<thead>
<tr>
<th>ACF/VTAM</th>
<th>MVS/ESA</th>
<th>OS/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>MVS/XA</td>
<td>OS/400</td>
</tr>
<tr>
<td>NetView</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RACF</td>
<td>VM/XA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AS/400</td>
<td></td>
</tr>
</tbody>
</table>
About This Book

This book applies to the IBM* licensed program NetView* File Transfer Program Version 2 Release 2 Modification Level 1 for the MVS operating system (abbreviated to NetView FTP V2.2.1 MVS in this book), which is a member of the NetView family of products. This book describes how to use NetView FTP.

Abbreviations and Special Terms

Throughout this book, the following abbreviations are used:

NetView FTP
NetView File Transfer Program (regardless of version number and operating system)

NetView FTP V2 MVS
NetView File Transfer Program Version 2 for MVS (regardless of release number)

NetView FTP V2.1 MVS
NetView File Transfer Program Version 2 Release 1 for MVS

NetView FTP V2.2 MVS
NetView File Transfer Program Version 2 Release 2 for MVS

NetView FTP V2.2.1 MVS
NetView File Transfer Program Version 2 Release 2 Modification Level 1 for MVS

NetView FTP VSE
NetView File Transfer Program for VSE

NetView FTP VM
NetView File Transfer Program for VM

NetView FTP/400
NetView File Transfer Program for OS/400*

NetView FTP for Workstations (OS/2* and AIX*)
NetView File Transfer Program Server/2 and
NetView File Transfer Program Client/2
NetView File Transfer Program Server AIX and
NetView File Transfer Program Client AIX

NetView FTP Server for OS/2
NetView File Transfer Program Server/2

NetView FTP Client for OS/2
NetView File Transfer Program Client/2

NetView FTP Server AIX
NetView File Transfer Program Server AIX

NetView FTP Client AIX
NetView File Transfer Program Client AIX
In this book:

- The abbreviation MVS refers to MVS/370, MVS/XA*, and MVS/ESA* systems.
- The abbreviation VM refers to VM/SP and VM/XA* systems.
- The terms data set, file, and cluster are used as synonyms.

If you come across an unfamiliar word, refer to the glossary on page 145.

Who Should Read This Book

This book is written for people who want to use NetView FTP V2.2.1 MVS to transfer files from one node of a network to another. This book assumes that readers know how to create and work with the types of files they want to transfer, and that users of the NetView FTP V2.2.1 MVS panels are already familiar with the Interactive System Productivity Facility (ISPF).

How to Use This Book

Read Chapter 1 first, it gives you an overview of what NetView FTP V2.2.1 MVS is, how it works, and what it can do.

If you are using the NetView FTP V2.2.1 MVS panels, read Chapter 2, “Using the NetView FTP V2.2.1 MVS Panels” on page 27. You should also refer to the NetView FTP Parameter Reference for descriptions of the parameters.

If you are writing batch jobs or application programs, read Chapter 3, “Writing NetView FTP V2.2.1 MVS Batch Jobs” on page 55, or Chapter 4, “Writing NetView FTP V2.2.1 MVS Application Programs” on page 67. You should also refer to the NetView FTP Parameter Reference for descriptions of the parameters.

The following parts have been moved to a new publication, the NetView FTP Parameter Reference, SH12-6052:

- “Parameter Reference”
- “Alphabetical List of Parameters”
- “Alphabetical List of Keywords”
- “Alphabetical List of Field Names”
- “How NetView FTP Creates Names for Receiving Files”
- “How NetView FTP Gets VSAM Parameter and DD Statement Values”
- “Relationships between Sending File and Receiving File Attributes”.
Chapter 1. What NetView File Transfer Program for MVS Is and Does

NetView FTP V2.2.1 MVS is an application program that lets a node of a network transfer files quickly and reliably to or from other nodes of that network using either Systems Network Architecture (SNA) protocols or Open Systems Interconnection (OSI) protocols.

Any computer system at which NetView FTP V2.2.1 MVS is installed can send files to or retrieve files from any other computer system at which one of the following is installed:
- NetView FTP V2.2 MVS
- NetView FTP V2.1 MVS
- NetView FTP VM
- NetView FTP VSE
- NetView FTP/400
- NetView FTP for Workstations.

NetView FTP transfers files directly from a storage device at one system to a storage device at another system, without intermediate spooling.

NetView FTP V2.2.1 MVS can also transfer files to or from a node in a non-SNA network using OSI File Transfer Access and Management (FTAM) protocols. These transfers are conducted by the IBM licensed program OSI/File Services. For more information about OSI/File Services, refer to OSI/File Services General Information Manual.

Figure 1. Network with NetView FTP

Because the transfer programs at any two systems work together to transfer files, they can communicate only on the level of the less sophisticated of the two transfer programs.
Creating File-Transfer Requests

You use NetView FTP V2.2.1 MVS by creating file-transfer requests and submitting them to NetView FTP V2.2.1 MVS for processing. A request tells NetView FTP V2.2.1 MVS such things as:

- Whether to send a file to or retrieve a file from another system
- Where to send the file to or retrieve it from
- Which file is to be sent
- What name the file is to have when it is received
- Whom to notify after the file has been transferred.

The system where the request is submitted is called the requesting system.

You create and submit requests by:

- Filling in the fields of a set of panels
- Writing and then running a batch job
- Writing and then running an application program.

After you submit a request, you can immediately submit another—you do not need to wait until NetView FTP has processed the previous request.

After you submit a request, NetView FTP adds it to the request queue where it waits to be processed. NetView FTP assigns an identifier to this request, the request number. This request number remains associated with the request until the request is deleted. You must specify the request number when you want to query, modify, restart, hold or release a specific request.

After a request is added to the queue, but before it is processed, it is said to have the status waiting.

While a request is being processed and while its corresponding file transfer is taking place, the request is said to have the status active. After the file transfer is complete, NetView FTP V2.2.1 MVS changes the status of the corresponding request from active to finished.
Controlling File-Transfer Requests

NetView FTP provides functions to control file-transfer requests after they have been added to the queue.

You can:
- Ask NetView FTP V2.2.1 MVS for information about the requests.
- Modify some of the attributes of your waiting requests.
- Put on hold any waiting request that you submitted. This prevents the request from being processed until you release it.
- Restart file transfers that have not finished successfully.
- Delete requests from the queue. Deleting a waiting request prevents NetView FTP V2.2.1 MVS from processing the request. Deleting an active request interrupts the file transfer in progress. Deleting requests makes room in the queue for new requests.

You can control requests by:
- Filling in the fields of a set of panels
- Writing and then running a batch job
- Writing and then running an application program.

How NetView FTP V2.2.1 MVS Processes Requests

The processing of requests takes place independently of the adding of requests to the request queue. To process requests, NetView FTP V2.2.1 MVS uses components called servers. For transfers between nodes in an SNA network NetView FTP V2.2.1 MVS uses NetView FTP SNA servers (abbreviated to SNA servers in this book). For transfers between nodes in a non-SNA network NetView FTP V2 MVS uses NetView FTP OSI servers (abbreviated to OSI servers in this book).

An OSI server does not transfer files, it only passes the requests to OSI/File Services, which conducts the file transfer.

The following description applies to file transfers using SNA protocols only. In the following illustrations, a server is represented by a small box with an S inside. Each server runs independently of any other servers.
A server at the system where you submitted a request (the *requesting* system) begins processing the request by retrieving a copy of the request from the request queue and changing the request’s status from *waiting* to *active*.

The server examines the request to determine, among other things, the name of the system to or from which it is to transfer a file (the *responding* system). The server then contacts a server at the responding system and initiates a *conversation*.

After the conversation has been established, the requesting server sends information contained in the request to the responding server. Among this information is the direction of the file transfer (to or from the responding system). The direction of the file transfer is independent of which server initiated the conversation.
After both servers have examined the request, they both know which of them is to send the file and which of them is to receive the file. The sending server allocates the file to be sent (the *sending file*) and the receiving server allocates the file into which the sending file is to be received (the *receiving file*). If the receiving file does not already exist, the receiving server allocates it before the file transfer begins. Both servers then work together to transfer the file.

After the servers have finished transferring the file, they terminate the conversation. Each server enters data in a report file and sends a message to users whose user IDs were specified in the request. The server that initiated the conversation changes the request’s status from *active* to *finished*, and is then ready to process the next request.

The requesting server retrieves a copy of the next request from the queue and the process begins again.
All the servers at each system simultaneously follow the same procedure to transfer files throughout the network.
Using NetView FTP V2.2.1 MVS

The following describes the ways you can use NetView FTP V2.2.1 MVS when it is the requesting system.

Using the NetView FTP V2.2.1 MVS Panels

NetView FTP V2.2.1 MVS has a set of Interactive System Productivity Facility (ISPF) panels that make it easy for you to work with your file-transfer requests. You can use the panels to:

- Create a request and then submit it for processing, save it in an ISPF table data set for future use, or both
- List your saved requests and select saved requests you want to submit, change, or discard
- Query and delete submitted requests
- Modify the classes and priorities of submitted requests
- Hold and release submitted requests or entire server classes
- Restart interrupted file transfers.

The NetView FTP V2.2.1 MVS panels include a set of help panels that provide you with online help whenever you need it.

Which of the panels NetView FTP V2.2.1 MVS displays and the order in which it displays them depends on the task you select.

Using NetView FTP V2.2.1 MVS Batch Jobs and Application Programs

You can also create, query, delete, restart, modify, hold, and release requests by writing and running a batch job or application program. Samples of such batch jobs and application programs are shown in Figure 2 and Figure 3.

```plaintext
/* NetView FTP Batch Job for MVS */
/* SMP JOB (8459,0),'UID',MSGCLASS=X, */
/* CLASS=A,NOTIFY=UID,USER=UID */
/* */
/* */
/* STEP1 EXEC PGM=DVGIFBI */
/* DVGLOG DD SYSOUT=* */
/* SYSIN DD * */
FUNCTION=ADD
XMODE=TO
RMTNODE=NEWYORK
SFILERG=PS
SFILERO='NET.DS.SEQ/zerodot1'
* */
/* */
/* */
```

Figure 2. Example of a NetView FTP V2.2.1 MVS Batch Job
In a NetView FTP V2.2.1 MVS batch job or application program, you can specify that the job or program is not to continue running until NetView FTP V2.2.1 MVS has completed the file transfer. This way, you can use the return code from the file transfer for subsequent processing.

Application programs written in PL/I, Assembler, REXX, or C language can use macros, provided by NetView FTP V2.2.1 MVS, to load and call the NetView FTP V2.2.1 MVS application program interface routine and to map the interface control block.

NetView FTP V2.2.1 MVS application programs can be written in other high-level IBM languages, but such programs must load and call the NetView FTP V2.2.1 MVS application program interface routine directly, and must carry out their own data mapping.

Identifying a Request's Originator

An originator can submit a file-transfer request using any of the interfaces NetView FTP provides. NetView FTP V2.2.1 MVS automatically identifies the originator of a request and substitutes the user ID into the request.

One way that NetView FTP uses the request-originator identifier is for access-security checking.

NetView FTP V2.2.1 MVS determines the request-originator identifier differently for each interface.

Determining the Identifier of a Request Originated in the Interactive Interface

If a security product, such as RACF*, is active, NetView FTP V2.2.1 MVS uses the security user ID of the TSO user as the identifier, otherwise the TSO user ID is used.
Determining the Identifier of a Request Originated in the Batch Job Interface

For a submitted batch job when a security product is active, NetView FTP V2.2.1 MVS uses the security user ID of the job submitter as the identifier, otherwise the job name is used.

For a request submitted as a started task when a security product is active:

- If the table ICHRIN03 is defined, with the started task in it, NetView FTP V2.2.1 MVS uses the security user ID assigned to the started task.
- If the table ICHRIN03 is not defined, or is defined without the started task in it, NetView FTP V2.2.1 MVS uses the name of the started task.

For a request submitted as a started task when no security program is active, NetView FTP V2.2.1 MVS uses the name of the started task.

Determining the Identifier of a Request Originated in the Application Program Interface

For a request invoked in a batch job or started task, the rules as described for the batch job interface apply.

If it is invoked as a batch TSO job, the TSO user ID is taken.

If it is invoked within a TSO command processor and a security program is active, the security user ID of the TSO user is taken, otherwise the TSO user ID is taken.

Protecting Requests from Unauthorized Access

NetView FTP V2.2.1 MVS notes the originator ID of every interactive user, batch job, or application program that submits a request to the queue. Only users sharing your originator ID, or users who know a special password called the master password, are able to query, delete, restart, modify, hold, or release requests that you submit. If other users share your originator ID, you can protect your requests by assigning a request password to each request. Anyone sharing the same originator ID must know the password or the master password to be able to query, delete, restart, modify, hold, or release submitted requests.

If you know the master password, you can query, delete, restart, modify, hold, or release any request, regardless of whether it is password-protected, and regardless of who submitted it. The master password is set by your system administrator.
Assigning Server Classes and Priorities to Requests

To each request you create, you assign one of 36 server classes (A to Z and 0 to 9) and one of 10 request priorities (0 to 9). The server class you specify determines which servers can process it. A server can process a request only if the server class specified in that request is one of those assigned to that server.

A NetView FTP V2.2.1 MVS system can use the different classes to group different types of file transfers. For example, at your system a certain class might be reserved for one of the following transfers:

- Of certain types of files
- Of files of a certain size
- To or from a certain remote system
- To or from a certain storage device type
- Using OSI protocols.

When a server is ready to process a request, it asks the queue handler to obtain one for it. The queue handler is the NetView FTP component that controls access to the request queue. The queue handler searches through all the requests in the queue for those with classes the server is allowed to process. Of those, it selects the request with the highest priority. If there is more than one request with the highest priority, it selects the request with the highest class. If there is more than one request with the highest class, it selects the request that was submitted earliest. After the queue handler selects a request, it passes it to the server.

Ask your NetView FTP administrator which server classes are reserved for OSI file transfers and which classes are for SNA file transfers. If you transfer a file using OSI protocols, and you specify a server class that is reserved for SNA file transfers, your request will remain in the request queue and will not be processed.

Specifying when NetView FTP V2.2.1 MVS Can Process a Request

In each request, you can specify dates and times before which and after which NetView FTP V2.2.1 MVS is not to process that request.

You can specify that a request that is submitted every day is scheduled for processing within a specific time frame, provided that the not-after time is later than the not-before time. NetView FTP V2.2.1 MVS assumes the not-before date or the not-after date to be the current date if you do not specify a date or dates.

If you do not specify dates and times, your request can be scheduled at any time after it has been submitted.
Identifying the Responding System

For SNA file transfers, you can identify the system to or from which you want to transfer a file in one of the following ways:

- In a request, you can specify the LU name of a server at the responding system. When a requesting server processes your request, it tries to establish a conversation with that server. If that server has not started or is busy with other file transfers, the requesting server will try again periodically.

In transfers to or from OS/2 workstations, the LU name identifies NetView FTP Server for OS/2. A NetView FTP Client for OS/2 workstation is addressed via the APPC conversation security parameter.

In transfers to or from AIX workstations, the LU name identifies NetView FTP Server AIX. A NetView FTP Client AIX is addressed via the remote workstation ID parameter.

- If your NetView FTP system uses a server group table, you can, in a request, specify the name of a group of servers at the responding system instead of specifying the LU name of just one server. A server group table is a special file, maintained by your system administrator, in which the LU names of servers can be grouped together and each group given a name. The server group names can be meaningful, which makes them easier to remember than LU names. The requesting server that processes the request will then try to establish a conversation with the server that is alphabetically first in the group. If that one is not started or is busy, the requesting server tries the next one. The requesting server keeps trying until it has succeeded in establishing a conversation or has tried all the servers in the group.

An entry in the server group table for NetView FTP for Workstations (AIX or OS/2), however, does include only the LU name of the NetView FTP Server.

Using server groups lets idle servers fill in for servers that are not yet started or are busy, and thereby improves network throughput. The server group table can also contain information about each server, such as the operating system at the server’s system. NetView FTP V2.2.1 MVS uses this information to check requests for incorrect or missing parameter values before adding the requests to the queue.

For OSI file transfers, you identify the remote FTAM application by specifying the Application Entity Title (AET). LU names and server groups do not apply to OSI file transfers.
Transferring Files To or From Workstations

NetView FTP V2.2.1 MVS can transfer files to and from workstations running:

- **OS/2:**
  - A stand-alone workstation that is running NetView FTP Server for OS/2 and is connected to a communication partner in an SNA network on a peer-to-peer level
  - A workstation in a local area network that is running:
    - NetView FTP Server for OS/2 and is connected to a communication partner in an SNA network on a peer-to-peer level
    - NetView FTP Client for OS/2 and is connected to the SNA network via a gateway. Gateway and routing functions are provided by NetView FTP Server for OS/2.

For file transfers between NetView FTP V2.2.1 MVS and NetView FTP Client for OS/2, both of the following connections have to be established:

- The connection between NetView FTP V2.2.1 MVS and NetView FTP Server for OS/2
- The connection between NetView FTP Server for OS/2 and NetView FTP Client for OS/2.

- **AIX:**
  - A stand-alone workstation that is running NetView FTP Server AIX and is connected to a communication partner in an SNA network on a peer-to-peer level
  - A workstation in a local area network that is running:
    - NetView FTP Server AIX and is connected to a communication partner in an SNA network on a peer-to-peer level
    - NetView FTP Client AIX and is connected to the SNA network via a gateway. Gateway and routing functions are provided by NetView FTP Server AIX.

For file transfers between NetView FTP V2.2.1 MVS and NetView FTP Client AIX, both of the following connections have to be established:

- The connection between NetView FTP V2.2.1 MVS and NetView FTP Server AIX
- The connection between NetView FTP Server AIX and NetView FTP Client AIX.
Automatic Logon Retry

Sometimes a server at an MVS, VSE, or VM system cannot initiate a conversation with a server at a remote system because:

- All the servers at the remote system are busy with other transfers
- No servers at the remote system have been started
- ACF/VTAM* is temporarily unable to find a path between the two servers.

When this happens, the server at your system automatically changes the status of the request from **active** back to **waiting**, and goes on to process the next request in the request queue. Later, NetView FTP tries again to process the request. It keeps trying until it succeeds in initiating a conversation. This is called using **automatic logon retry**.

For OSI requests, because OSI/File Services is responsible for the file transfer, NetView FTP V2.2.1 MVS does not perform any automatic logon retry.

Running Jobs after a File Transfer

NetView FTP V2.2.1 MVS lets you specify, in a request, the names of up to 12 jobs at each of the two systems involved in the file transfer. Which of the jobs are submitted depends on whether the file transfer ended successfully or not. You can use these jobs to do such things as:

- Submit a request for a second file transfer after the first file transfer ends
- Submit a job after using NetView FTP V2.2.1 MVS to transfer it to a remote system.

**Note:** For OSI file transfers, jobs can only be submitted at the system that initiated the request.
What You Need to Know About Files

The following describes the things you need to know about files to send them to a remote system or retrieve them from a remote system.

The sending and receiving systems need not have the same operating systems. Also, a file being received need not be of the same type as the file being sent.

Types of Files NetView FTP Can Transfer

The following section gives an overview about the file organizations that NetView FTP supports and how the file organizations can be converted.

File Organizations Supported by NetView FTP V2.2.1 MVS, NetView FTP VM, and NetView FTP VSE

The following list gives the access methods NetView FTP MVS, VM, and VSE use when processing the different file types:

<table>
<thead>
<tr>
<th>Access Method</th>
<th>Types of Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSAM</td>
<td>ESDS—Entry Sequenced Data Set (MVS, VM, VSE)</td>
</tr>
<tr>
<td></td>
<td>KSDS—Key Sequenced Data Set (MVS, VM, VSE)</td>
</tr>
<tr>
<td></td>
<td>RRDS—Relative Record Data Set (MVS)</td>
</tr>
<tr>
<td></td>
<td>SAM ESDS—VSAM Managed Sequential (VSE)</td>
</tr>
<tr>
<td></td>
<td>LDS—Linear Data Set (MVS)</td>
</tr>
<tr>
<td>QSAM</td>
<td>Physical sequential file stored on a DASD (MVS, VM)</td>
</tr>
<tr>
<td></td>
<td>Physical sequential file stored on labeled tape (MVS, VM)</td>
</tr>
<tr>
<td></td>
<td>Physical sequential file stored on unlabeled tape (MVS, VM)</td>
</tr>
<tr>
<td></td>
<td>Single PDS member, without directory information (MVS)</td>
</tr>
<tr>
<td>SAM</td>
<td>Physical sequential file stored on labeled tape (VSE)</td>
</tr>
<tr>
<td></td>
<td>Physical sequential file stored on unlabeled tape (VSE)</td>
</tr>
<tr>
<td>BPAM and BSAM</td>
<td>Single PDS member, with directory information (MVS)</td>
</tr>
<tr>
<td></td>
<td>Group of PDS members, with directory information (MVS)</td>
</tr>
<tr>
<td></td>
<td>Entire PDS, with directory information (MVS).</td>
</tr>
</tbody>
</table>

Notes:

1. For OSI requests, you can only transfer physical sequential files that are stored on DASD. The remote operating system is irrelevant.

2. NetView FTP does not allow records to be truncated. If a sending file record is bigger than the maximum record size of the receiving file (non-VSAM at an MVS or VM system), the transfer is aborted.

3. NetView FTP has the following record processing limitations:

   a. For non-VSAM files at an MVS or VM system, the maximum logical record length supported is 32760; for non-VSAM files at a VSE system, the maximum is 32767.

   b. For VSAM files, the maximum record length supported is 32761. If a VSAM cluster contains longer records, you can unload the VSAM file to a sequential file and transfer the sequential file with NetView FTP. After the file transfer, you can reload the sequential file to the VSAM file.

   c. Spanned records are supported as long as the above length limitations are not exceeded.
File Organizations Supported by NetView FTP/400

NetView FTP/400 can process the following types of files:

**Physical data file**
A member in a physical data file can contain both numeric and character data of any AS/400* data type. The maximum length of this type of file is 32766 bytes.

**Physical source file**
A physical source file generally stores text or source statements instead of data as follows:
- The first field of a source file is always a sequence number (6 bytes).
- The second field always contains a date (6 bytes).
- The third part contains text. Text fields are of the type character.

The maximum length of this type of file is 32766 bytes.

**Save file**
Is used to store data with a fixed record length of 528 bytes on a disk.

<table>
<thead>
<tr>
<th>Sending File</th>
<th>Receiving File</th>
</tr>
</thead>
<tbody>
<tr>
<td>data file (DTAF) at an OS/400 system</td>
<td>physical sequential at an MVS system single PDS member without directory information at an MVS system VSAM ESDS at an MVS system VSAM KSDS at an MVS system</td>
</tr>
<tr>
<td>source file (SRCF) at an OS/400 system</td>
<td>physical sequential at an MVS system single PDS member without directory information at an MVS system VSAM ESDS at an MVS system VSAM KSDS at an MVS system</td>
</tr>
<tr>
<td>save file (SAVF) at an OS/400 system</td>
<td>physical sequential at an MVS system single PDS member without directory information at an MVS system VSAM ESDS at an MVS system</td>
</tr>
<tr>
<td>physical sequential at an MVS system single PDS member without directory information at an MVS system VSAM ESDS at an MVS system VSAM KSDS at an MVS system</td>
<td>data file (DTAF) at an OS/400 system</td>
</tr>
<tr>
<td>physical sequential at an MVS system single PDS member without directory information at an MVS system VSAM ESDS at an MVS system VSAM KSDS at an MVS system</td>
<td>source file (SRCF) at an OS/400 system</td>
</tr>
<tr>
<td>physical sequential at an MVS system single PDS member without directory information at an MVS system VSAM ESDS at an MVS system VSAM KSDS at an MVS system</td>
<td>save file (SAVF) at an OS/400 system</td>
</tr>
</tbody>
</table>
File Organizations Supported by NetView FTP for Workstations

NetView FTP for Workstations can process files of any file organization. There are, however, different modes of file handling:

**Record-oriented file handling mode**
Record delimiters are not regarded as part of a file’s data. NetView FTP for Workstations at the sending node removes any record delimiters from the sending file’s data. On the receiving node, NetView FTP for Workstations adds record delimiters to the received records.

**Byte-oriented file handling mode**
NetView FTP for Workstations handles the data as a consecutive stream of bytes.

The following combinations are valid when transferring files between MVS and workstations:

<table>
<thead>
<tr>
<th>Sending File</th>
<th>Receiving File</th>
</tr>
</thead>
<tbody>
<tr>
<td>record-oriented at a workstation</td>
<td>physical sequential at an MVS system</td>
</tr>
<tr>
<td></td>
<td>VSAM ESDS at an MVS system</td>
</tr>
<tr>
<td></td>
<td>VSAM KSDS at an MVS system</td>
</tr>
<tr>
<td>byte-oriented at a workstation</td>
<td>physical sequential at an MVS system</td>
</tr>
<tr>
<td></td>
<td>VSAM ESDS at an MVS system</td>
</tr>
<tr>
<td></td>
<td>VSAM LDS at an MVS system</td>
</tr>
<tr>
<td>physical sequential at an MVS system</td>
<td>record-oriented at a workstation 1)</td>
</tr>
<tr>
<td></td>
<td>byte-oriented at a workstation 1)</td>
</tr>
<tr>
<td>VSAM ESDS at an MVS system</td>
<td>record-oriented at a workstation 1)</td>
</tr>
<tr>
<td></td>
<td>byte-oriented at a workstation 1)</td>
</tr>
<tr>
<td>VSAM KSDS at an MVS system</td>
<td>record-oriented at a workstation 1)</td>
</tr>
<tr>
<td></td>
<td>byte-oriented at a workstation 1)</td>
</tr>
<tr>
<td>VSAM LDS at an MVS system</td>
<td>byte-oriented at a workstation</td>
</tr>
</tbody>
</table>

**Note:** 1) The record structure will be lost.

**General Considerations**
You can send files of one type to files of most other types, with the following exceptions:

- A VSAM LDS can only be transferred to another VSAM LDS.
- A VSAM RRDS can only be transferred to another VSAM RRDS.
- An entire PDS, or selected members of a PDS with directory information, can only be transferred to another PDS.

You can use NetView FTP V2.2.1 MVS to transfer files of types other than those mentioned in the list. However, to do this you must use one of the following:

- A routine that gains access to such files and passes their records on to NetView FTP V2.2.1 MVS. Such routines are called **user-written file handlers**.
- A utility routine that converts such files into files that NetView FTP V2.2.1 MVS can transfer. NetView FTP V2.2.1 MVS can call such routines directly, via user exits.
If you are transferring files to or from OS/400, then you should consider the following points. Files at OS/400 locations have records that all have fixed length. With the exception of source files, the records contain user data.

- If a source file is received from NetView FTP/400, only the user data is stored.
- A save file cannot be created on a system other than OS/400. Therefore, before a save file can be sent to NetView FTP/400, it must have been previously received.

A save file received from NetView FTP/400 should not be sent back as another file type. A source or data file received from NetView FTP/400 cannot be sent back as a save file.

- If you send a file to NetView FTP/400 and its records are shorter than those specified for the OS/400 file, or they are of variable length, NetView FTP/400 stores the records with blanks padded to the correct length.
- The file sent to NetView FTP/400 must not have a logical record length that exceeds the maximum possible (32766 bytes). The file you receive from NetView FTP/400 must not have a record length that exceeds the maximum allowed for the specified record format.

If you are transferring files from a workstation in record-oriented mode, you should consider that the records sent are of variable length. NetView FTP V2.2.1 MVS does not allow to transfer a file with variable length records into a file with fixed length records unless a padding character is explicitly specified in the request.

**User-Written File Handlers**

User exit routines that gain access to files and pass their records on to NetView FTP V2.2.1 MVS are called *user-written file handlers*. NetView FTP V2.2.1 MVS has a special user exit for user-written file handlers. This user exit is described in the *NetView FTP Customization* guide.

NetView FTP V2.2.1 MVS communicates with user-written file handlers through a storage area for which it also provides an Assembler mapping macro. You can pass data to user-written file handlers by assigning values to special parameters provided for that purpose.

**Note:** You cannot use user-written file handlers for file transfers with OSI protocols or for file transfers into and from a workstation.

**Converting File Types**

You can use a utility routine to convert a file from a file organization that NetView FTP V2.2.1 MVS does not support to a file organization NetView FTP V2.2.1 MVS supports. Examples of such routines are:

- The Access Method Services (AMS) EXPORT and REPRO commands (for DL/I databases)
- The Hierarchic Direct Access Method (HDAM) UNLOAD function (for DL/I databases)
- The Hierarchic Indexed Sequential Access Method (HISAM) UNLOAD function (for DL/I databases)
• The Information Management System/Virtual Storage (IMS/VS) image copying utilities DFSUDMP0 and DFSURDB0 (for DL/I databases)
• IEHMOVE (to unload members into a physical sequential file).

You can use the NetView FTP V2.2.1 MVS pre-transfer user-exit routine to call such routines. The pre-transfer user-exit routine is described in the NetView FTP Customization guide.

Options for PDSs
When you transfer members from one PDS to another PDS you can specify that each member of the sending PDS is to be written to the receiving PDS:

• Only if a member with the same name does not already exist in the receiving PDS. The member is said to be inserted.
• Only if a member with the same name already exists in the receiving PDS. The member is said to be updated.
• Regardless of whether a member with the same name already exists in the receiving PDS. The member is said to be replaced.

If you transfer several members at once, they need not all have the same PDS option.

Options for VSAM Key-Sequenced Data Sets
If the receiving file is a VSAM key-sequenced data set (KSDS), NetView FTP V2.2.1 MVS can do the following:

• Add the records being sent to an empty receiving file
• Add new records to the receiving file and replace existing records with those being sent
• Add new records to the receiving file; discontinue the transfer if any records in the sending and receiving files have the same key
• Replace records in the receiving file with those being sent; discontinue the transfer if any record in the sending file has a key that no record in the receiving file has
• Delete those records in the receiving file that have the same key as those being sent.

Transferring Physical Sequential Files Stored on Tape
NetView FTP V2.2.1 MVS can transfer files to or from labeled or unlabeled tapes. You cannot, however, transfer the contents of an entire tape containing more than one file.

Transferring Files Migrated by the Hierarchic Storage Manager
Files migrated by the hierarchic storage manager are not recalled before dynamic allocation. When files are migrated to tape, parallel-running file transfers can be delayed because the server cannot dynamically allocate the next file before the previous allocation is complete.
Data Compression

To reduce the time needed to transfer a file, you can have the transfer program at the sending system compress the file before sending it, and the transfer program at the receiving system decompress the file after receiving it. You can select which of the following compression methods the transfer programs are to use:

- SNA compression and SNA compaction
- Adaptive compression
- No compression.

SNA compression works by replacing strings of identical characters with bytes that tell the receiving transfer program which character is repeated and how many times it is repeated. SNA compaction works by packing two characters into a byte that normally holds only one. (NetView FTP V2.2.1 MVS uses SNA compaction for numeric characters only.)

Adaptive compression works by replacing character strings that are repeated in a file with references to a directory of such character strings.

Which compression method is the most effective depends on the nature of the data in the file being transferred.

Note: You cannot use data compression for OSI file transfers.

Character Data Conversion

In file transfers between an EBCDIC based system, such as MVS and an ASCII based system as OS/2 or AIX, it can be necessary to convert character data. If you, for example, send an ASCII file to an MVS system, the data can be stored there, but it cannot be processed without having been converted. This conversion process is called character data conversion (CDC). For CDC, a coded character set identifier (CCSID) is needed, which represents a specific encoding scheme.

CDC is usually done at the receiving system. If, however, the receiving system does not support this function actively, the sending system can convert the data.

NetView FTP V2.2.1 MVS provides only passive support for this function. So, when you request CDC in file transfers between NetView FTP V2.2.1 MVS and NetView FTP for Workstations, NetView FTP for Workstations does the data conversion.

Automatically Creating Names for New Receiving Files

If a user does not specify a name for a new receiving file, NetView FTP V2.2.1 MVS uses information contained in the request and obtained from the system in which it runs to create one. For more information about how NetView FTP V2.2.1 MVS creates names for receiving files refer to the NetView FTP Parameter Reference.

Note: For more information about how names are created for OSI file transfers, refer to the OSI/File Services System/370 User's Guide.

---

1 NetView FTP/400 only uses adaptive compression. NetView FTP for Workstations uses adaptive compression or no compression.
Allocating and Creating Files

In a request, you can specify values for file parameters. NetView FTP uses these values to allocate the sending and receiving files. Examples of such parameters are: data set name, DCB parameters, and file access parameters. If you do not specify values for these parameters, NetView FTP, wherever possible, substitutes values that correspond to the characteristics of the sending file.

Before a file transfer can take place, the sending server must allocate the sending file and the receiving server must allocate the receiving file. A server allocates its file just before the file transfer takes place, and uses allocation parameters specified in the request. This is called dynamic allocation.

However, sometimes you might want a server to allocate a file using allocation parameters that cannot be specified in a NetView FTP V2.2.1 MVS file-transfer request. You can do this by specifying the allocation parameters for that file in the DD statements in a server’s startup job and then arranging for that server to carry out the file transfer. The server allocates the file as soon as the server is started, so that, when the time comes to transfer the file, the file is already allocated. This is called job allocation.

Note: To use job allocation, you must be authorized to create or modify server startup jobs. How to do this is described in the NetView FTP V2 MVS Installation, Operation, and Administration guide.

If, when allocating a sending or receiving file, you want a server to use:

- Dynamic allocation, you must specify, in the request, the name of the file.
- Job allocation, you must specify, in the request, the name of the DD statement in the server (MVS) or the DLBL statement in the NetView FTP partition (VSE) startup job that specifies the name of the file.

If, in a request, you specify both a file ID and the name of a DD or DLBL statement, the server uses job allocation, but only carries out the file transfer if the name of the file specified in the request is the same as the name specified in the DD or DLBL statement.

When you use job allocation, you must make sure that the server that allocates a file is also the server that is chosen to participate in the file transfer. You can do this by specifying the LU name of the server in the request, or, if the server is at your system, by specifying, in the request, a server class for which only that server is started.

NetView FTP can allocate the receiving file if it does not already exist at the receiving system. If the file is a VSAM cluster, you can specify which of the following NetView FTP is to do:

- Use an existing cluster at the receiving system as a model for the new cluster
- Use attributes of the cluster being sent for the new cluster or, if the sending file is not a VSAM cluster, use default values to define and allocate the new cluster.

You can also specify values for many of the AMS parameters used to define new clusters. The values you specify override the values AMS would otherwise use.

Note: For OSI file transfers, file allocation is handled by OSI/File Services.
Specifying File Organizations
If the file being sent already exists and is at your system, you need not specify its organization—NetView FTP V2.2.1 MVS retrieves this information automatically. However, sometimes you might have to specify a file type explicitly, for example:

- When the file does not exist at the time you create a request to transfer it.
- When the file is migrated by the hierarchic storage manager.

Checking Requests at the Remote System
You can have your requests checked by the NetView FTP at the remote system before transferring the file. In this way, you can be warned about inconsistencies in your requests before the file is transferred. This is particularly useful when you specify a not-before time or date.

The contents of the request are sent to, and validated at, the remote system immediately, regardless of whether the request has HOLD status, you have specified a not-before time or date, or a server with the specified class is available. However, the file transfer is not performed at this time. The results of the check are provided in the log file of the server that processed the remote check. If you have specified your user ID as the report recipient, you are also sent a remote check completion message and the remote check report.

If the remote request check completed without any errors, the request is scheduled for file transfer processing according to the criteria that you specified. Otherwise, the request status is set to FINISHED. In the case of errors being found, the return and reason codes indicate that the remote request check found errors.

Logging and Reporting
NetView FTP V2.2.1 MVS provides the following to help satisfy your logging and reporting needs:

- File-transfer reports and file-transfer completion messages
- File-transfer return and reason codes
- Log files for the batch job interface routine
- Queue handler log file
- Server log files
- System Management Facility (SMF) records
- Messages for NetView CLISTs.

OSI file transfers use the same logging and reporting facilities as SNA file transfers except that OSI servers do not issue messages for NetView CLISTs.

File-Transfer Reports and File-Transfer Completion Messages
After every file transfer, whether successful or not, the servers at the sending and receiving locations record such things as the outcome of the file transfer, the transfer start and stop times, and the names of the sending and receiving files. In a request, you can specify the user IDs where the servers send this information.

---

2 Available only at the location that initiated the request.
At NetView FTP systems, each server then places this information into a file called a *file-transfer report* and uses its operating system's network job entry (NJE) facility (JES, VSE/POWER PNET, or RSCS) to send the report to the user you specify. An example of a file-transfer report is shown in Figure 6.

A server that sends a user a file-transfer report also sends that user a short message, called a *file-transfer completion message*, which summarizes the outcome of the file transfer. A NetView FTP V2.2.1 MVS server can send completion messages only to users at the same JES node.

**File-Transfer Return and Reason Codes**

File-transfer return and reason codes are included in several messages issued by NetView FTP. The file-transfer return and reason codes are described in the *NetView FTP Messages and Codes* manual. Do not confuse file-transfer return and reason codes, which describe the outcome of attempts to transfer files, with the queue handler command return and reason codes, which describe the outcome of attempts to do such things as add requests to the queue or query requests.

If the return or reason code of a waiting or active request is greater than 0, the request was interrupted. If the status of the request is waiting, it is waiting to be automatically restarted. If the status of the request is active, it has been automatically restarted.
Log Files for the Batch Job Interface Routine

All messages issued by the batch job interface routine are written into a log file. You specify the name of this log file in your NetView FTP V2.2.1 MVS batch job.

Queue Handler and Server Log Files

The queue handler and each server logs information about its processing in its own log file. These log files are allocated when the components are started. The contents of each server log file include the messages that make up the corresponding server’s file-transfer report. For more information about the queue handler and server log files see the NetView FTP V2 MVS Installation, Operation, and Administration guide.

System Management Facility Records

NetView FTP V2.2.1 MVS can create system management facility (SMF) records, which can be added to the SMF file at your system. You can specify the SMF record type of these records. Other application programs can use the information contained in the records to do such things as produce accounting statistics and measure performance. For more information about the SMF records see the NetView FTP V2 MVS Installation, Operation, and Administration guide.

Messages for NetView Command Lists (CLISTs)

NetView FTP V2.2.1 MVS can issue messages about events that occur during its processing in such a way that NetView CLISTs can interpret their significance and take appropriate action. This lets you use NetView CLISTs to automate the operation of NetView FTP. Your system administrator can specify for which events a NetView FTP server or queue handler component is to issue such messages.

Security

NetView FTP has several facilities to help ensure the security of the data it transfers.

Providing Security Information in a Request

In a request, you can include the following security information:

- VSAM passwords:
  - Cluster password
  - Catalog password
  - Model password.

  These passwords, however, are in effect only if no security product is installed and active.

- Access security parameters:

  For a file at a VSE system you can specify the Access Control parameters:
  - User ID
  - Password.

  At an MVS or VM location, the operating system provides a generic interface to security products, the System Authorization Facility (SAF). An installation has the possibility to run a security program, such as RACF.
You can specify the security parameters:
- User ID
- Password
- Group ID.

- VM link parameters for a file at a VM location:
  - Disk user ID
  - Disk address
  - Link password.

- APPC conversation security parameters:
  - User ID
  - Password.

Both the requesting and responding transfer programs can use this information to gain access to protected files without compromising the security of the files.

Automatically Retrieving Values for Access Security Parameters
NetView FTP V2.2.1 MVS can use the installed SAF program product, such as RACF, to automatically retrieve information that it needs to gain access to a file. NetView FTP V2.2.1 MVS can retrieve the password and the group ID that correspond to the user ID of the request originator. This relieves users of having to specify such information.

Note: For OSI file transfers, NetView FTP V2.2.1 MVS checks with RACF whether the specified user ID is allowed to access the specified file.

Data Encryption
NetView FTP V2.2.1 MVS can use Programmed Cryptographic Facility (PCF) to encrypt data before sending it and decrypt data after receiving it. Using data encryption causes the data in a file to remain unrecognizable as it travels through the network.

Note: Data encryption is only supported in transfers between NetView FTP V2 MVS systems.

Error Recovery
Telecommunication lines are prone to interruptions. However, NetView FTP V2.2.1 MVS has a method for reducing the effect of line interruptions on your file transfers. For OSI file transfers, recovery from transmission errors and interrupted file transfer requests is handled by OSI/File Services.

Taking Checkpoints
Periodically during a file transfer, a receiving server can record how much of a file has been transferred successfully—this is called taking a checkpoint. If the file transfer is interrupted and later resumed, the receiving server can tell the sending server where in the file it last took a checkpoint. The sending server can then continue from the last checkpoint instead of transferring the whole file again from the beginning.

No checkpoints are taken by NetView FTP V2.2.1 MVS during transfers using OSI protocols.
Transfer Restart

When you create a request, you can specify that, if one of the following situations arises, NetView FTP is to change the status of the request back to waiting:

- One of the following at either location is terminated:
  - The queue handler
  - The server
  - The NetView FTP partition (VSE only)
  - The partner FTAM application (OSI file transfers only).

- NetView FTP Server is stopped.
- NetView FTP Client is not available.
- The server at either location temporarily cannot allocate the file to be transferred.
- A pre-transfer user-exit routine at either location temporarily rejects the file transfer.
- There is a prolonged conversation interruption.

The servers at your system will then periodically reprocess the request until one of them succeeds in restarting the transfer. This is called using automatic transfer restart. When a server restarts the transfer, it resumes transferring the file from either the last recorded checkpoint or the beginning of the sending file. You can specify the restart point in your request.

If NetView FTP V2.2.1 MVS is unable to restart a transfer automatically (or if you specify in your request that you do not want it to try), the queue handler changes the status of the corresponding request to finished and does not reprocess it. However, you can request a restart of this file transfer with the restart function. You must identify the request by its request number. This is called using manual transfer restart. If you use manual restart, you can specify in the request whether NetView FTP V2.2.1 MVS is to continue the transfer from the last recorded checkpoint or from the beginning of the file.
Chapter 2. Using the NetView FTP V2.2.1 MVS Panels

NetView FTP V2.2.1 MVS has a set of Interactive System Productivity Facility (ISPF) panels that make it easy for you to create, save, change, submit, query, delete, modify, restart, hold, and release requests. You use the panels by entering data in the input fields.

The Panel Layout

The way in which the text and the input fields on a panel are arranged is called the panel layout. The layouts of the NetView FTP V2.2.1 MVS panels are all similar. Figure 7 shows the layout of a typical panel. The shaded areas represent input or output fields.

![Figure 7. Layout of a Typical NetView FTP V2.2.1 MVS Panel](image)

Centered at the top of each panel is the panel title.

Near the upper left-hand corner of each panel is the word Command, followed by an arrow (==>). To the right of the arrow is an input field called the command line. On the command line, you can type in an ISPF primary command. For example, you can tell ISPF to display the settings of the PF keys by entering the command PFSHOW on the command line. You can also enter ISPF primary commands by using the program function (PF) keys.

The line just below the command line is called the message line. Any messages from NetView FTP V2.2.1 MVS, such as messages telling you that you have entered incorrect input, are displayed there.
Some of the NetView FTP V2.2.1 MVS panels are *table display panels*, that is, panels that contain scrollable lists. The upper right-hand corner of a table display panel contains the current date and time and the panel’s *scroll amount*. The scroll amount is the amount that the list on a panel is scrolled up or down when you press the **UP** or **DOWN** PF key or enter the **UP** or **DOWN** command.

The large area in the middle of each panel can contain any combination of the following:

- A list of numbered options for you to select from
- A series of input fields for you to fill
- A scrollable list of items, next to which you can type an action.

You select an option from a list of numbered options by typing, in the option field, the number that corresponds to the option you want and then pressing the ENTER key. Each input field holds a fixed number of characters. To the right of some input fields is, in parentheses, a list of the values you can enter in those fields. To the right of other input fields is a short description of the values you can enter in those fields.

### Entering Data

You use the panels by entering values in the panels’ option field and input fields. To enter a value, type it in the appropriate field and then press the ENTER key. Use the -->|, |<--, and <-' keys to move from field to field.

Sometimes NetView FTP V2.2.1 MVS fills an input field with a value that it will use unless you change it to something else. Such a value is called a *default value*. To change a default value, type over it.

After entering values in the fields, press the ENTER key. NetView FTP V2.2.1 MVS checks whether you have entered data in all fields that require it, and, for some fields, whether the data you entered is valid. If NetView FTP V2.2.1 MVS detects missing or incorrect data, it displays a message telling you so and moves the cursor to the beginning of the corresponding field. When you have entered all data completely and correctly, NetView FTP V2.2.1 MVS displays the next panel in the series.
Invoking the NetView FTP V2.2.1 MVS Interactive Interface

You invoke the interactive interface in either of the following ways:

- If your system programmer has added a NetView FTP V2.2.1 MVS entry to one of the selection menus for TSO application programs, invoke NetView FTP V2.2.1 MVS by selecting that entry from the menu.
- Entering one of the following commands in TSO READY mode:

```
ISPSTART PGM(type) [PARM(langparm)]
```

**Note:** The left and right square brackets indicate optional parameters and are not included in the command. *type* can be one of the following:

<table>
<thead>
<tr>
<th>type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGIFII</td>
<td>Use all the functions of the interactive interface.</td>
</tr>
<tr>
<td>DVGIFIQ</td>
<td>Use only the control functions.</td>
</tr>
<tr>
<td>DVGIFIIQA</td>
<td>Use only the administrator functions.</td>
</tr>
</tbody>
</table>

`lang` indicates which message module NetView FTP V2.2.1 MVS uses. It can be one of the following:

<table>
<thead>
<tr>
<th>lang</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>English</td>
</tr>
<tr>
<td>F</td>
<td>French</td>
</tr>
<tr>
<td>G</td>
<td>German</td>
</tr>
<tr>
<td>I</td>
<td>Italian</td>
</tr>
<tr>
<td>J</td>
<td>Japanese</td>
</tr>
<tr>
<td>P</td>
<td>Portuguese</td>
</tr>
<tr>
<td>S</td>
<td>Spanish</td>
</tr>
<tr>
<td>U</td>
<td>A language requiring single-byte encoding</td>
</tr>
<tr>
<td>D</td>
<td>A language requiring double-byte encoding</td>
</tr>
</tbody>
</table>

**Notes:**

1. The support of a specific language in this parameter does not necessarily mean that the associated message definition module is available from IBM.

2. The NetView FTP V2.2.1 MVS libraries must be concatenated as described in the *NetView FTP V2 MVS Installation, Operation, and Administration* manual.
After you invoke NetView FTP V2.2.1 MVS, NetView FTP V2.2.1 MVS displays its Main Task Selection panel (Figure 8).

You use this panel to tell NetView FTP V2.2.1 MVS which task you want to perform.

**Getting Help from NetView FTP V2.2.1 MVS**

You can get information from NetView FTP V2.2.1 MVS about each of its panels. To ask for help about the contents of a particular panel, press the HELP PF key (or enter the HELP command) when that panel is displayed. NetView FTP V2.2.1 MVS displays a help panel that describes the contents of the NetView FTP V2.2.1 MVS panel. Sometimes you are presented with a menu from which you can select a topic. In this case, make your selection and press the ENTER key. Press the END PF key (or enter the END command) to return to the NetView FTP V2.2.1 MVS panel.

If you ask for help while NetView FTP V2.2.1 MVS is displaying the NetView FTP Main Task Selection panel, NetView FTP V2.2.1 MVS displays the NetView FTP V2.2.1 MVS Tutorial panel. From this panel, you can select topics about which you want to know more.
Creating File-Transfer Requests

Each file-transfer request is made up of parameters, to which you assign values. The values you assign to the parameters that make up a request tell NetView FTP V2.2.1 MVS the things it needs to know to be able to transfer a file. In each request you create, there are some parameters for which you must specify values, some for which you can specify values, and some for which you must not specify values.

To create an SNA request, select ‘Create SNA’ by typing 1 in the option field of the NetView FTP Main Task Selection panel (Figure 8 on page 30), and then press the ENTER key. NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel (Figure 9).

Figure 9. NetView FTP Main Transfer Parameters (DVGME0AE)
To create an OSI request, select ‘Create OSI’ by typing 5 in the option field of the NetView FTP Main Task Selection panel (Figure 8 on page 30), and then press the ENTER key. NetView FTP V2.2.1 MVS displays the OSI Transfer Parameters panel (Figure 10).

![NetView FTP OSI Transfer Parameters](DVGME01E)

To find out which parameters to specify, refer to the NetView FTP Parameter Reference. You can also find examples of using the panels in Chapter 6, “Examples of Using the NetView FTP V2.2.1 MVS Panels” on page 95.

Specify the parameters that you or NetView FTP V2.2.1 MVS require and press the ENTER key. Depending on the values you specify, NetView FTP V2.2.1 MVS displays one of several panels. Continue to fill in the panels and press the ENTER key until NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 11 on page 33).
Selecting a Current Request Task

After you specify values for the parameters that require them, NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel which is shown in Figure 11.

The request you are working on at any given time is called the current request. It is made up of the values of all the parameters on all the input panels at that time. You use the Current Request Task Selection panel to:

- Submit the current request
- Change the current request
- Save the current request
- View a summary of the request parameters specified.

![Current Request Task Selection Panel](image)

**Figure 11. Current Request Task Selection (DVGME0KE)**

Submitting the Current Request

To submit the current request, type 1 in the option field of the Current Request Task Selection panel (Figure 11), and then press the ENTER key. Submitting a request tells NetView FTP V2.2.1 MVS to add the request to the request queue. NetView FTP V2.2.1 MVS adds the request to the request queue and displays a message telling you the request number it assigned to the request.

If a request contains parameter values that conflict with one another, NetView FTP V2.2.1 MVS does not add the request to the request queue but issues an error message. If this happens, check the summary panels for incompatible parameter values.
Changing the Current Request
To change the current request, type 2 in the option field of the Current Request Task Selection panel (Figure 11 on page 33), and then press the ENTER key. NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel (Figure 9 on page 31). All the input fields of that panel and of all the other panels are filled in with the values contained in the current request. You can change any of the values on any of the panels, exactly as if you were creating the request from the beginning. When NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 11 on page 33), you can select any of the options it lists.

The Change option is especially helpful when you want to create several requests that are very similar (perhaps only one or two values are different), and you do not want to create each request from the beginning.

Saving the Current Request
If you need to, you can save the current request for future use. If, sometime later, you do not need a saved request anymore, you can discard it. How to do these things is explained in “Listing Saved Requests: Submitting, Changing, Discarding, Viewing” on page 37.

To save the current request, type 3 in the option field of the Current Request Task Selection panel and then press the ENTER key. NetView FTP V2.2.1 MVS displays a panel with two input fields.

In the first field, type the name you want to give the request to be saved. The name can be up to eight characters long. If you specify a name that is the same as that of another of your saved requests, NetView FTP V2.2.1 MVS replaces the existing version with the new version.

In the second field, type a short (up to 28 character) description of the request to be saved and press the ENTER key. NetView FTP V2.2.1 MVS saves the request, displays the Current Request Task Selection panel (Figure 11 on page 33), and awaits your next selection. Later, when you ask NetView FTP V2.2.1 MVS to show you a list of your saved requests, this short description appears in that list.

Viewing the Current Request Summary
To view the current request, type 4 in the option field of the Current Request Task Selection panel, and then press the ENTER key. NetView FTP V2.2.1 MVS displays panels summarizing the parameter values specified for that request.
Transferring PDS Members

There are several ways to transfer members of a PDS using NetView FTP V2.2.1 MVS. These ways are described here and demonstrated in “Example 3: Sending Members of a PDS” on page 108.

Transferring a Single Member

Specify the name of a member of the PDS in parentheses after the data set name. For example: 'partit.ds.name(memname)'. NetView FTP V2.2.1 MVS transfers that member without its directory information. The type of the other data set need not be PO. If the sending and receiving data sets are both dynamically allocated PDSs and if you do not specify a name for the receiving member, NetView FTP V2.2.1 MVS makes the name of the receiving member the same as that of the sending member.

Transferring an Entire PDS

Specify an * in parentheses after the name of the sending data set. For example: 'partit.ds.name(*)'. NetView FTP V2.2.1 MVS transfers the entire PDS specified in the data set name along with the directory information of each of its members. You can specify this value only if both of the following are true:

- The transfer program at both locations is NetView FTP V2 MVS.
- The type of both the sending and receiving data sets is PO.

The value of the PDS Option parameter tells NetView FTP V2.2.1 MVS under which circumstances it is to transfer each of the members of the sending data set.

The PDS Option parameter is described in the NetView FTP Parameter Reference.

Transferring Members Selected from a List

To select or exclude members, do not specify parentheses after the name of the sending data set. For example: 'partit.ds.name'. NetView FTP V2.2.1 MVS displays the PDS List Type panel (Figure 12):

```
DVGME1PE ----------------------------------- PDS List Type -----------------------------------
Command  >>>

Select one of the following options:

>>> 1 Selection  - Displays list from which you can select the members you want to transfer

2 Exclusion  - Displays list from which you can select the members you do not want to transfer

PF01-HELP   PF03-END   PF04-RETURN
```

Figure 12. PDS List Type (DVGME1PE)
You can select from a list of members those members to be transferred, this is called selecting members. You can also select from a list of members those members that are not to be transferred, this is called excluding members.

However, you can do this only if all of the following is true:

- The transfer program at both locations is NetView FTP V2 MVS.
- The sending data set is at the local (your) location.
- The type of both the sending and receiving data sets is PO.

If the sending data set is at the remote location and you specify the data set name without parentheses, the entire data set is transferred.

Selecting Members
Select option 1 from the PDS List Type panel (Figure 12 on page 35). NetView FTP V2.2.1 MVS displays a list of all the members in the sending data set. An example of such a list is shown in Figure 75 on page 111. Type the first letter of a PDS option in the Act column next to each member you want NetView FTP V2.2.1 MVS to transfer. The PDS options are INSERT, UPDATE, and REPLACE. The letter you type tells NetView FTP V2.2.1 MVS under which circumstances it is to transfer each member to the PDS at the receiving system:

- **i** Transfer the member only if a member with the same name does not already exist in the receiving PDS.
- **u** Transfer the member only if a member with the same name already exists in the receiving PDS. If one of the following is true, the member is created and allocated, but no data is transferred and the new member remains empty:
  - The value of the File Status Option is MUSTNOTEXIST and the value of the End-of-Processing Option is KEEP.
  - The value of the File Status Option is MUSTNOTEXIST and the value of the End-of-Processing Option is CATLG.
- **r** Transfer the member regardless of whether a member with the same name already exists in the receiving PDS.

If you transfer several members at once, they need not all have the same PDS option.

You can also specify a new name for any member you select. That name is given to the member at the receiving system, provided the PDS option allows the transfer of that member to take place. If you specify a new name:

- The PDS option applies to that name and not to the old name.
- The first character must be alphabetic, the remaining characters (up to 7) can be A to Z, 0 to 9, @, $, and #.
Excluding Members

Select Option 2 from the PDS List Type panel (Figure 12 on page 35). NetView FTP V2.2.1 MVS displays a list of all the members in the sending data set. An example of such a list is shown in Figure 78 on page 114. Type an x (for exclude) in the Act column next to each member you do not want NetView FTP V2.2.1 MVS to transfer. NetView FTP V2.2.1 MVS transfers all the other members that satisfy the condition of the PDS option you specified.

If NetView FTP V2.2.1 MVS transfers one or more members specified either by selection or exclusion, it also transfers the directory information of those members. NetView FTP V2.2.1 MVS does not transfer any member of a PDS that is marked as being unmovable.

When NetView FTP V2.2.1 MVS is to transfer a member whose name is not an alias and one or more of that member's aliases, NetView FTP V2.2.1 MVS transfers the member and adds the aliases to the directory of the receiving PDS. When NetView FTP V2.2.1 MVS is to transfer only aliases of a member, NetView FTP V2.2.1 MVS transfers the data contained in the member to the receiving system, which assigns the data to the first alias and adds the other aliases to the directory of the receiving PDS.

If a PDS option prevents NetView FTP V2.2.1 MVS from transferring any one member of a group consisting of one member and its aliases, NetView FTP V2.2.1 MVS does not transfer that member or any other member of the group. “Example 3: Sending Members of a PDS” on page 108 provides an example of transferring PDS members.

Listing Saved Requests: Submitting, Changing, Discarding, Viewing

You can submit, change, discard, or view a summary of a saved request by selecting the request from the Saved Requests panel. You can also use the Saved Requests panel to view a summary of the parameters specified for the request. Figure 13 shows a typical Saved Requests panel.

You can ask NetView FTP V2.2.1 MVS to display a list of the requests you have created and saved. By typing a command next to a request on this list, you can tell NetView FTP V2.2.1 MVS to:

- Submit the request to the request queue
- Display that request so that you can change it
- Discard that request from the list of saved requests
- View a summary of the parameters specified for that request.
To display a list of all saved requests, first return to the NetView FTP Main Task Selection panel (Figure 8 on page 30), type 2 and then press the ENTER key. NetView FTP V2.2.1 MVS displays a Saved Requests panel similar to the one shown in Figure 13.

![Figure 13. Saved Requests (DVGME0RE)](image)

The requests on the Saved Requests panel are listed in alphabetical order. The panel displays the following information about each request:

- Request name
- Transfer mode
- Remote name (remote server group, remote LU name, or Application Entity Title)
- Request description.

A list of saved requests might be too long to fit on your screen. If so, you can use the UP and DOWN PF keys (or commands) to scroll through it. You can locate a request by entering I and the request name on the command line. You can use the asterisk as a global file-name character when specifying the request name, for example, STOCK*.

You can submit, change, discard, or view any of the requests in the list by typing the first letter of the appropriate action in the Act column next to that request. However, you cannot enter different commands simultaneously (for example, you cannot submit the first three requests in the list and delete the fourth and fifth at the same time).

To submit one or more saved requests to the request queue, type s in the Act column next to the requests you want to submit and then press the ENTER key. NetView FTP V2.2.1 MVS displays a message that explains the outcome of the adding of the requests to the queue, displays the Saved Requests panel, and awaits your next selection.
To change a saved request, type \texttt{c} in the \texttt{Act} column next to the request you want to change. NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel. The input fields are filled in with the values that were specified for that request. You can change any of the values, then move to any of the other panels exactly as you would if you were creating the request from the beginning. The input fields of those panels are also filled with the values that were specified earlier. Again, you can type over them, and thereby change the request. When NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel, you can select any of the options it lists.

To discard one or more saved requests, type \texttt{d} in the \texttt{Act} column next to the requests you want to discard, and then press the ENTER key. NetView FTP V2.2.1 MVS discards the requests from the ISPF Table, displays the Saved Requests panel, and awaits your next selection.

To view a summary of the request parameters specified, type \texttt{v} in the \texttt{Act} column next to the request you want to view, and then press the ENTER key. NetView FTP V2.2.1 MVS displays panels summarizing the parameters specified.

**Controlling Submitted Requests**

You can ask NetView FTP V2.2.1 MVS to show you a list of the requests you submitted. This is called \textit{controlling} your submitted requests. You can:

- Query any of your requests in the request queue
- Delete any of your requests from the request queue
- Restart interrupted file transfers from the request queue
- Modify the class or priority of any of the requests that you submitted and that have the status \texttt{waiting}
- Hold any of the requests that you submitted and that have the status \texttt{waiting}
- Release any of the requests that you submitted and that are held.

If you know the NetView FTP V2.2.1 MVS master password, you can also query, delete, restart, modify, hold, or release requests submitted by other NetView FTP V2.2.1 MVS users.
To control submitted requests, return to the NetView FTP Main Task Selection panel (Figure 8 on page 30), and select Option 3 (Control). NetView FTP V2.2.1 MVS displays the Request Control panel, which is shown in Figure 14.

```
DVGMEQAE ---------------------- Request Control -------------------------------
Command ===>  
Select an option by number
1  Query all - Query all requests starting with the specified number and with the specified originator
2  Query - Query the request with the specified number
3  Delete all - Delete all requests of one originator
4  Delete - Delete the request with the specified number
5  Delete finished - Delete all finished requests of one originator
6  Modify - Modify class or priority, or hold or release
7  Force - Force the request with the specified number
8  Restart - Restart the request with the specified number
Request Number ....... ===>  
Request Originator .... ===>  
Request Password ...... ===>  
Hold or Release ....... ===>  (H=Hold,R=Release)
Server Class ........... ===>  (0 to 9; A to Z)
Request Priority ...... ===>  (0 to 9)
Restart Point ......... ===>  (B=Begin,C=Checkpoint)
PF01=HELP PF03=END PF04=RETURN
```

Figure 14. Request Control (DVGMEQAE)

You can delete, restart, modify, hold, and release requests in one of two ways:

- By specifying, on the Request Control panel, the number of a request
- By querying any number of requests, and after the Submitted Requests panel is displayed (Figure 20 on page 45), selecting requests from it.

You use the Request Control panel by selecting one of the options and, if necessary, typing additional information in the input fields below the list of options.

Figure 15, Figure 16, and Figure 17 on page 43 show the ways you can query, delete, or restart submitted requests, and the information you need to give NetView FTP V2.2.1 MVS.
<table>
<thead>
<tr>
<th>Which Requests Do You Want to Query</th>
<th>Option Number</th>
<th>Request Number Field</th>
<th>Priority Field</th>
<th>Server Class Field</th>
<th>Request Originator Field</th>
<th>Request Password Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>All requests that you submitted.</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>that have numbers greater than or</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>equal to n.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and that have the priority p.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td>n</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>that have numbers greater than or</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>equal to n, and that have the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority p.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and that have the server class c.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>that have numbers greater than or</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>equal to n, and that have the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>server class c.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>that have the priority p and the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>server class c.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted</td>
<td></td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>that have numbers greater than or</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>equal to n, the priority p, and the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>server class c.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>another originator.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>another originator and that have</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>numbers greater than or equal to n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>another originator and that have</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the priority p.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>another originator, that have</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>numbers greater than or equal to n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and the priority p.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Which Requests Do You Want to Query</td>
<td>Option Number</td>
<td>Request Number Field</td>
<td>Priority Field</td>
<td>Server Class Field</td>
<td>Request Originator Field</td>
<td>Request Password Field</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>----------------------</td>
<td>----------------</td>
<td>--------------------</td>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>All requests that were submitted by another originator and that have the server class c.</td>
<td>1</td>
<td></td>
<td>c</td>
<td>originator ID</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by another originator, that have numbers greater than or equal to n, and the server class c.</td>
<td>1</td>
<td>n</td>
<td>c</td>
<td>originator ID</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by another originator and that have the priority p and the server class c.</td>
<td>1</td>
<td>p</td>
<td>c</td>
<td>originator ID</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by another originator, that have numbers greater than or equal to n, the priority p, and the server class c.</td>
<td>1</td>
<td>n</td>
<td>p</td>
<td>c</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests in the request queue.</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests in the request queue that have the priority p.</td>
<td>1</td>
<td>p</td>
<td></td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests in the request queue that have the server class c.</td>
<td>1</td>
<td></td>
<td>c</td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests in the request queue that have the priority p and the server class c.</td>
<td>1</td>
<td>p</td>
<td>c</td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that have numbers greater than or equal to n.</td>
<td>1</td>
<td>n</td>
<td></td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that have numbers greater than or equal to n and that have the priority p.</td>
<td>1</td>
<td>n</td>
<td>p</td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that have numbers greater than or equal to n and that have the server class c.</td>
<td>1</td>
<td>n</td>
<td>c</td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that have numbers greater than or equal to n, the priority p, and the server class c.</td>
<td>1</td>
<td>n</td>
<td>p</td>
<td>c</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>The request that you submitted that has the number n.</td>
<td>2</td>
<td>n</td>
<td></td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>A request that you did not submit that has the number n.</td>
<td>2</td>
<td>n</td>
<td></td>
<td></td>
<td>master password</td>
<td></td>
</tr>
</tbody>
</table>
### Which Requests Do You Want to Delete

<table>
<thead>
<tr>
<th>Option Number</th>
<th>Request Number Field</th>
<th>Request Originator Field</th>
<th>Request Password Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>password assigned to the requests when they were created, or master password</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>originator ID</td>
<td>master password</td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>password assigned to the request when it was created, or master password</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>originator ID</td>
<td>master password</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td></td>
<td>password assigned to the request when it was created, or master password</td>
</tr>
<tr>
<td>8</td>
<td>n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 17. Restarting Requests (NetView FTP V2.2.1 MVS Panels)

<table>
<thead>
<tr>
<th>Which Request Do You Want to Restart</th>
<th>Option Number</th>
<th>Request Number Field</th>
<th>Request Originator Field</th>
<th>Request Password Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request that you submitted, that is not password-protected, and that has the number ( n ).</td>
<td>7</td>
<td>( n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number ( n ).</td>
<td>7</td>
<td>( n )</td>
<td></td>
<td>password assigned to the request when it was created, or master password</td>
</tr>
<tr>
<td>The request that another originator submitted that has the number ( n ) (whether password-protected or not).</td>
<td>7</td>
<td>( n )</td>
<td>originator ID</td>
<td>master password</td>
</tr>
</tbody>
</table>

Figure 18 and Figure 19 on page 45 show how to modify the class or priority of a request, and how to hold or release a request or an entire class.

Figure 18. Modifying Submitted Requests (NetView FTP V2.2.1 MVS Panels)

<table>
<thead>
<tr>
<th>For Which Requests Do You Want to Modify The Class or Priority</th>
<th>Option Number</th>
<th>Request Number Field</th>
<th>Request Password Field</th>
<th>Server Class or Priority Field</th>
<th>Hold or Release Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request that you submitted, that is not password-protected, and that has the number ( n ).</td>
<td>6</td>
<td>( n )</td>
<td></td>
<td>new class or priority</td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number ( n ).</td>
<td>6</td>
<td>( n )</td>
<td>password assigned to the request when it was created, or master password</td>
<td>new class or priority</td>
<td></td>
</tr>
<tr>
<td>A request that another originator submitted that has the number ( n ) (whether password-protected or not).</td>
<td>6</td>
<td>( n )</td>
<td>master password</td>
<td>new class or priority</td>
<td></td>
</tr>
</tbody>
</table>
Figure 19. Holding and Releasing Submitted Requests (NetView FTP V2.2.1 MVS Panels)

<table>
<thead>
<tr>
<th>Which Requests Do You Want to Hold or Release</th>
<th>Option Number</th>
<th>Request Number Field</th>
<th>Request Password Field</th>
<th>Server Class or Priority Field</th>
<th>Hold or Release Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request that you submitted, that is not password-protected, and that has the number ( n ).</td>
<td>6</td>
<td>( n )</td>
<td></td>
<td></td>
<td>( H ) to hold; ( R ) to release</td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number ( n ).</td>
<td>6</td>
<td>( n )</td>
<td>password assigned to the request when it was created, or master password</td>
<td></td>
<td>( H ) to hold; ( R ) to release</td>
</tr>
<tr>
<td>The request that another originator submitted that has the number ( n ) (whether password-protected or not).</td>
<td>6</td>
<td>( n )</td>
<td>master password</td>
<td>in the Server Class field, the number or letter of the class to be held or released</td>
<td>( H ) to hold; ( R ) to release</td>
</tr>
<tr>
<td>An entire server class</td>
<td>6</td>
<td></td>
<td>master password</td>
<td></td>
<td>( H ) to hold; ( R ) to release</td>
</tr>
</tbody>
</table>

If you select Option 1 (Query All) or Option 2 (Query), NetView FTP V2.2.1 MVS displays a Submitted Requests panel similar to the one shown in Figure 20.

```
DVGMEQBE -------------------- Submitted Requests --------------------
Command ===> SCROLL ===>
PF01=HELP PF03=END PF04=RETURN PF05=UP PF06=DOWN
Commands: locate refresh sortn sortd sorto 94/10/14 14:31
Actions: Cn to change class to n, Pn to change priority to n, H to hold, R to release, D to delete, F to force delete, TB to restart from begin, TC to restart from checkpoint
Act Numbr Name Cl Pr St Ret Rsn Records Originat Rst Date Time
---------------------------------------------------------
 238 NEWSEND 2 3 W 0 PETER 94/10/14 17:01
 367 CARLPROF 1 3 H 0 CARL 94/10/14 17:13
 412 SWATFILE 1 3 A 121 CARL 94/10/14 17:13
 521 WEDJOB 5 7 F 12 302 30348 JOBNAME1 94/10/14 12:33
 1344 FRJJOB 4 9 F 0 400 JOBNAME2 94/10/14 13:44
 1678 STOCKTOK 2 3 W 0 MARIE 94/10/14 15:55
 1867 MYACCT 2 3 W 0 JERRY 94/10/14 16:01
******************************************************************************
Figure 20. Submitted Requests (DVGMEQBE)
This panel contains a list of submitted requests. You can use this panel to delete any of the requests in the list, or to modify, hold, or release any of the waiting requests in the list. You do this by typing the appropriate action in the Act column to the left of that request’s number. For example:

- To change a waiting request’s class to 3, type **c3** in the Act column next to that request.
- To change a waiting request’s priority to 8, type **p8** next to that request.
- To restart an unsuccessfully finished request from begin, type **tb** next to it.
- To restart an unsuccessfully finished request from checkpoint, type **tc** next to it.
- To delete a request, type **d** next to it.
- To force deletion of a request, type **f** next to it.
- To hold a waiting request, type **h** next to it; to release a waiting request that is held, type **r** next to it.

The Submitted Requests panel contains the following information about each request:

- Request number
- Request name (for saved requests only)
- Server class and priority
- Request status:
  - **W** Waiting
  - **H** Waiting and on hold
  - **D** Waiting and deferred
    - (if this condition persists, contact your system programmer)
  - **B** Waiting and blocked (contact your system programmer)
  - **A** Active
  - **F** Finished

- File-transfer return and reason codes
- Number of records transferred during the last completed transfer attempt, or the number of records transferred during an active transfer (refreshing the panel updates this number). For an OSI file transfer, the character string OSI is displayed in place of the number of records.
- Originator who added the request to the queue
- Restart information (**Y** = restartable request)
- Date and time that:
  - The request was added to the queue (if status = W, H, D, or B)
  - The transfer started (if status = A)
  - The transfer finished (if status = F).

**Note:** For active requests, before the server begins to transfer the corresponding file, the date and time shown is 00/00/00 00:00. Once the transfer starts, the transfer start time is shown.

For more information about return and reason codes refer to “File-Transfer Return and Reason Codes” on page 22.
You can sort the list by request number, by date, or by originator by entering `sortn`, `sortd`, or `sorto`, respectively, on the command line. You can locate a specific request by entering `l` and the corresponding request number on the command line. You can refresh the panel by typing `r` in the command line and pressing the ENTER key.

A list of submitted requests might be too long to fit on your screen. If so, you can use the UP and DOWN PF keys (or commands) to scroll through it. There is a limit to the number of requests a list can contain. If you query more requests than fit in one list, NetView FTP V2.2.1 MVS displays as many as fit, and displays in the message line the number of the first request that did not fit. You can then query all requests with numbers greater than or equal to that number, and NetView FTP V2.2.1 MVS will display a new list.

---

**Administering the Request Queue**

If you know the master password, you can use a special set of panels to display information about the queue, and control the requests in it. To do this, select Option 4 (Administer) from the NetView FTP Main Task Selection panel. NetView FTP V2.2.1 MVS displays the Administration Task Selection panel, which is shown in Figure 21.

---

You use this panel to tell NetView FTP V2.2.1 MVS whether you want to:

- Display statistics about the request queue
- Control all waiting, active, or finished requests.

---

*Figure 21. NetView FTP Administration Task Selection (DVGMEAPE)*
Viewing Request-Queue Statistics

To display statistics about the request queue, select Option 1. NetView FTP V2.2.1 MVS displays the Request Queue Statistics panel, a sample of which is shown in Figure 22.

```
DVGMEAQE ------------------ Request Queue Statistics -------------------------
COMMAND ===> SCROLL ===> PAGE
PF01=HELP PF03=END PF04=RETURN PF07=UP PF08=DOWN
Queue handler name : IQAQHNM Queue handler started: 94/10/14 07:28
Queue size : 199 Current date and time: 94/10/14 16:19
Unused request slots: 129

Class  --------  Requests --------  -----  Class -----  
          Waiting  Active  Finished  Interrupted  Not-served  Held
ALL       21       0       20        2       X
D          6       0       -        -       X
M          4       0       -        -       X
P          8       0       -        -        

*************************************************************************
BOTTOM OF DATA  ****************************************************************************
```

Figure 22. Request Queue Statistics (DVGMEAQE)

This panel displays the following information:

- The name of the queue handler
- When the queue handler was started
- The number of requests in the queue that were formerly active but that were interrupted and are presently waiting to be restarted
- The number of active requests in the queue
- The number of finished requests in the queue
- The number of requests that could still fit in the queue
- The maximum number of requests the queue can hold
- For each class:
  - The number of waiting requests
  - The number of active requests
  - Whether a server is started for that class
  - Whether that class is on hold.

Controlling Waiting, Active, or Finished Requests

If you want to display and to control all waiting, active, or finished requests, select option 2, 3, or 4 in the Administration Task Selection panel (Figure 21 on page 47). Before you press the ENTER key, specify in the fields at the bottom of the panel whether you want NetView FTP V2.2.1 MVS to display:

- The name of the sending file for each request
- The requests of a single class only and, if so, which class.

If you do not need to know the name of the sending file, it is best to specify NO in the Display Data Set Name field. This reduces the amount of time NetView FTP V2.2.1 MVS needs to prepare the statistics panel it will display, and improves your system's overall performance.
A sample of one of these panels, the Waiting Requests panel, is shown in Figure 23.

![Waiting Requests Panel](image)

**Figure 23. Waiting Requests (DVGMEAXE)**

The layouts of the panels for waiting, active, and finished requests are all similar. The Waiting Requests panel contains the following information for each request:

- Request number
- Request originator
- Server class and priority
- Request status
  - W: Waiting
  - H: Waiting and on hold
  - D: Waiting and deferred (if this condition persists, contact your system programmer)
  - B: Waiting and blocked (contact your system programmer)
  - A: Active
  - F: Finished
- If the request was restarted and, if so, if it was automatically (A) or manually (M) restarted
- Date and time the request was submitted
- Server name
- Local LU name or OSI server name
- Remote server group, remote LU name, or Application Entity Title.

A local LU name is displayed for a waiting request when either of the following is true:

- The corresponding request is to be automatically restarted.
- A local LU name or OSI server name was specified in the request.
A server name is displayed for a waiting request when either of the following is true:

- The corresponding request is to be automatically restarted.
- A local LU name or OSI server name was specified in the request and the server with that name has been started.

A request for which a local LU name but no server name is displayed will not be processed until the server with that name is started.

The Active Requests panel contains the following information for each request:

- Request number
- Request originator
- Server class and priority
- Date and time the request was submitted
- Date and time the transfer started
- Server name
- Local LU name or OSI server name
- Remote LU name or application entity title.

**Note:** For active requests, before the server begins to transfer the corresponding file, the date and time shown is 00/00/00 00:00. Once the transfer starts, the transfer start time is shown.

The Finished Requests panel contains the following information for each request:

- Request number
- Request originator
- Server class and priority
- Date and time the request was submitted
- Date and time the transfer started
- Date and time the transfer finished
- Restarted request (yes or no)
- Number of records transferred during the last completed transfer attempt. For an OSI file transfer, the character string OSI is displayed in place of the number of records
- File-transfer return and reason codes.

For more information about return and reason codes refer to “File-Transfer Return and Reason Codes” on page 22.

You can delete any request, restart any unsuccessfully finished request, or modify, hold, or release any waiting request by typing the appropriate action in the Act column to the left of that request’s number. For example:

- To change a waiting request’s class to 3, type c3 in the Act column next to that request.
- To change a waiting request’s priority to 8, type p8 next to that request.
- To restart an unsuccessfully finished request from begin, type tb next to it.
- To restart an unsuccessfully finished request from checkpoint, type tc next to it.
- To delete a request, type d next to it.
- To force deletion of a request, type f next to it.
- To hold a waiting request, type h next to it; to release a request, type r next to it.

To locate a specific request enter l and the corresponding request number on the command line.
If, on the Administration Task Selection panel, you specified YES in the Display Data Set Name field, NetView FTP V2.2.1 MVS replaces the information for each request—except for the request number, originator, and class—with the name of the sending file for SNA file transfers, and the local file name (transfer mode=TO) or the local OSI file name (transfer mode=FROM) for OSI file transfers. To view information not shown on the panel, press the SCROLL RIGHT PF key (or enter the RIGHT command). To redisplay the information that was previously on the panel, press the SCROLL LEFT PF key (or enter the LEFT command).

You can refresh the panel by typing r in the command line and pressing the ENTER key.

Setting Defaults for File Transfer Requests

You can set defaults for several file transfer parameters. To do so, select Option 8 (Defaults) from the NetView FTP Main Task Selection panel. NetView FTP V2.2.1 MVS displays a NetView FTP User Profile panel, which is shown in Figure 24.

```
DVGME00E ------------------ NetView FTP User Profile --------------------------
Command >>>

Following values are used as defaults and may be changed:
Local report recipient:
  User ID
  Node ID
Priority................... (/zerodot,..,9)
Server class............... (A,S,N)
Automatic transfer restart (Y,N)
  Restart point ....... (B,C)
Local Security User ID .... Group ID ............
  Password
Remote Security User ID ... Group ID ............
  Password

Press ENTER to save the values, press PF03 to quit.

PF01=HELP PF03=END PF04=RETURN
```

Figure 24. NetView FTP User Profile (DVGME00E)
Change the values to the defaults you want to have for your file transfer requests and press ENTER to save the values. NetView FTP V2.2.1 MVS displays the further profile panels, which are shown in Figure 25, Figure 26, and Figure 27 on page 53. In these panels you can change the default values of some file allocation and file processing parameters.

**Figure 25. NetView FTP User Profile (DVGME0QE)**

**Figure 26. NetView FTP User Profile (DVGME0NE)**
Converting Saved Requests

Whenever your NetView FTP is reinstalled a new request table is created. You must convert any saved requests that you still want to use and copy them to the new request table. To do so, select Option 9 (Convert) from the NetView FTP Main Task Selection panel. NetView FTP V2.2.1 MVS displays the NetView FTP Convert saved requests panel, which is shown in Figure 28.

Select the type of requests you want to convert and press ENTER. The requests are converted and copied to the new request table.
If the new request table already contains requests, the warning panel DVGMETBE is displayed (see Figure 29):

```
DVGMETBE -------- NetView FTP Convert saved requests - warning ---------------
Command ---->

Warning: The new request table DVGM221S exists already and is not empty.
Press ENTER if you want to add the old requests.
Press PF03 if you want not to add the old requests.

PF01=HELP PF03=END PF04=RETURN
```

Figure 29. NetView FTP Convert Saved Requests - Warning (DVGMETBE)

Press ENTER to copy the converted requests into your new request table.
Chapter 3. Writing NetView FTP V2.2.1 MVS Batch Jobs

A NetView FTP V2.2.1 MVS batch job is made up of *job control statements* and *NetView FTP V2.2.1 MVS control statements*. The job control statements are the same or similar for each batch job. Which NetView FTP V2.2.1 MVS control statements a batch job contains depends on what the job tells NetView FTP V2.2.1 MVS to do (add, query, modify, restart, or delete a request) and on what information NetView FTP V2.2.1 MVS must have to do what it is told. The following describes both types of control statements and how to code them.

You can write batch jobs to:

- Create requests
- Query requests in the queue
- Modify the server class or priority of requests in the queue
- Release and hold requests in the queue
- Restart interrupted file transfers
- Delete requests from the queue.

You tell NetView FTP V2.2.1 MVS what to do by assigning the appropriate value to the Function parameter, and by assigning values for:

- *Request parameters* which tell NetView FTP V2.2.1 MVS about the request or requests it is to add, query, modify, restart, hold, release, or delete.
- *Transfer parameters* which tell NetView FTP V2.2.1 MVS how it is to conduct the file transfer.
- *File parameters* which tell NetView FTP V2.2.1 MVS about the file it is to transfer.

These parameters are described in the *NetView FTP Parameter Reference*. For more information about administering the request queue, rebuilding the request queue, and changing the master password refer to Chapter 5, “Controlling Requests and Administering the Queue” on page 87.
Coding Job Control Statements

The following describes the job control statements you must supply when you code a batch job. Sample jobs are provided in the NetView FTP V2.2.1 MVS sample library. A similar job is shown in Figure 30.

```
//jobname JOB (accn,n),'id',
//   MSGLEVEL=(n,n),MSGCLASS=m,
//   CLASS=c,NOTIFY=uid,USER=uid
//JOBLIB DD DSN=DVG.V2R2M1.DVG1MD0,DISP=SHR
//ADREQ EXEC PGM=DVGIFBI,PARM=ENGLISH,REGION=1024K
//SYSUDUMP DD SYSOUT=*  
//DVGLOG DD SYSOUT=* 
//SYSIN DD *  
FUNCTION =ADD,XMODE = TO  
RMRLU = LUNAME  
CLASS = S  
PRIORITY = 1  
SFILEORG = PS  
SFILEID = 'DATA.SET.NAME'  
/  
/
```

Figure 30. Job Control Statements for a NetView FTP V2.2.1 MVS Batch Job

The lines that begin with a double slash (//) or a slash and an asterisk (/*) contain job control statements. The others contain sample NetView FTP V2.2.1 MVS control statements.

In your batch job, substitute appropriate values for those values shown in lowercase in Figure 30. You can change the values shown in bold type and you can specify other NetView FTP V2.2.1 MVS control statements to suit your needs. All other values must be specified as they are in the sample.

The following list explains the contents of the EXEC job control statement in the sample job:

**PGM**

DVGIFBI is the load module name of the batch job interface routine. You can run the batch job interface routine as a stand-alone utility or as a step within a multi-step job.
PARM
The value of this parameter tells NetView FTP V2.2.1 MVS which message
definition module to use. If you specify a language other than English, a
message-definition module for that language must be available, and its name
must be DVGCCMDx, where x is the first letter of the name of the specified
language. The value must be in uppercase.

<table>
<thead>
<tr>
<th>x</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>English (default)</td>
</tr>
<tr>
<td>F</td>
<td>French</td>
</tr>
<tr>
<td>G</td>
<td>German</td>
</tr>
<tr>
<td>I</td>
<td>Italian</td>
</tr>
<tr>
<td>J</td>
<td>Japanese</td>
</tr>
<tr>
<td>P</td>
<td>Portuguese</td>
</tr>
<tr>
<td>S</td>
<td>Spanish</td>
</tr>
<tr>
<td>U</td>
<td>User (any language requiring single-byte encoding)</td>
</tr>
<tr>
<td>D</td>
<td>Double (any language requiring double-byte encoding)</td>
</tr>
</tbody>
</table>

If you specify a letter other than one of those in the list, the English language
message-definition module is used.

NetView FTP V2.2.1 MVS is available from IBM with only English or Japanese
messages. However, you can translate the NetView FTP V2.2.1 MVS
messages yourself and create your own message-definition modules. For more
information about how to do this, see the NetView FTP V2 MVS Installation,
Operation, and Administration guide.

REGION
The value of this parameter specifies the amount of storage to be allocated for
the NetView FTP V2.2.1 MVS batch job interface routine.

The following list explains the contents of the sample job’s DD statements:

STEPLIB (or JOBLIB)
This DD statement allocates the libraries from which the operating system loads
the batch job interface routine. If none of these libraries contain the batch job
interface routine, the operating system searches for it from the link pack area.
If batch job interface routine was not loaded from the link pack area, the
operating system searches for it in the libraries defined in the LINKLSTxx
member of the SYS1 PARMLIB. If you use a message-definition module other
than the one for English messages, it must be available from the current
JOBLIB, STEPLIB, LPALIB, or LINKLIB.

SYSUDUMP
This DD statement allocates the file that the system uses to produce a dump if
the NetView FTP V2.2.1 MVS batch job interface routine abnormally
terminates. Although it is rarely needed, you should specify this DD statement
so that you obtain enough information to diagnose the cause of a problem,
should one arise.
**DVGLOG**

This DD statement allocates the file into which the NetView FTP V2.2.1 MVS batch job interface routine writes all its messages except for those that are displayed only on the operator's console.

Figure 30 on page 56 shows DVGLOG allocated to SYSOUT. However, DVGLOG can also be allocated to a DASD or tape-resident file. The record format of such a file must be fixed (RECFM=F) or fixed blocked (RECFM=FB), and the logical record length must be set to 120 (LRECL=120). If you specify RECFM=FB, you must also specify a block size that is a multiple of the logical record length.

**SYSIN**

This DD statement allocates the file from which the NetView FTP V2.2.1 MVS batch job interface routine reads the NetView FTP V2.2.1 MVS control statements. This file is called the *request definition file* (RDF).

Figure 30 on page 56 shows a job in which the NetView FTP V2.2.1 MVS control statements are contained in the job input stream. However, they can also be contained in a DASD resident file. The record format of such a file must be fixed (RECFM=F) or fixed blocked (RECFM=FB), and the logical record length must be set to 80 (LRECL=80).

---

**Coding NetView FTP V2.2.1 MVS Control Statements**

The NetView FTP V2.2.1 MVS control statements tell NetView FTP V2.2.1 MVS what it is to do (add, query, modify, restart, or delete a request), and provide NetView FTP V2.2.1 MVS with the information it needs to do it.

NetView FTP V2.2.1 MVS control statements have the form `keyword=value`, with just one exception. The keyword tells NetView FTP V2.2.1 MVS which parameter is being set; the value tells NetView FTP V2.2.1 MVS which value the parameter is being set to. The only exception is the End-of-Request parameter. This parameter gives you the possibility to code more than one request in one request definition file. The control statement for this parameter consists only of the keyword /EOR.

See Figure 31 on page 60 for examples of NetView FTP V2.2.1 MVS control statements.

The following rules apply when you code NetView FTP V2.2.1 MVS control statements:

**Sequence**

You can code NetView FTP V2.2.1 MVS control statements in any sequence.

**Columns**

NetView FTP V2.2.1 MVS control statements need not begin in column one.

**Lines**

You must code a NetView FTP V2.2.1 MVS control statement entirely on one line. You cannot code a NetView FTP V2.2.1 MVS control statement that continues onto a second line.

You can code more than one NetView FTP V2.2.1 MVS control statement on one line. However, you must place commas between NetView FTP V2.2.1 MVS control statements coded on a single line.
Blanks
You can place blanks before and after the equal sign (=) within a NetView FTP V2.2.1 MVS control statement.

However, NetView FTP V2.2.1 MVS interprets a blank after a NetView FTP V2.2.1 MVS control statement as the end of the line on which that control statement appears. Therefore, do not place blanks before or after a comma (,) that separates two NetView FTP V2.2.1 MVS control statements.

Comments
You can include comments:

- After the last control statement in a line, separated from that control statement by at least one blank.
- In comment lines placed between lines containing NetView FTP V2.2.1 MVS control statements. A comment line must begin with an asterisk in column one.

Values
A value sometimes consists of a list of items. These items must be separated by commas, and the value must be enclosed in parentheses.

You can enclose any value within single quotes. However, there are some values that you must enclose within single quotes. The descriptions of the control statement values tell you which values are of this type.

You must enclose within single quotes any value that contains characters other than the following:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
@#$

You must code a single quote that is part of a value as two single quotes. For example, code the value user’s file like this: ‘user’s file’. However, when you code a hexadecimal number that contains the hexadecimal digits 7D (EBCDIC code for a single quote), you do not have to code the 7D twice.

In general, NetView FTP V2.2.1 MVS translates all lowercase characters to uppercase. To prevent NetView FTP V2.2.1 MVS from doing so, include the statement TRANSLATE=NO in your request. After having encountered this statement, NetView FTP V2.2.1 MVS translates only the keywords to uppercase characters and does not change the values you entered. This setting is valid until either the end of the request or the statement TRANSLATE=YES is encountered.

With the TESTRUN option, you can tell NetView FTP whether to check the request parameters for correctness without or before adding the file-transfer request to the request queue.

If you specify TESTRUN=YES, NetView FTP validates the request without adding it to the request queue. If you code a NetView FTP control statement incorrectly, NetView FTP issues an error message.

If you specify TESTRUN=NO, NetView FTP validates the request before before adding it to the request queue. If all NetView FTP control statements are correct, the request is added to the request queue. Otherwise, NetView FTP issues an error message and the request is rejected.
If you code a NetView FTP V2.2.1 MVS control statement incorrectly, NetView FTP V2.2.1 MVS rejects the request and issues an error message. Whether NetView FTP V2.2.1 MVS detects the error before or after it begins transferring the file depends on the type of error.

NetView FTP control statements are repeated in the log file of the batch job interface routine, and MVS removes multiple blanks (blank suppression) from the control statements. If you do not want blank suppression to be performed, code a blank followed by a '|' (X'4F') in the comment area of a control statement of the batch job. When the control statements are written to the batch job interface routine MVS does not perform blank suppression following this character combination.

```
//ADD JOB (ACCN,N),'YOURID-X',
//     MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A,NOTIFY=YOURID,USER=YOURID
//JOBLIB DD DSN=DVG.NFTP221.SDVGLMD1,DISP=SHR
//BIADD EXEC PGM=DVGIFBI,REGION=2M
//SYSUDUMP DD SYSOUT=* 
//DVGLOG DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD * 
*---------------------------------------------------------------* 
*  JOB THAT ADDS A FILE-TRANSFER REQUEST MVS -> OS2  * 
*---------------------------------------------------------------* 
FUNCTION = ADD
XMODE=TO,CLASS=P
RMTLU=LUNAME
*
SSECURP='*
SFILEID='DVG.NFTP221.VERIFY.TEST' physical sequential file 
*
SECAPPC=(USERID,PASSWORD)
*
CONVERT=YES
*
RFILEID='D:\NFTPWORK\BASKET\MVS22.TXT', OS/2 file
RSTATOPT=MNE
/ 
```

Figure 31. NetView FTP V2.2.1 MVS Control Statements for a NetView FTP V2.2.1 MVS Batch Job
Creating Requests

To create a request, your RDF must specify the appropriate value for the Function parameter and for the appropriate request, transfer, and file parameters. These parameters are described in the *NetView FTP Parameter Reference*.

The following is a step-by-step description of what happens when a batch job creates a request:

1. The batch job interface routine reads the parameters from SYSIN and checks them for correctness.

2. If they are correct and you specified TESTRUN=NO in your request, the batch job interface routine uses the parameters to create a file-transfer request. NetView FTP MVS then adds the request to the request queue.

   If you specified TESTRUN=YES anywhere in your request, the batch job interface routine checks the request parameters for correctness, but the request is not added to the request queue.

3. If the value of the Wait parameter is YES, the batch job interface routine issues a message indicating the success of adding the request to the queue, and waits for the end of the file transfer before continuing. It monitors the status of the request by periodically querying the request. When the request has the status finished, the batch job interface routine:
   a. Issues an end-of-file-transfer message (with file-transfer return and reason codes)
   b. Inserts the file-transfer return code into register 15
   c. Returns control to the batch job.

   If an error is encountered while the request is being added to the queue or being periodically queried, the routine inserts a return code into register 15 and returns control to the batch job.

4. If the value of the Wait parameter is NO, the batch job interface routine does not wait for the end of the file transfer before continuing. The routine:
   a. Issues a message indicating the success of adding the request
   b. Inserts the corresponding return code into register 15
   c. Returns control to the batch job.
Transferring Members, Groups of Members, and Entire PDSs

The following examples show you how to write RDFs to create a request to select members, exclude members, and transfer entire PDSs.

Writing an RDF to Create a Request to Transfer a Group of Members

Code one MEMBER control statement for each member to be transferred. Before each member name, specify the first letter of a PDS option. The letter you specify tells NetView FTP V2.2.1 MVS under which circumstances it is to transfer the member to the PDS at the receiving system. If you transfer several members at once, they need not all have the same PDS option. You can also specify a new name for any of the members to be transferred.

Writing an RDF to Create a Request to Transfer All Members Except Those Selected

Code one MEMBER control statement for each member that is not to be transferred. Before each member name, specify an X (for exclude). Assign a value to the PDS Option parameter by coding the PDSOPT control statement. NetView FTP V2.2.1 MVS will transfer all the other members that satisfy the condition prescribed by the value of the PDS Option parameter. The PDS Option parameter is described in the NetView FTP Parameter Reference.

Writing an RDF to Create a Request to Transfer an Entire PDS

To transfer an entire PDS, do the following:

- Do not code the MEMBER, SPDSMEM, or RPDSMEM control statements.
- Specify the name of the PDS to be transferred in the Data Set Name parameter.

Querying Requests in the Queue

NetView FTP V2.2.1 MVS provides a set of query functions. To set up a query request, you must assign the appropriate values to the Function parameter and the respective request parameters in the RDF allocated in the SYSIN DD statement.

The following is a step-by-step description of what happens when a batch job queries one or more requests that are in the queue:

1. The batch job interface routine reads the parameters from SYSIN and validates them.
2. If they are valid, the batch job interface routine:
   a. Queries the appropriate request or requests
   b. Prints the queried information in the form of a message
   c. Issues a message indicating the success of querying the request
   d. Inserts the corresponding return code into register 15
   e. Returns control to the batch job.

If they are not valid, the batch job interface routine writes error messages to the batch job interface routine log file.
Modifying, Holding, and Releasing a Request in the Queue

As long as the file transfer has not started, you can modify the server class and the priority of the request. Also, you can prevent a waiting request from being processed immediately by holding it. You can then release the request whenever appropriate. To set up a modify, a hold, or a release request, you must assign the appropriate values to the Function parameter and the respective request parameters in the RDF allocated in the SYSIN DD statement.

The following is a step-by-step description of what happens when a batch job modifies, holds, or releases a request in the queue:

1. The batch job interface routine reads the parameters from SYSIN and validates them.
2. If they are valid, the batch job interface routine:
   a. Modifies the appropriate request
   b. Issues a message indicating the success of modifying the request
   c. Inserts the corresponding return code into register 15
   d. Returns control to the batch job.

If they are not valid, the batch job interface routine writes error messages to the batch job interface routine log file.

Restarting Requests in the Queue

If a file transfer failed and is requeued with the status finished and a return code greater than 8, you have the option to restart the file transfer manually. You can request a transfer restart either from checkpoint or from the beginning. You can restart the file transfer as long as the respective file transfer request is in the queue and both the sending file and the receiving file are still available and are unmodified.

To restart a file transfer, you must assign the appropriate value to the Function parameter and the respective request parameters in the RDF allocated in the SYSIN DD statement. Optionally, you can reassign values to the following parameters to override the values set in the initial request:

- Request originator
- Request password
- Server class
- Priority
- Not-before time
- Not-after time
- Status.
The following is a step-by-step description of what happens when a batch job **restarts** a request in the queue:

1. The batch job interface routine reads the parameters from SYSIN and validates them.
2. If they are valid, the batch job interface routine:
   a. Restarts the appropriate request
   b. Issues a message indicating the success of restarting the request
   c. Inserts the corresponding return code into register 15
   d. Returns control to the batch job.

If they are not valid, the batch job interface routine writes error messages to the batch job interface routine log file.

---

**Deleting Requests from the Queue**

You can delete one or all of your requests from the request queue. To delete a request, you must assign the appropriate value to the Function parameter and the respective request parameters in the RDF allocated in the SYSIN DD statement.

The following is a step-by-step description of what happens when a batch job **deletes** one or more requests from the queue:

1. The batch job interface routine reads the parameters from SYSIN and validates them.
2. If they are valid, the batch job interface routine:
   a. Deletes the appropriate request or requests
   b. Issues a message indicating the success of deleting the request
   c. Inserts the corresponding return code into register 15
   d. Returns control to the batch job.

If they are not valid, the batch job interface routine writes error messages to the batch job interface routine log file.

---

**Coding More Than One Request in One Batch Job**

You can set up more than one request in one batch job. So you can, for example, create a request, submit it, and immediately query that request, without having to call the batch job interface routine a second time. To tell the batch job interface routine that another request is to follow immediately, you use the /EOR control statement in your RDF. See the sample NetView FTP batch jobs in Chapter 7, “Sample NetView FTP V2.2.1 MVS Batch Jobs” on page 129 for an example.
Messages Issued by the Batch Job Interface Routine

The batch job interface routine issues messages to the following destinations:

**Batch Job Interface Routine Log File**
This file is defined in the startup job for the batch job interface routine, using the DVGLOG DD statement. It contains all messages issued by the batch job interface routine during its run time, except for those messages that were issued exclusively to the operator’s console. Among the messages in the batch job interface routine log file, you can find the following:

- Error messages issued when parameter values are incorrect, conflicting, or missing
- Messages that contain the return and reason codes from the adding, querying, modifying, restarting, or deleting of a request (these are queue handler command return and reason codes)
- Messages that contain the information retrieved when requests were queried
- A list of the NetView FTP V2.2.1 MVS control statements that served as input to the batch job interface routine.

**Operator’s Console**
The batch job interface routine routes a message to the operator’s console only when it cannot find a specified national-language message definition module.

Converting NetView FTP V2.1 MVS Batch Jobs for Use with NetView FTP V2.2.1 MVS

You can change most NetView FTP V2.1 MVS batch jobs to convert them into NetView FTP V2.2.1 MVS batch jobs.

When converting a NetView FTP V2.1 MVS batch job for use with NetView FTP V2.2.1 MVS note that some NetView FTP V2.1 MVS keywords were replaced by new NetView FTP V2.2.1 MVS keywords. The NetView FTP V2.1 MVS keywords are still accepted by NetView FTP V2.2.1 MVS and result in the same functions, but the batch job interface routine issues a warning message for the obsolete parameters. New batch jobs, however, should contain the NetView FTP V2.2.1 MVS keywords.

For more information on the NetView FTP parameters refer to the *NetView FTP Parameter Reference.*
Chapter 4. Writing NetView FTP V2.2.1 MVS Application Programs

This chapter contains General-Use Programming Interface and Associated Guidance Information.

The information in this chapter is of interest only to writers of application programs. For more information about how to write batch jobs, see Chapter 3, “Writing NetView FTP V2.2.1 MVS Batch Jobs” on page 55.

Note: Although the following descriptions are biased towards PL/I, the principles are applicable to Assembler, REXX, and C language.

You can write application programs to:

- Create requests
- Query requests in the queue
- Modify the server class or priority of requests in the queue
- Release and hold requests in the queue
- Restart interrupted file transfers
- Delete requests from the queue.

For more information about administering the request queue, rebuilding the request queue, and changing the master password, refer to Chapter 5, “Controlling Requests and Administering the Queue” on page 87.

Such an application program and NetView FTP V2.2.1 MVS use control blocks called the application program parameter list (APL) and the application program parameter list extension (APX) to pass information to and from each other.

An application program uses the fields of the APL to tell NetView FTP V2.2.1 MVS the following:

- Whether it is to add a request to the queue, or to query, modify, restart, or delete requests that are already in the queue
- Everything NetView FTP V2.2.1 MVS needs to know to be able to add, query, modify, restart, or delete a request
- The address and length of the query response area (QRA), which is where NetView FTP V2.2.1 MVS places the information it retrieves about queried requests
- The address and length of the message area, which is where NetView FTP V2.2.1 MVS writes any messages it issues to an application program.

3 The APL bears no relation to the programming language of the same name.
NetView FTP V2.2.1 MVS uses the fields of the APL to tell an application program the following:

- The return and reason codes from the adding, querying, modifying, restarting, or deleting of a request (these are queue handler command return and reason codes)
- For how many queried requests it retrieved information and for how many of those it wrote the information to the QRA
- How many messages the application program interface routine issued and how many of those it wrote to the message area.

An application program uses the fields of the APX to tell NetView FTP V2.2.1 MVS which members of a PDS are to be sent, and with which PDS option, or which members are not to be sent.

An application program passes control to NetView FTP V2.2.1 MVS by invoking the NetView FTP V2.2.1 MVS application program interface routine, which has the name DVGAPI (the alias name of DVGIFAI) and providing the APL as a parameter.

Space for the APL, the APX, the QRA, and the message area must be allocated by the application program. All of these areas must be defined on a doubleword boundary and must reside in storage to which DVGAPI has the authorization to write. This storage can be above the 16MB\(^4\) boundary. The storage in which a NetView FTP V2.2.1 MVS application program resides can also be above the 16MB boundary.

The sample libraries ADVGSAM0, ADVGSAC0, and ADVGSAR0 on the distribution tape contain sample application programs. You can use these application programs as provided, or you can modify them to suit the requirements of your system.

## Initializing and Assigning Values to the Fields of the APL and APX

Before an application program can assign values to the fields of the APL or APX, it must first initialize the APL or APX, or both, by doing one of the following:

- Set the value of the APLVBC field to APLCLEAR, the value of the APLID field to APLINIT, and the value of the APLLNGTH field to LENGTH(APL). Call DVGAPI which performs the correct initialization of all APL fields. APLVBC, APLINIT, and APLLNGTH remain unchanged. APLLNGTH must be specified to provide DVGAPI with the length of the APL passed by the application program.

  **Note:** All APL addresses are reset to zero. So do not store any information in the APL (except APLVBC, APLID, and APLLNGTH) before you call DVGAPI with APLVBC=APLCLEAR.

- Set the values of all APL fields to the initialization values described in the *NetView FTP Parameter Reference*.

---

\(^4\) MB equals 1,048,576 bytes.
For APLID and APXID, assign the control block identifier to the appropriate field. If the application program uses a mapping macro, it can do this by moving the appropriate symbolic constant to that field.

<table>
<thead>
<tr>
<th>Control Block</th>
<th>Control Block Identifier Field</th>
<th>Initialization Value</th>
<th>Mapping Macro</th>
<th>Symbolic Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>APL</td>
<td>APLID</td>
<td>&quot;DVGAPL&quot;</td>
<td>DVGAPL</td>
<td>APLACCR</td>
</tr>
<tr>
<td>APX</td>
<td>APXID</td>
<td>&quot;DVGAPX&quot;</td>
<td>DVGAPX</td>
<td>APXACCR</td>
</tr>
</tbody>
</table>

**Note:** Addresses of areas, such as the message area, the long-file area, or the feedback area, and their associated length, must be set to zero if not used.

The following rules apply when an application program assigns values to the fields of the APL or APX:

- Character input must be left-adjusted with padding blanks.
- Numeric input must be right-adjusted with leading zeros.
- If you code a value or a field name incorrectly, NetView FTP V2.2.1 MVS rejects the request. Whether NetView FTP V2.2.1 MVS detects the error before or after it begins transferring the file depends on the type of error.

### Creating Requests

To create a request, your application program must specify the value ADD to the Function parameter and values for the appropriate request, transfer, and file parameters. These parameters are described in the *NetView FTP Parameter Reference*.

The following is a step-by-step description of what happens when an application program creates a request.

1. The application program sets the value of the APLVBC field to a value that indicates that NetView FTP V2.2.1 MVS is to add a request to the queue.
2. It sets the values of the appropriate transfer, file, and request parameters to their appropriate values.
3. It loads and calls the NetView FTP V2.2.1 MVS application program interface routine, DVGAPI, the alias name of the module DVGIFAI, then:
   a. DVGAPI checks the contents of the APL and APX for correctness. The pre-queuing user-exit routine is called.
   b. If the contents of the APL and APX do not have inconsistencies, DVGAPI adds the request to the request queue.
   c. DVGAPI places the request number into the field APLRQNUM.
   d. DVGAPI places a return code and a reason code that indicates the success of the adding of the request into the fields APLRC and APLRSN, respectively.
   e. DVGAPI places the same return code into register 15.
   f. DVGAPI returns control to the application program.

**Note:** NetView FTP V2.2.1 MVS does not wait for a server to begin processing the request before it returns control to the application program.
After NetView FTP V2.2.1 MVS has added the request to the request queue, it does not report to the application program any more information about the request. However, the application program can be written so that it periodically queries the request and reacts accordingly to the information that is retrieved. For example, an application program can be written so that it delays further processing until it learns by querying a request that the request has the status finished.

### Transferring Members, Groups of Members, and Entire PDSs

This section lists some examples of how to write application programs to create a request for selecting members, excluding members, and transferring entire PDSs.

#### Writing an Application Program to Create a Request to Transfer a Group of Members

1. Assign appropriate values to the fields of the APL.
2. Assign the value APLGMBRN\(^5\) to the field APLVBC.
3. Call DVGAPI. DVGAPI returns a value in the field APLSMBRN.\(^5\) This value is the maximum number of members that can be selected at once.
4. Use the following formula to calculate the minimum number of bytes of main storage you must get for the APX:
   
   \[16 + (17 \times N)\]
   
   where \(N\) is the number of members being selected. This number cannot be larger than the value returned to APLSMBRN.\(^5\)
5. Get the amount of main storage calculated in the formula shown in step 4.
6. Place the pointer to the main storage (returned to register 1) into the fields APXPTR and APLAPXPT.\(^6\)
7. Assign the value APXACCR (sym) or *DVGAPX* (char) to the field APXID.
8. Assign the value APXTYPS to the field APXTYP.
9. Place the amount of main storage got (in bytes) into the field APXLEN.
10. Assign the value 0 (num) to the field APXENT.
11. Set the value of the field APXSPPTR to the value of the field APXPTR plus the length of the APX header (16 bytes).\(^6\)
12. Place the name of a member being selected into the field APXSNME.
13. Place the new name (if any) the member is to have in the receiving PDS into the field APXRNME.
14. Place the PDS option to be used to determine under which circumstances the member is to be written to the receiving PDS into the field APXOPT.
15. Increment the value of APXENT by 1.
16. Increment the value of APXSPPTR by the length of APXSEL.\(^6\)
17. Repeat steps 12 through 16 until all members have been placed in the list.

---

\(^5\) APLXMBR#, APLGMBR#, and APLSMBR# can still be used for existing PL/I and Assembler programs.

\(^6\) Omit this step if you write an application program in REXX.
18. Assign the value APLADDRE to the field APLVBC.
19. Call DVGAPI. DVGAPI creates the file-transfer request and adds it to the request queue.

**Writing an Application Program to Create a Request to Transfer All Members Except Those Selected**

1. Assign appropriate values to the fields of the APL.
2. Assign the value APLGMBRN to the field APLVBC.
3. Call DVGAPI. DVGAPI returns a value in the field APLXMBRN. This value is the maximum number of members that can be excluded at once.
4. Use the following formula to calculate the minimum amount of main storage you must get for the APX:
   
   \[ 16 + (8 \times N) \]
   
   where \( N \) is the number of members being excluded. This number cannot be larger than the value returned to APLXMBRN.
5. Get the amount of main storage calculated in the formula shown in step 4.
6. Place the pointer to the main storage (returned to register 1) into the fields APXPTR and APLAPXPT.
7. Assign the value APXACCR (sym) or *DVGAPX* (char) to the field APXID.
8. Assign the value APXTYPX to the field APXTYP.
9. Place the amount of main storage got (in bytes) into the field APXLEN.
10. Assign the value 0 (num) to the field APXENT.
11. Set the value of the field APXXPTR to the value of the field APXPTR plus the length of the APX header (16 bytes).
12. Place the PDS option to be used to determine under which circumstances the members that are not excluded are to be written to the receiving PDS into the field APLXOPT.
13. Place the name of a member being excluded into the field APXXNME.
15. Increment the value of APXXPTR by the length of APXEXCL.
16. Repeat steps 13 through 15 until all members have been placed in the list.
17. Assign the value APLADDRE to the field APLVBC.
18. Call DVGAPI. DVGAPI creates the file-transfer request and adds it to the request queue.

**Writing an Application Program to Create a Request to Transfer an Entire PDS**

To transfer an entire PDS, do the following:

- Do not place values into the APXPTR, APLSDYMM, or APLRDYMM fields.
- Specify the name of the PDS to be transferred in the Data Set Name parameter.
Querying Requests in the Queue

NetView FTP V2.2.1 MVS provides a set of query functions. To set up a query request, your application program must assign the appropriate values to the Function parameter and the respective request parameters.

The following is a step-by-step description of what happens when you query one or more requests in the queue using an application program:

1. The application program sets the value of the APLVBC field to a value that indicates that NetView FTP V2.2.1 MVS is to query one or more requests that are in the queue. For more information, see the description of the Function parameter in the NetView FTP Parameter Reference.

2. The application program places the address and length of the QRA in the fields APLFBAP and APLFBALN, respectively.

3. The QRA size is the user query record (QSA) size or the administrator query record size multiplied by the expected number of records. For more information, examine the macros DVGQSR and DVGQAR in the DVGMAC0 library. You can also find a listing, with offsets and field lengths, in member DVGCXMOL in the DVGSAM0 library.

4. It loads and calls the NetView FTP V2.2.1 MVS application program interface routine, DVGAPI, the alias name of module DVGIFAI.
   a. DVGAPI checks the contents of the APL for correctness.
   b. If the contents of the APL are correct, DVGAPI creates one record of information, called a query record, for each of the queried requests. A query record can be of one of the following types:
      - A user query record, which DVGAPI creates when you ask for information about requests that is intended for users.
      - An administrator query record, which DVGAPI creates when you ask for information about requests that is intended for the NetView FTP V2.2.1 MVS administrator.
   c. DVGAPI moves each of these records, one after the other, to the QRA.
   d. DVGAPI places a return code and a reason code that indicates the success of the querying of the requests into the fields APLRC and APLRSN, respectively.
   e. DVGAPI places the same return code into register 15.
   f. DVGAPI places the number of query records retrieved into the field APLQTCNT.
   g. DVGAPI places the number of query records moved to the QRA into the field APLQACNT.
   h. If the value of APLQTCNT is greater than that of APLQACNT, the application program should repeat querying the requests until the values for APLQTCNT and APLQACNT are equal.
   i. DVGAPI returns control to the application program.
Modifying, Holding, and Releasing a Request in the Queue

As long as the file transfer has not started, you can modify the server class and the priority of the request. Also, you can prevent a waiting request from being processed immediately by holding it. You can then release the request whenever appropriate. To set up a modify, a hold, or a release request, your application program must assign the appropriate values to the Function parameter and the respective request parameters.

The following is a step-by-step description of what happens when you modify, hold, or release a request in the queue using an application program:

1. The application program sets the value of the APLVBC field to a value that indicates that NetView FTP V2.2.1 MVS is to modify, hold, or release a request in the queue.
2. It sets the values of the appropriate request parameters to appropriate values.
3. It loads and calls the NetView FTP V2.2.1 MVS application program interface routine, DVGAPI, the alias name of module DVGIFAI.
   a. DVGAPI checks the contents of the APL for correctness.
   b. If the contents of the APL are correct, DVGAPI modifies the request specified in the APL.
   c. DVGAPI places a return code and a reason code into the fields APLRC and APLRSN, respectively.
   d. DVGAPI places the same return code into register 15.
   e. DVGAPI returns control to the application program.

Restarting Requests in the Queue

If a file transfer failed and is requeued with the status finished and a return code greater than 8, you have the option to restart the file transfer manually. In a restart request you can request a transfer restart either from checkpoint or from the beginning. You can restart the file transfer as long as the respective file transfer request is in the queue and both the sending file and the receiving file are still available and are unmodified.

To restart a file transfer, your application program must assign the appropriate value to the Function parameter and the respective request parameters. Optionally, your application program can reassign values to the following parameters to override the values set in the initial request:

- Request originator
- Request password
- Server class
- Priority
- Not-before time
- Not-after time
- Status.
The following is a step-by-step description of what happens when an application program **restarts** a request in the queue:

1. The application program sets the value of the APLVBC field to a value that indicates that NetView FTP V2.2.1 MVS is to restart a request in the queue.
2. It sets the values of the appropriate request parameters to appropriate values.
3. It loads and calls the NetView FTP V2.2.1 MVS application program interface routine, DVGAPI, the alias name of module DVGIFAI.
   a. DVGAPI checks the contents of the APL for correctness.
   b. If the contents of the APL are correct, DVGAPI restarts the request specified in the APL.
   c. DVGAPI places a return code and a reason code into the fields APLRC and APLRSN, respectively.
   d. DVGAPI places the same return code into register 15.
   e. DVGAPI returns control to the application program.

### Deleting Requests from the Queue

You can delete one or all of your requests from the request queue. To delete a request, your application program must assign the appropriate value to the Function parameter and the respective request parameters in the request definition file allocated in the SYSIN DD statement.

The following is a step-by-step description of what happens when an application program **deletes** one or more requests from the queue:

1. The application program sets the value of the APLVBC field to a value that indicates that NetView FTP V2.2.1 MVS is to delete one or more requests from the queue.
2. It sets the values of the appropriate request parameters to appropriate values.
3. It loads and calls the NetView FTP V2.2.1 MVS application program interface routine, DVGAPI, the alias name of module DVGIFAI.
   a. DVGAPI checks the contents of the APL for correctness.
   b. If the contents of the APL are correct, DVGAPI deletes the requests specified in the APL.
   c. DVGAPI places a return code and a reason code into the fields APLRC and APLRSN, respectively.
   d. DVGAPI places the same return code into register 15.
   e. DVGAPI returns control to the application program.

### Using the Message Area

Before the application program interface routine adds, queries, modifies, or deletes a request, it checks the contents of the APL and APX. If it detects values that are missing, incomplete, incorrect, or incompatible with other values, it issues one or more messages. An application program can define an area to which the interface routine can write such messages. This area is called the **message area**.
An application program defines the message area by specifying values for the APLMAPTR and APLMSGLN parameters. These parameters are described in “Address and Length of the Message Area” on page 84. When setting the length of the message area, take into account that a typical message from the application program interface routine is 80 bytes on average. However, some messages can exceed 132 bytes. Therefore you should check the length of each message before moving it to your print area. For more information about the number of messages moved to the message area, see “Number of Messages Issued and Moved to the Message Area” on page 84.

The messages placed into the message area are the same as for the batch job interface. This results in showing the equivalent batch job interface keywords for an APL field. Each message the application program interface routine issues is preceded by a 4-byte field. This field contains the length of the message as a binary integer. Figure 32 shows the layout of a string of such messages.

### Figure 32. Layout of a String of Messages

<table>
<thead>
<tr>
<th>Byte position:</th>
<th>0 4 8............ n (n+4)............</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of</td>
<td>Length of</td>
</tr>
<tr>
<td>messages</td>
<td>first</td>
</tr>
<tr>
<td></td>
<td>message</td>
</tr>
</tbody>
</table>

**Note:** If an application program specifies an incorrect parameter value, an error message may contain a string of Xs instead of the erroneous value.

## NetView FTP V2.2.1 MVS Macros Used by Application Programs

Application programs written in PL/I, Assembler, or C language can use macros provided by NetView FTP V2.2.1 MVS to load and call DVGAPI, and to carry out mapping for the APL, APX, and QRA.

These macros are:

- **DVGCALL** Loads and calls DVGAPI, the alias name for the module DVGIFAI (PL/I, Assembler)
- **DVGAPL** Maps the contents of the entire APL (PL/I, Assembler)
- **DVGAPX** Maps the contents of one record of the APX (PL/I, Assembler)
- **DVGQSR** Maps the contents of one user query record in the QRA (PL/I, Assembler)
- **DVGQAR** Maps the contents of one administrator query record in the QRA (PL/I, Assembler)
- **DVGCXCAP** Maps the contents of the entire APL (C language)
- **DVGCXCAX** Maps the contents of one record of the APX (C language)
- **DVGCXCQS** Maps the contents of one user query record in the QRA (C language)
- **DVGCXCQR** Maps the contents of one administrator query record in the QRA (C language).

**Note:** Application programs written in REXX use external functions to access the APL, APX, and QRA. For more information about REXX external functions see the NetView FTP Customization guide.
The macros identified in this section are provided to allow a customer installation to write programs that use the services of NetView FTP V2.2.1 MVS. Only those macros identified in this section should be used to request or receive the services of NetView FTP V2.2.1 MVS. These macros are general-use programming interfaces.

Application programs written in other high-level IBM languages must load and call DVGAPI directly, and must carry out their own data mapping. These macros are located in the NetView FTP MVS macro library. Ask your system programmer how you can obtain copies of them.

How to Call NetView FTP V2.2.1 MVS

The following explains how to call NetView FTP V2.2.1 MVS from an application program written in PL/I, Assembler, REXX, or the C language.

Calling NetView FTP V2.2.1 MVS from a PL/I Language Program

If you write an application program in PL/I, you can use the DVGCALL executable macro directly to load and call DVGAPI, the alias name for module DVGIFAI. Figure 33 shows how to code the DVGCALL macroinstruction in a PL/I application program.

```
%INCLUDE SYSLIB(DVGCALL);
```

Figure 33. DVGCALL Macroinstruction (PL/I)

There are no macro parameters for the DVGCALL macro. Because DVGCALL uses the address of the APL, an application program that issues the DVGCALL macroinstruction must first establish the addressability of the APL.

Calling NetView FTP V2.2.1 MVS from an Assembler Language Program

If you write an application program in Assembler, you can use the DVGCALL executable macro directly to load and call DVGAPI, the alias name for module DVGIFAI.

Before DVGCALL is invoked, register 13 must point to a standard register save area that is addressable in the primary mode.

After DVGAPI is invoked:

- The contents of registers 0, 1, and 14 are altered unpredictably.
- The contents of registers 2 through 13 are unchanged.
- Register 15 contains a return code from NetView FTP V2.2.1 MVS that indicates the success of the adding, querying, or deleting that took place.

Figure 34 shows how to code the DVGCALL macroinstruction in an Assembler application program.

```
COPY DVGCALL
... name DVGCALL PARM=addr or name DVGCALL REG=number
```

Figure 34. DVGCALL Macroinstruction (Assembler)
Where:

**name**  Is the name of the macroinstruction.

**addr**  Is the A-type address of a field that contains the address of the APL.

**number**  Is the number of the register that contains the address of a field that contains the address of the APL. If you specify a register number, specify a number from 2 to 12.

Figure 35 and Figure 36 show examples of the coding of a DVGCALL macroinstruction whose name is CALL01.

### Figure 35. Coding a DVGCALL Macroinstruction Using an A-Type Address

```
COPY DVGCALL
  CALL01 DVGCALL PARM=APLADDR
  APLADDR DC A(DVGAPL)
```

### Figure 36. Coding a DVGCALL Macroinstruction Using a Register

```
COPY DVGCALL
  LA 2,APLADDR
  CALL01 DVGCALL REG=2
  APLADDR DC A(DVGAPL)
```

### Calling NetView FTP V2.2.1 MVS from a C-Language Program

You can call NetView FTP V2.2.1 MVS from an application program written in the C language. This is done using an assembly-interface routine. The C-language file DVGCXCAD must be included to declare the assembly-interface routine DVGCXCAI. DVGCXCAD contains the statements shown in Figure 37.

### Figure 37. Coding a C-Language EXEC

```
#pragma linkage(DVGCXCAI,OS)
extern long int DVGCXCAI();
long int DGRETC;
```

DVGCXCAI is an Assembler function that is coded in such a way that a C-language application program can invoke it. DVGCXCAI must be linked with the C-language application program.

For detailed information about application programs written in the C language, see the *NetView FTP Customization* guide.
Calling NetView FTP V2.2.1 MVS from a REXX Application Program

You can use the REXX external function DVGCXRIA to call the NetView FTP V2.2.1 MVS module DVGAPI. The REXX external function DVGCXRIA prepares and offers the APL and APX control blocks, message area, and query response area to NetView FTP V2.2.1 MVS. It also adds to the APL the addresses of the message area, query response area, and APX, and transfers control to NetView FTP V2.2.1 MVS via the DVGAPI.

Note: This is similar to a PL/I application program calling the NetView FTP V2.2.1 MVS module DVGAPI.

Calling the REXX External Function DVGCXRIA: When NetView FTP V2.2.1 MVS returns control to DVGCXRIA, the control blocks are made available to the REXX application program where they can be inspected by the external function DVGCXRGT.

To call DVGCXRIA, code the statement as shown in the following:

```
call dvgcxria (parm-1,parm-2,parm-3,parm-4,parm-5,parm-6,parm-7)
```

Figure 38. Calling the REXX External Function DVGCXRIA

The values the parameters can have are explained in the following:

- **PARM-1**: Defines the APL control block name (DVGAPL). You must specify a value for this parameter.
- **PARM-2**: Defines the message area (DVGMSG). This parameter is optional.
- **PARM-3**: Defines the query response area (DVGQRA). This parameter is optional.
- **PARM-4**: Defines the APX control block (DVGAPX). This parameter is optional.
- **PARM-5**: Defines DVGSPDAT. This parameter is optional.
- **PARM-6**: Defines DVGRPDAT. This parameter is optional.
- **PARM-7**: Defines DVGUXDAT. This parameter is optional.

Specify these parameters in the sequence shown. If you do not specify the correct sequence, an error occurs and the routine terminates.

The control blocks are assigned to the subroutine in the following sequence:

DVGAPL, DVGMSG, DVGQRA, DVGAPX, DVGSPDAT, DVGRPDAT, DVGUXDAT

You must specify PARM-1, the other parameters are optional. If you omit an optional parameter, include a comma (,) at the position of the omitted parameter, unless it is at the end of the sequence.

Example: The following example shows how to call the external function DVGCXRIA:

```
x = dvgcxria(dvgapl,dvgmsg,,dvgspdat,dvgrpdat)
```
**Return Values:** The external function returns the return code from the REXX Variable Access Interface (IRXEXCOM). These codes are described in *TSO/E Version 2—REXX/MVS Reference*.

When NetView FTP V2.2.1 MVS returns control to the external function, the following control blocks are available to the REXX application program:

- DVGAPL
- DVGMSG
- DVGQRA
- DVGAPX
- DVGSPDAT
- DVGRPDAT
- DVGUXDAT.

**Error Handling:** The external function checks the transferred parameters. If no parameters are transferred, return code 4 is set in register 15 and returned to the calling location.

**What DVGCXRIA Does:** DVGCXRIA checks the call statement to determine which parameters are specified. What DVGCXRIA does next depends on the following:

- If PARM-2 is specified, the address of the query response area is loaded into the APL field APLMAPTR.
- If PARM-3 is specified, the address of the query response area is loaded into the APL field APLFBAP.
- If PARM-4 is specified, the address of the APX area is loaded into the APL field APLAPXPT.
- If PARM-5 is specified, the address of the data area for post-transfer (sending) program is loaded into the APL field APLSPPTR.
- If PARM-6 is specified, the address of the data area for post-transfer (receiving) program is loaded into the APL field APLRPPTR.
- If PARM-7 is specified, the address of the data area for the user exit is loaded into the APL field APLUXPTR.

Next, DVGCXRIA calls the NetView FTP V2.2.1 MVS module DVGIFAI and transfers the address of the APL control block to NetView FTP V2.2.1 MVS. When the subroutine is returned from NetView FTP V2.2.1 MVS, the following REXX variables are set with the REXX interface VARIABLE ACCESS (IRXECOM):

- DVGAPL, DVGMSG, DVGAPX, DVGSPDAT, DVGRPDAT, DVGUXDAT

Only the control blocks specified in the function call are made available to the REXX application program.

A control-block field can then be inspected by using the external function DVGCXRGRT.
NetView FTP V2.2.1 MVS Mapping Macros

DVGAPL, DVGAPX, DVGQSR, and DVGQAR are nonexecutable mapping macros that provide names for the fields of the APL, the APX, and the query records. You can use:

**DVGAPL** To map the entire APL
**DVGAPX** To map one record of the APX
**DVGQSR** To map one user query record in the QRA
**DVGQAR** To map one administrator query record in the QRA.

DVGAPL also contains a set of symbolic constants (in PL/I called constants; in Assembler called equates) that can be assigned to the fields of the APL instead of literal values.

The layouts of the APL, APX, and the query records can be found in the data sets DVGCXMOL and DVGMXMOL in the sample library. Ask your system programmer how you can obtain copies of them.

DVGQSR and DVGQAR also contain header lines, which are symbolic constants. An application program can write a header line to an output file and then write contents of the QRA below it to form a table. Such a table might look like the one shown in Figure 39.

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>NAME</th>
<th>CL</th>
<th>PR</th>
<th>ST</th>
<th>RET</th>
<th>RSN</th>
<th>RECORDS</th>
<th>ORIGINATOR</th>
<th>DATE</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>238</td>
<td>FTPSENDW</td>
<td>2</td>
<td>3</td>
<td>W</td>
<td>M</td>
<td>0</td>
<td></td>
<td>MARY</td>
<td>94/10/22</td>
<td>17:01</td>
</tr>
<tr>
<td>367</td>
<td>FTPPROF1</td>
<td>1</td>
<td>3</td>
<td>H</td>
<td>J</td>
<td>0</td>
<td></td>
<td>JOBNAMEA</td>
<td>94/10/22</td>
<td>17:13</td>
</tr>
<tr>
<td>412</td>
<td>FTPPROF2</td>
<td>1</td>
<td>3</td>
<td>A</td>
<td>J</td>
<td>0</td>
<td></td>
<td>JOBNAMEA</td>
<td>94/10/22</td>
<td>17:13</td>
</tr>
<tr>
<td>521</td>
<td>MYPROF0X</td>
<td>5</td>
<td>7</td>
<td>F</td>
<td>12</td>
<td>302</td>
<td>30348</td>
<td>JOBNAMEX</td>
<td>94/10/22</td>
<td>12:33</td>
</tr>
<tr>
<td>1344</td>
<td>FTPRECVS</td>
<td>4</td>
<td>9</td>
<td>F</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>YOURID</td>
<td>94/10/22</td>
<td>13:44</td>
</tr>
<tr>
<td>1678</td>
<td>FTPPROF2</td>
<td>2</td>
<td>3</td>
<td>W</td>
<td>J</td>
<td>0</td>
<td></td>
<td>JOB222</td>
<td>94/10/22</td>
<td>15:55</td>
</tr>
<tr>
<td>1867</td>
<td>MYPROF14</td>
<td>0</td>
<td>3</td>
<td>W</td>
<td></td>
<td></td>
<td>0</td>
<td>CARLB</td>
<td>94/10/22</td>
<td>16:01</td>
</tr>
</tbody>
</table>

Figure 39. Sample Table of Queried Information

**DVGAPL, DVGAPX, DVGQSR, and DVGQAR Macroinstructions (PL/I)**

Figure 40 shows how to code the DVGAPL macroinstruction in a PL/I application program.

```pli
%DCL APLSRCE CHARACTER; /* Declare SRCE macro variable */
%DCL APLSTOR CHARACTER; /* Declare STOR macro variable */
%APLSRCE = 'PLI'; /* Assign a value to SRCE macro variable */
%APLSTOR = ' ' ; /* Assign a value to STOR macro variable */
//%APLSTOR = 'BASED'; /* Assign a value to STOR macro variable */
%INCLUDE SYSLIB(DVGAPL);
```

Figure 40. Coding a DVGAPL Macroinstruction (PL/I)
Macroinstructions DVGAPX, DVGQSR, and DVGQAR are coded in the same way. Macroinstruction DVGAPL must declare or specify a value for the APLSRCE macro variable. This value must be ‘PLI’.

The value of the APLSTOR macro variable can be one of the following:

‘ ’ The macro generates a structure that is not based and that has the same name as the macroinstruction (DVGAPL, DVGAPX, DVGQSR, or DVGQAR).

‘BASED’ The macro generates a structure that is based and that has the name macname BASED(pointer), where:

macname Is the name of the macro being used (DVGAPL, DVGAPX, DVGQSR, or DVGQAR).

pointer Is the name of a pointer.

The name of the pointer and the value to which it must be set depends on which macro is being used.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Name of Pointer</th>
<th>Value to Which Pointer Must Be Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGAPL</td>
<td>APLPPTR</td>
<td>Address of the APL</td>
</tr>
<tr>
<td>DVGAPX</td>
<td>APXPTR</td>
<td>Address of the APX</td>
</tr>
<tr>
<td>DVGQSR</td>
<td>QSRPTR</td>
<td>Address of the QRA</td>
</tr>
<tr>
<td>DVGQAR</td>
<td>QARPTR</td>
<td>Address of the QRA</td>
</tr>
</tbody>
</table>

The declaration for the pointer is automatically generated in the macro.

Figure 41 shows examples of the coding of DVGAPL, DVGAPX, DVGQSR, and DVGQAR macroinstructions.

```
%DCL APLSRCE CHARACTER;
%DCL APLSTOR CHARACTER;
%APLSRCE = 'PLI';
%APLSTOR = 'BASED';
%INCLUDE SYSLIB(DVGAPL);

%DCL APXSTOR CHARACTER;
%APXSTOR = ' ';  
%INCLUDE SYSLIB(DVGAPX);

%DCL QSRSTOR CHARACTER;
%QSRSTOR = ' ';  
%INCLUDE SYSLIB(DVGQSR);

%DCL QARSTOR CHARACTER;
%QARSTOR = 'BASED';
%INCLUDE SYSLIB(DVGQAR);
```

Figure 41. Examples of Macroinstructions (PL/I)
DVGAPL, DVGAPX, DVGQSR, and DVGQAR Macroinstructions (Assembler)

Figure 42 shows how to code the DVGAPL, DVGAPX, DVGQSR, and DVGQAR macroinstructions in an Assembler application program.

```
label macname DSECT=val
```

*Figure 42. Coding DVGAPL, DVGAPX, DVGQSR, and DVGQAR Macroinstructions (Assembler)*

Where:

- **label** is the label of the statement.
- **macname** is the name of the macro (DVGAPL, DVGAPX, DVGQSR, or DVGQAR).
- **val** is one of the following:
  - **NO**: This value causes the control block to be made a part of the control section where the corresponding macro is coded. The control block is given a label that is the same as the name of the corresponding macro (that is, DVGAPL, DVGAPX, DVGQSR, or DVGQAR). This label can be used to address the control block. NO is the default.
  - **YES**: This value causes the control block to be made a dummy control section with the same name as the corresponding macro. A valid CSECT must immediately precede this DSECT.

Figure 43 shows examples of the coding of DVGAPL, DVGAPX, DVGQSR, and DVGQAR macroinstructions.

```
COPY DVGAPL
COPY DVGAPX
COPY DVGQSR
COPY DVGQAR

: CALL02 DVGAPL DSECT=YES
CALL03 DVGAPX DSECT=NO
CALL04 DVGQSR DSECT=NO
CALL05 DVGQAR DSECT=YES
```

*Figure 43. Examples of Macroinstructions (Assembler)*

**Note:** When processing the information returned by a query all command (APLVBC=APLQRYAR), only one Query Status Record (QSR) is made available per invocation of macro DVGCALL if macro DVGQSR was invoked to create a CSECT for the QSR.

For better performance, invoke macro DVGQSR so that a DSECT is created, which can be used to map each QSR in the QRA, provided that you have defined the QRA big enough to allow for several QSRs.
If you invoke macro DVGQAR so that a CSECT is created for the Query Administrator Record (QAR), then only one QAR is made available per invocation of macro DVGCALL. This is the case for all QAR subtypes. So if you expect more than one class information in the general information or several requests, invoke macro DVGQAR so that a DSECT is created, which can be used to map each QAR in the Query Response Area (QRA), and define the QRA big enough to allow for several QARs.

---

### Specifying the APL and APX Communication Fields

The APL and APX contain fields that application programs and NetView FTP V2.2.1 MVS use to communicate with each other. These fields are called *communication fields* and are described in the following.

#### APL Control Block Identifier

The field APLID contains the APL identifier, and must always contain the character string “DVGAPL”. This field enables NetView FTP V2.2.1 MVS to identify the APL, and makes it easier to locate the APL in a dump.

#### Return and Reason Codes

After NetView FTP V2.2.1 MVS has added, queried, modified, restarted, or deleted requests, it places a return code in the field APLRC. It also places the same return code into register 15. This return code, which is a *queue handler command return code*, explains the outcome of the adding, querying, modifying, restarting, or deleting; that is, it indicates if anything went wrong and, if so, what. NetView FTP V2.2.1 MVS also places a reason code into field APLRSN. The reason code gives additional information about an error.

When an application program passes control over the APL to NetView FTP V2.2.1 MVS, it need not set a return code. However, when NetView FTP V2.2.1 MVS passes control back to an application program, it should check the return code to determine whether an error has occurred. For a description of the return and reason codes set by NetView FTP V2.2.1 MVS, see *NetView FTP Messages and Codes*.

#### Current Level of NetView FTP and Length of the APL

An application program should put the current level of NetView FTP into the field APLVIDC using the constant APLVI22, and the length of the APL into the field APLLNGTH using the constant APLLEN. You can enter all values defined in earlier releases of NetView FTP in the field APLLNGTH. Therefore you need not compile user programs that define APLs with every new release. Note that new fields are added sequentially to the APL and that NetView FTP is using only fields located within the defined value.
Address and Length of the QRA

An application program uses the fields APLFBAP and APLFBALN to tell NetView FTP V2.2.1 MVS the address and length, respectively, of the QRA. An application program must define the QRA on a doubleword boundary and specify values for these fields if it queries one or more requests.

Note: REXX and C-language application programs do not use addresses or lengths. To enable REXX and C language to interpret the address and length of the QRA, variables are used.

Number of Query Records Retrieved and Moved to the QRA

NetView FTP V2.2.1 MVS uses the field APLQTCNT to tell an application program how many query records it retrieved. It uses the field APLQACNT to tell an application program how many of the query records it moved to the QRA.

If the value of APLQTCNT is greater than that of APLQACNT, then the QRA is too small. This is indicated by APLRC=24 and APLRSN=196. If this occurs, NetView FTP V2.2.1 MVS places the request number of the first query record, that does not fit into the QRA, into the field APLRQNUM. If the application program then queries all requests again, NetView FTP V2.2.1 MVS retrieves query records only for the requests with numbers greater than or equal to this number. The application program can repeat querying all requests until the values in APLQTCNT and APLQACNT are equal. This way, all of the retrieved query records can eventually be moved to the QRA.

Address and Length of the Message Area

An application program uses the fields APLMAPTR and APLMSGLN to tell NetView FTP V2.2.1 MVS the address and length, respectively, of its message area. When setting the length of the message area, take into account that a typical message from the application program interface routine is about 80 bytes long. The number of messages that the interface routine issues depends on the number of errors it detects while checking parameter values for completeness and validity.

Note: REXX and C-language application programs do not use addresses or lengths. To enable REXX and C language to interpret the address and length of the message area, variables are used.

Number of Messages Issued and Moved to the Message Area

The application program interface routine uses the field APLMTOT to tell an application program how many messages it issued. It uses the field APLMACT to tell an application program how many of those messages it moved to the message area.

If the value of APLMTOT is greater than that of APLMACT, then the message area is too small to hold all of the messages that were issued. When this is so, the messages that did not fit into the message area are lost.
Maximum Number of Members that Can Be Selected or Excluded

An application program can ask NetView FTP V2.2.1 MVS to tell it how many members it can place in a list of members to be selected or excluded. The maximum number of members that can be selected is returned in the field APLSMBRN; the maximum number of members that can be excluded is returned in the field APLXMBRN. How to use these fields is described in the *NetView FTP Parameter Reference*.

Specifying the APX Control Block Identifier

The field APXID contains the APX identifier, and must always contain the character string “DVGAPX”. This field enables NetView FTP V2.2.1 MVS to identify the APX, and makes it easier to locate the APX in a dump.

---

7 APLXMBR# and APLSMBR# can still be used for existing PL/I and Assembler programs.
Chapter 5. Controlling Requests and Administering the Queue

You can write batch jobs or application programs that:

- Control requests, that is:
  - Retrieve information about (query) requests that are in the queue
  - Modify, release, and hold requests that are in the queue
  - Restart requests that are in the queue
  - Delete requests from the queue.

- Administer the request queue, that is:
  - Retrieve information about (query) the queue
  - Query all the requests in the queue that have a certain status.

Such batch jobs and application programs tell NetView FTP V2.2.1 MVS what to do by assigning the appropriate value to the Function parameter. They give NetView FTP V2.2.1 MVS the information it needs, to carry out this function by assigning values to other parameters. Which parameters you must specify values for and the values they can have depends on the function.

Controlling Requests

To query, modify, restart, or delete requests, your batch job or application program must specify the appropriate value to the Function parameter and for the appropriate request parameters. Which parameters you need to specify values for and what values these parameters should have depends on what you are doing. The parameters and their values are shown in Figure 44, Figure 45, Figure 46, Figure 47, and Figure 48, and are described in the NetView FTP Parameter Reference.

NetView FTP V2.2.1 MVS retrieves the following information for each request:

- Request number
- Request name
- Server class
- Request priority
- Request status:
  - W Waiting
  - H Waiting and on hold
  - D Waiting and deferred (if this condition persists, contact your system programmer)
  - B Waiting and blocked (contact your system programmer)
  - A Active
  - F Finished
- File-transfer return and reason codes (if any)
- Number of records transferred during the last completed transfer attempt (if any). The number of records transferred does not apply for OSI file transfers.
- Originator who added the request to the queue
- Date and time that:
  - The request was added to the queue (if status = W, H, D, or B)
  - The transfer started (if status = A)
  - The transfer finished (if status = F).
The file-transfer return and reason codes are described in *NetView FTP Messages and Codes*. Do not confuse the file-transfer return and reason codes (which describe the outcome of attempts to transfer files) with the queue handler command return and reason codes (which describe the outcome of attempts to do such things as add requests to the queue or query requests). If the return or reason code of a waiting or active request is greater than 0, the request was interrupted and then restarted.

### Figure 44. Querying Requests (Batch Jobs and Application Programs)

<table>
<thead>
<tr>
<th>Which Requests Do You Want to Query</th>
<th>Value of Function Parameter</th>
<th>Value of Request Number Parameter</th>
<th>Value of Request Originator Parameter</th>
<th>Value of Request Password Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>All requests that you submitted.</td>
<td>QRYALL or APLQRYALL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted and that have numbers greater than or equal to n.</td>
<td>QRYALL or APLQRYALL</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by another originator.</td>
<td>QRYALL or APLQRYALL</td>
<td></td>
<td>originator ID</td>
<td>master password</td>
</tr>
<tr>
<td>All requests that were submitted by another originator and that have numbers greater than or equal to n.</td>
<td>QRYALL or APLQRYALL</td>
<td>n</td>
<td>originator ID</td>
<td>master password</td>
</tr>
<tr>
<td>All requests in the request queue.</td>
<td>QRYALL or APLQRYALL</td>
<td></td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>All requests that have numbers greater than or equal to n.</td>
<td>QRYALL or APLQRYALL</td>
<td>n</td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>The request that you submitted and that has the number n.</td>
<td>QUERY or APLQRYSR</td>
<td>n</td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>The request that you did not submit and that has the number n.</td>
<td>QUERY or APLQRYSR</td>
<td>n</td>
<td></td>
<td>master password</td>
</tr>
</tbody>
</table>
## Figure 45. Deleting Requests (Batch Jobs and Application Programs)

<table>
<thead>
<tr>
<th>Which Requests Do You Want to Delete</th>
<th>Value of Function Parameter</th>
<th>Value of Request Number Parameter</th>
<th>Value of Request Originator Parameter</th>
<th>Value of Request Password Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>All requests that you submitted, that are not password-protected, and that have finished successfully.</td>
<td>DELALL or APLDELAR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted, that are password-protected, and that have finished successfully.</td>
<td>DELALL or APLDELAR</td>
<td></td>
<td>password assigned to the requests when they were created, or master password</td>
<td></td>
</tr>
<tr>
<td>All requests that were submitted by another originator and that have finished successfully.</td>
<td>DELALL or APLDELAR</td>
<td>originator ID</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is not password-protected, that has the number ( n ), and that has finished successfully.</td>
<td>DELETE or APLDELSR</td>
<td>( n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, that has the number ( n ), and that has finished successfully.</td>
<td>DELETE or APLDELSR</td>
<td>( n )</td>
<td>password assigned to the request when it was created, or master password</td>
<td></td>
</tr>
<tr>
<td>The request that was submitted by another originator, that has the number ( n ) (whether password-protected or not), and that has finished successfully.</td>
<td>DELETE or APLDELSR</td>
<td>( n )</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number ( n ), regardless of its state.</td>
<td>FORCEDEL or APLDELSR</td>
<td>( n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number ( n ), regardless of its state.</td>
<td>FORCEDEL or APLDELSR</td>
<td>( n )</td>
<td>password assigned to the request when it was created, or master password</td>
<td></td>
</tr>
<tr>
<td>The request that was submitted by another originator and that has the number ( n ) (whether password-protected or not and regardless of its state).</td>
<td>FORCEDEL or APLDELSR</td>
<td>( n )</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that you submitted and that have successfully finished.</td>
<td>DELFIN or APLDELSR</td>
<td></td>
<td></td>
<td>master password</td>
</tr>
<tr>
<td>All requests that were submitted by another originator and that have successfully finished.</td>
<td>DELFIN or APLDELSR</td>
<td>originator ID</td>
<td>master password</td>
<td></td>
</tr>
<tr>
<td>All requests that have successfully finished.</td>
<td>DELFIN or APLDELSR</td>
<td></td>
<td>master password</td>
<td></td>
</tr>
</tbody>
</table>
**Figure 46. Restarting Requests (Batch Jobs and Application Programs)**

<table>
<thead>
<tr>
<th>Which Request Do You Want to Restart</th>
<th>Value of Function Parameter</th>
<th>Value of Request Number Parameter</th>
<th>Value of Request Originator Parameter</th>
<th>Value of Request Password Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request that you submitted, that is not password-protected, and that has the number n.</td>
<td>RESTART or APLRSTRT</td>
<td>n</td>
<td></td>
<td>password assigned to the request when it was created, or master password</td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number n.</td>
<td>RESTART or APLRSTRT</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The request that another originator submitted and that has the number n (whether password-protected or not).</td>
<td>RESTART or APLRSTRT</td>
<td>n</td>
<td>originator ID</td>
<td>master password</td>
</tr>
</tbody>
</table>

**Note:** In a restart request, you can also modify the values for NOTBEFORE, NOTAFTER, CLASS, PRIORITY, QSTAT, and RSTPNT.

---

**Figure 47. Modifying Requests (Batch Jobs and Application Programs)**

<table>
<thead>
<tr>
<th>For Which Requests Do You Want to Modify the Class or the Priority</th>
<th>Value of Function Parameter</th>
<th>Value of Request Number Parameter</th>
<th>Value of Request Password Parameter</th>
<th>Value of Server Class or Priority Parameter</th>
<th>Status Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request that you submitted, that is not password-protected, and that has the number n.</td>
<td>MODIFY or APLM FYRQ</td>
<td>n</td>
<td></td>
<td>new class or priority</td>
<td></td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number n.</td>
<td>MODIFY or APLM FYRQ</td>
<td>n</td>
<td>password assigned to the request when it was created, or master password</td>
<td></td>
<td>new class or priority</td>
</tr>
<tr>
<td>The request that another originator submitted and that has the number n (whether password-protected or not).</td>
<td>MODIFY or APLM FYRQ</td>
<td>n</td>
<td>master password</td>
<td></td>
<td>new class or priority</td>
</tr>
</tbody>
</table>

**Note:** You cannot modify the class or priority of a request that has already been modified by an administrator.
<table>
<thead>
<tr>
<th>Which Requests Do You Want to Hold or Release</th>
<th>Value of Function Parameter</th>
<th>Value of Request Number Parameter</th>
<th>Value of Request Password Parameter</th>
<th>Value of Server Class or Priority Parameter</th>
<th>Value of Status Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request that you submitted, that is not password-protected, and that has the number n.</td>
<td>MODIFY or APLMFYRQ</td>
<td>n</td>
<td></td>
<td></td>
<td>HOLD or RELEASE</td>
</tr>
<tr>
<td>The request that you submitted, that is password-protected, and that has the number n.</td>
<td>MODIFY or APLMFYRQ</td>
<td>n</td>
<td>password assigned to the request when it was created, or master password</td>
<td></td>
<td>HOLD or RELEASE</td>
</tr>
<tr>
<td>The request that another originator submitted and that has the number n (whether password-protected or not).</td>
<td>MODIFY or APLMFYRQ</td>
<td>n</td>
<td>master password</td>
<td></td>
<td>HOLD or RELEASE</td>
</tr>
<tr>
<td>An entire server class.</td>
<td>MODIFY or APLMFYRQ</td>
<td></td>
<td>master password</td>
<td>for the Server Class parameter, the number or letter of the class to be held or released</td>
<td>HOLD or RELEASE</td>
</tr>
</tbody>
</table>
Administering the Request Queue

The information about administering the request queue can be obtained by issuing commands known as the administrator query commands. The information provided allows the administrator to take appropriate actions. It is necessary to know the master password to use these commands and perform the appropriate actions.

To query the request queue or query all the requests of a certain status, your batch job or application program must specify the appropriate value for the Function parameter and for the appropriate request parameters. Which parameters you need to specify values for and what values these parameters should have depends on what you are doing. These parameters and values are shown in Figure 49 and are described in the NetView FTP Parameter Reference.

<table>
<thead>
<tr>
<th>What Do You Want to Retrieve Information About</th>
<th>Value of Function Parameter</th>
<th>Value of Request Password Parameter</th>
<th>Value of Status Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>The request queue.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>GEN or APLQGEN</td>
</tr>
<tr>
<td>All waiting requests.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>WAITING or APLQWT</td>
</tr>
<tr>
<td>All waiting requests and the names of the sending files.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>(WAITING,DSN) or APLQWTD</td>
</tr>
<tr>
<td>All active requests.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>ACTIVE or APLQACT</td>
</tr>
<tr>
<td>All active requests and the names of the sending files.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>(ACTIVE,DSN) or APLQACTD</td>
</tr>
<tr>
<td>All finished requests.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>FINISHED or APLQFIN</td>
</tr>
<tr>
<td>All finished requests and the names of the sending files.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>(FINISHED,DSN) or APLQFIND</td>
</tr>
<tr>
<td>All requests.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>ALL or APLQALL</td>
</tr>
<tr>
<td>All requests and the names of the sending files.</td>
<td>QRYADM or APLQRYAD</td>
<td>master password</td>
<td>(ALL,DSN) or APLQALLD</td>
</tr>
</tbody>
</table>

NetView FTP V2.2.1 MVS places the information it retrieves in one of the following:

- For a batch job, in the logfile specified in its DVGLOG DD statement
- For an application program, into the QRA.
NetView FTP V2.2.1 MVS retrieves the following information about the queue:

- The name of the queue handler.
- When the queue handler was started.
- The number of waiting requests in the queue.
- The number of requests in the queue that were formerly active but that were interrupted.
- The number of active requests in the queue.
- The number of finished requests in the queue.
- The number of requests that could still fit in the queue.
- The maximum number of requests the queue can hold.
- For each class:
  - The number of waiting requests
  - The number of active requests
  - Whether a server is started for that class
  - Whether that class is on hold.

NetView FTP V2.2.1 MVS retrieves the following information about each waiting request:

- Request number.
- Request originator.
- Server class and priority.
- Request status:
  - W Waiting
  - H Waiting and on hold
  - D Waiting and deferred (if this condition persists, contact your system programmer)
  - B Waiting and blocked (contact your system programmer).
- Date and time the request was submitted.
- Server name.
- Local LU name or OSI server name.
- Remote server group, remote LU name, or application entity title.
- If the request was restarted.
- Name of the sending file (if requested). For OSI requests, the local file name is retrieved for file transfers with the transfer mode TO, and the OSI file name is retrieved for file transfers with the transfer mode FROM.

NetView FTP V2.2.1 MVS retrieves the following information about each active request:

- Request number.
- Request originator.
- Server class and priority.
- Date and time the request was submitted.
- Date and time the transfer started.
- Server name.
- Local LU name or OSI server name.
- Remote LU name or application entity title.
- Name of the sending file (if requested). For OSI requests, the local file name is retrieved for file transfers with the transfer mode TO, and the OSI file name is retrieved for file transfers with the transfer mode FROM.
NetView FTP V2.2.1 MVS retrieves the following information about each finished request:

- Request number.
- Request originator.
- Server class and priority.
- Date and time the request was submitted.
- Date and time the transfer started.
- Date and time the transfer finished.
- Number of records transferred during the last completed transfer attempt. The number of records transferred is not retrieved for OSI file transfers.
- File-transfer return and reason codes.
- Name of the sending file (if requested). For OSI requests, the local file name is retrieved for file transfers with the transfer mode TO, and the OSI file name is retrieved for file transfers with the transfer mode FROM.
Chapter 6. Examples of Using the NetView FTP V2.2.1 MVS Panels

This chapter gives examples of how you can use the NetView FTP V2.2.1 MVS panels to do the following:

- Send a PS data set, see “Example 1: Sending a PS Data Set”
- Receive a labeled tape file as a PS data set, see “Example 2: Receiving a Labeled Tape File as a PS Data Set” on page 103
- Send members of a PDS, see “Example 3: Sending Members of a PDS” on page 108
- Receive a physical sequential DASD data set as a VSAM cluster, see “Example 4: Receiving a Physical Sequential DASD Data Set as a VSAM Cluster” on page 114
- Send a PS data set to OS/2, see “Example 5: Receiving an OS/2 Text File as a PS Data Set” on page 118
- Receive an OS/2 text file as a PS data set, see “Example 6: Sending a PS Data Set to OS/2” on page 121
- Send a PS data set using OSI/File Services, see “Example 7: Sending a PS Data Set Using OSI/File Services” on page 124.

Example 1 is the simplest. It uses default values wherever possible and does not require you to specify values for any of the additional parameters.

Example 1: Sending a PS Data Set

Assume the following:

- There is a physical sequential data set at your system; it has the name `yourid.ACCOUNT.INFO`, where `yourid` is your user ID.
- You will often want to transfer it to a remote system with the server group name `FTP01`, where a file with the same name and type does not already exist.
- You want the receiving file to have the same type as the sending file, but the name `MAIN.ACCOUNT.INFO`. 
This is how you create a request for this file transfer.

1. Select Option 1 from the Main Task Selection panel (Figure 8 on page 30), and press the ENTER key. NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel (Figure 50).

2. Fill it in as shown in Figure 50.

![Figure 50. Example 1: Main Transfer Parameters Panel](image)

3. Press the ENTER key. NetView FTP V2 MVS displays the Sending File Parameters panel (Figure 51).

4. Fill it in as shown in Figure 51. There are no single quotes around the name of the sending file. This tells NetView FTP V2.2.1 MVS to concatenate your user ID to the front of the file name that you specified. That is, NetView FTP V2.2.1 MVS will send the file with the name `yourid.ACCOUNT.INFO`. You do not have to specify the type of the sending data set. Because the sending file already exists and is at the local system, NetView FTP V2.2.1 MVS can determine its type automatically.
Figure 51. Example 1: Sending File Parameters Panel

5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 52), with the type field already filled.

6. Fill it in as shown in Figure 52.

Figure 52. Example 1: Receiving File Parameters Panel

7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 53).
Because you will often want to conduct this file transfer, you decide first to save this request. To do this, select Option 3 (Save) and press the ENTER key. NetView FTP V2.2.1 MVS displays the Save Current Request panel (Figure 54).

8. Fill it in as shown in Figure 54.

9. Press the ENTER key. NetView FTP V2.2.1 MVS saves the request and displays the Current Request Task Selection panel (Figure 53).

10. You now submit the request by selecting Option 1 (Submit). The queue handler adds the request to the request queue, and displays a message telling you it assigned request number 388 (in this example) to the request.
11. To query the request you submitted, press the **END** PF key twice. NetView FTP V2.2.1 MVS displays the NetView FTP Main Task Selection panel (Figure 8 on page 30).

12. Select Option 3. NetView FTP V2.2.1 MVS displays the Request Control panel (Figure 55).

```
OVGMEQAE ---------------------- Request Control ----------------------
Command ===> 

Select an option by number

1 Query all - Query all requests starting with the specified number and with the specified originator
2 Query - Query the request with the specified number
3 Delete all - Delete all requests of one originator
4 Delete - Delete the request with the specified number
5 Delete finished - Delete all finished requests of one originator
6 Modify - Modify class or priority, or hold or release
7 Force - Force the request with the specified number
8 Restart - Restart the request with the specified number

Request Number ........ ===> 388
Request Originator .... ===>
Request Password ...... ===>
Hold or Release ....... ===> (H=Hold,R=Release)
Server Class ........ ===> (/zerodot to 9; A to Z)
Request Priority ...... ===> (/zerodot to 9)
Restart Point ......... ===> (B=Begin,C=Checkpoint)
PF/zerodot1=HELP PF/zerodot3=END PF/zerodot4=RETURN
```

Figure 55. Example 1: Request Control Panel

13. Select Option 2 and specify the request number you want to query; it is **388** in this example. NetView FTP V2.2.1 MVS displays the Submitted Requests panel, which is filled in as shown in Figure 56.

```
OVGMEQBE --------------------- Submitted Requests ----------- ROW 1 TO 1 OF 1
Command ===> SCROLL ===> CSR
PF01=HELP PF03=END PF04=RETURN PF07=UP PF08=DOWN

Commands: locate refresh sortn sortd sorto 94/1/zerodot/18 17:54
Actions: Cn to change class to n Pn to change priority to n
H to hold R to release D to delete F to force delete
TB to restart from begin TC to restart from checkpoint

Act Numbr Name Cl Pr St Ret Rsn Records Originat Date Time
388 MAINACCT 0 0 W 0 YOURID 94/10/18 17:52:33
```

Figure 56. Example 1: Submitted Requests Panel (Panel 1)
14. Later, after the transfer is finished, you query request number 388 again. The Submitted Requests panel is filled out as shown in Figure 57.

Figure 57. Example 1: Submitted Requests Panel (Panel 2)

15. Look at a list of your saved requests by returning to the NetView FTP Main Task Selection panel (Figure 8), and selecting Option 2. NetView FTP V2.2.1 MVS displays the Saved Requests panel, which is filled out as shown in Figure 58.

Figure 58. Example 1: Saved Requests Panel
16. To view a summary of the transfer parameters you specified, type a v (for view) in the Act column next to the request you want to view and press the ENTER key. NetView FTP V2.2.1 MVS displays the Summary of Transfer Parameters panel, which is shown in Figure 59.

![Figure 59. Example 1: Summary of Transfer Parameters Panel]

17. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Summary of Sending File Parameters panel, which is shown in Figure 60.

![Figure 60. Example 1: Summary of Sending File Parameters Panel]
18. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Summary of Receiving File Parameters panel, which is shown in Figure 61.

![Figure 61: Example 1: Summary of Receiving File Parameters Panel](image)

19. To leave the NetView FTP V2.2.1 MVS panels, press the END PF key until you return to ISPF or TSO.
Example 2: Receiving a Labeled Tape File as a PS Data Set

Assume the following:

- You want to receive a file from a remote NetView FTP V2.2.1 MVS system whose server group name is FTP01.
- The file has the name ACCOUNT.LIST.A100, and is stored on a labeled tape at the remote system.
- The tape is on a tape drive with the volume serial number A15234.
- You want to receive the file to a physical sequential data set stored on an IBM 3380 disk drive that has the volume serial number MS81FP.
- You want to give the file the name ALIST.NEW YORK.MAR3194 and the expiration date 31 December 1994. The file is to be cataloged, if the file transfer was successful.
- You want to assign the receiving file 3 cylinders of primary space and 5 cylinders of secondary space.
- To protect your request, you decide to give it a password. The password you choose is rose.
- You do not want NetView FTP V2.2.1 MVS to process your request before 6 P.M. (18:00) today (31 October 1994).
- After the file has been transferred, you want the local (receiving) transfer program to notify you, and the remote (sending) transfer program to notify the user CARL at LONDON2.
- Your system programmer has told you that requests for files to be transferred from tape are to be given class 3, and must include the words tape check as input to a user-exit routine.
This is how you create a request for this file transfer:

1. Select Option 1 from the NetView FTP Main Task Selection panel (Figure 8 on page 30). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel (Figure 62).

2. Fill it in as shown in Figure 62. The request password is not displayed as you type it in.

```
DVGME0AE ----------------- NetView FTP Main Transfer Parameters ------------------
Command ===> 

Send a file to or retrieve a file from a remote system
Transfer Mode ........... ===> f (T=To,F=From)

Specify one, but not both, of the following:
Remote Server Group ...... ===> ftp/zerodot1
Remote LU Name ........... ===> 
Remote Operating System ... ===> mvs (MVS,VSE,VM,OS4/zerodot,OS2,AIX)

Specify:
Additional transfer parameters? .... y (Y=Yes,N=No)
Request parameters? ............... y (Y=Yes,N=No)
Input for a user-exit routine? ..... y (Y=Yes,N=No)
Post-transfer jobs? ................ N (Y=Yes,N=No)
PF/zerodot1=HELP PF/zerodot3=END PF/zerodot4=RETURN

Figure 62. Example 2: Main Transfer Parameters Panel
```

3. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Transfer Parameters panel (Figure 63).

4. Fill it in as shown in Figure 63.

```
DVGMO0BE ----------------- Additional Transfer Parameters (MVS) ------------------
Command ===> 

Data Encryption .............. ===> N (Y=Yes,N=No)
Data Encryption label ........ ===>
Automatic Transfer Restart ... ===> Y (Y=Yes,N=No)
Restart Point ................. ===> B (B=Begin,C=Checkpoint)
Compression Method ........... ===> A (A=Adaptive,S=SNA,N=None)
Character Data Conversion .... ===> N (Y=Yes,N=No)
Local LU Name ............... ===> 
Report Recipients
Report from sending program ===> xry at ===> stutt3
Report from receiving program ===> carl at ===> london2
Server Running Modes
Sending server ............... ===> C (C=Continuous,S=Single)
Receiving server ............. ===> C (C=Continuous,S=Single)
PF/zerodot1=HELP PF/zerodot3=END PF/zerodot4=RETURN

Figure 63. Example 2: Additional Transfer Parameters Panel
```
5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Request Parameters panel (Figure 64).

6. Fill it in as shown in Figure 64.

```
DVGME0CE ----------------- Request Parameters -----------------------------
Command ===>  
Server class .......... ===> 3 (0 to 9, A to Z)
Request priority ....... ===> 0 (0 to 9)
Request password ....... ===> rose
Not-before date and time ===> 94 / 10 / 31 (yy/mm/dd) 18 : 00 (hh:mm)
Not-after date and time ===> / / (yy/mm/dd) : (hh:mm)
Hold request ........... ===> N (Y=Yes,N=No)
Remote check ........... ===> N (Y=Yes,N=No)
```

**Figure 64. Example 2: Request Parameters Panel**

7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Input for User-Exit Routines panel (Figure 65).

8. Fill it in as shown in Figure 65.

```
DVGME0DE ----------------- Input for User-Exit Routines -----------------------------
Command ===>  
--> tape check
--> 
--> 
--> 
--> 
--> 
--> 
--> 
--> 
--> 
--> 
--> 
PF01=HELP PF03=END PF04=RETURN
```

**Figure 65. Example 2: Input for User-Exit Routines Panel**
9. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 66).

10. Fill it in as shown in Figure 66.

```
DVGME1AE ------------ Sending File Parameters (local,MVS) -------------------
Command ===> 
Data Set Name ===> 'account.list.a100'
DD name ........... ===> for job allocation only
File organization ... ===> ps (VSAM,PS,PO,USER)
Label type ....... ===> sl (SL=Standard Label, NL=No Label)
Volume ............. ===> a15234 specify if data set not cataloged
Coded Character Set ID ..... ===> (5 digits)
Access Security parameters
User ID ..... ===> userid
Password .... ===> a password is specified
Group ID .... ===>
Specify additional parameters for the sending file? ... N (Y=Yes,N=No)
PF/zerodot1=HELP PF/zerodot3=END PF/zerodot4=RETURN
```

Figure 66. Example 2: Sending File Parameters Panel

11. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 67).

12. Fill it in as shown in Figure 67. Clear the label type field.

```
DVGMEBAE ----------- Receiving File Parameters (local,MVS) -------------------
Command ===> 
Data Set Name ===> 'alist.newyork.mar3194'
DD name ........... ===> for job allocation only
File organization . ===> ps (VSAM,PS,PO)
Label type ........ ===> (SL=Standard label, NL=No label for tapes)
Access security parameters
User ID ..... ===> 
Password .... ===> a password is specified
Group ID .... ===>
Specify additional parameters for the receiving file? ... y (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

Figure 67. Example 2: Receiving File Parameters Panel
13. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Parameters for the Receiving File panel (Figure 68).

14. Fill it in as shown in Figure 68.

**Figure 68. Example 2: Additional Parameters for the Receiving File Panel**

15. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Parameters for a New Receiving File panel (Figure 69).

16. Fill it in as shown in Figure 69.

**Figure 69. Example 2: Parameters for a New Receiving File**

17. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel. You have told NetView FTP V2.2.1 MVS everything it needs to know to conduct the file transfer. The request is now ready for you to submit or to save.
Example 3: Sending Members of a PDS

Assume the following:

- A PDS with the name **ICE.CREAM.INFO** exists on your system. It contains only the following members:
  - VANILLA
  - MOCHA
  - FUDGE.

- Outdated copies of this data set also exist at the remote systems NEWYORK, RIO, and TOKYO.

- The members are to be transferred to NEWYORK under the following conditions:
  - **VANILLA** If a member with this name already exists in the data set at NEWYORK, it should not be overwritten.
  - **MOCHA** This member should be transferred to the data set at NEWYORK only if a member with the same name already exists there. The existing member should be replaced by the transferred member.
  - **FUDGE** This member should be transferred to NEWYORK regardless of whether a member with the same name already exists there.

- All of those members that have corresponding members in the data set at RIO are to be transferred to that system. Members that do not already exist in the copy of **ICE.CREAM.INFO** at RIO must not be inserted into that data set.

- All the members except MOCHA are to be transferred to TOKYO, regardless of whether members with the same name exist in the copy of **ICE.CREAM.INFO** at that system or not.
This example involves creating and submitting three separate file-transfer requests. This is how you do it:

1. Select Option 1 from the NetView FTP Main Task Selection panel (Figure 8). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel (Figure 70).

2. Fill it in as shown in Figure 70.

   ![Main Transfer Parameters Panel](image)

   **Figure 70. Example 3: Main Transfer Parameters Panel**

3. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 71).

4. Fill it in as shown in Figure 71. You do not need to fill in the type field.

   ![Sending File Parameters Panel](image)

   **Figure 71. Example 3: Sending File Parameters Panel**
5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 72). The type field is already filled with the value PO and the name field is blank.

6. Fill it in as shown in Figure 72.

```
DVGMEBAE ----------- Receiving File Parameters (remote,MVS) -------------------
Command ===> 
Data Set Name ===> 
DD name .......... ===> for job allocation only
File organization . ===> PO (VSAM,PS,PO)
Label type ....... ===> (SL=Standard label, NL=No label for tapes)
Coded Character Set ID ..... ===> (5 digits)
Access Security parameters
User ID .......... ===>
Password ......... ===>
Group ID ......... ===>
Specify additional parameters for the receiving file? ... N (Y=Yes,N=No)
```

Figure 72. Example 3: Receiving File Parameters Panel

7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 73).

```
DVGMEOKE ----------------- Current Request Task Selection -------------------
Command ===> 
Select an option by number:
  1 Submit - Submit the current request
  2 Change - Change the current request
  3 Save - Save the current request
  4 View - View a summary of the current request
```

Figure 73. Example 1: Current Request Task Selection Panel
8. Select Option 1 (Submit). NetView FTP V2.2.1 MVS displays the PDS List Type panel (Figure 74).

```
DVGME1PE ----------------------- PDS List Type --------------------------------
Command ===>

Select one of the following options:

--- 1  Selection - Displays list from which you can select
      the members you want to transfer

--- 2  Exclusion - Displays list from which you can select
      the members you do not want to transfer

PF01=HELP PF03=END PF04=RETURN
```

Figure 74. Example 3: PDS List Type Panel

9. Select Option 1. NetView FTP V2.2.1 MVS displays the PDS Members panel (Figure 75).

10. Fill it in as shown in Figure 75, and then press the ENTER key.

```
DVGME1SE ---------------------- PDS Members ------------------------------------
Command ===> SCROLL === > PAGE

Data Set Name: ICE.CREAM.INFO 94/10/18 15:56
Actions: I for insert  U for update  R for replace

Act Name Rename VV.MM Created Last Modified Size Init Mod ID
i VANILLA /zerodot1.29 93/11//zerodot4 94//zerodot3/16 14:/zerodot4 156 269 87 YOURID
u MOCHA /zerodot1./zerodot9 93//zerodot9/1/zerodot 94//zerodot3/17 15:16 445 873 3 JOHN
r FUDGE /zerodot1.22 93/1/zerodot/1/zerodot 94//zerodot3/18 /zerodot9:33 632 645 67 PAUL

******************************************************************************
```

Figure 75. Example 3: PDS Members Panel
11. Press the END PF key. NetView FTP V2.2.1 MVS adds the request to the queue, returns the request number, and returns to the Current Request Task Selection panel (Figure 53 on page 98). Now you are ready to create the second request.

12. Select Option 2 (Change). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel (Figure 70 on page 109). It still contains the values you specified for the last request.

13. Change the remote server group to RIO.

14. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 76). It still contains the values you specified for the last request.

15. Fill it in as shown in Figure 76. Specifying an asterisk in parentheses (*) after the data set name tells NetView FTP V2.2.1 MVS to transfer all the members in the PDS that meet the requirement specified in the PDS Option parameter.

```
DVGMEM1AE ------------------ Sending File Parameters (local,MVS) -------------------
Command ===> 
Data Set Name ===> 'ICE.CREAM.INFO(/c5197)' 
DD name ............. ===> for job allocation only
File organization ... ===> PO (VSAM,PS,PO,USER)
Label type ............. ===> (SL=Standard Label, NL=No Label)
Volume .............. ===> specify if data set not cataloged
Coded Character Set ID ..... ===> (5 digits)
Access Security parameters
User ID ..... ===> 
Password .... ===> 
Group ID .... ===> 
Specify additional parameters for the sending file? ... N (Y=Yes,N=No)

PF01=HELP PF03=END PF04=RETURN
```

*Figure 76. Example 3: Sending File Parameters Panel*

16. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 72).

17. Specify y for additional parameters for the receiving file.

18. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Parameters for the Receiving File panel (Figure 77).
19. The conditions under which the members of ICE.CREAM.INFO are to be transferred to RIO correspond to the PDS option UPDATE, so fill in the panel as shown in Figure 77.

![Figure 77. Example 3: Additional Parameters for the Receiving File](image)

20. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel. You have told NetView FTP V2.2.1 MVS everything it needs to know to conduct the file transfer.

21. Select Option 1 (Submit). NetView FTP V2.2.1 MVS adds the request to the queue and returns to the Current Request Task Selection panel. Now you are ready to create the third request.

22. Select Option 2 (Change). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel. It still contains the values you specified for the last request.

23. Change the remote server group to TOKYO.

24. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel. It still contains the values you specified for the last request.

25. Remove the asterisk and parentheses (*) from the name string and press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel. It still contains the values you specified for the last request.

26. Specify n for the additional parameters for the receiving file.

27. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel.

28. Select Option 1 (Submit). NetView FTP V2.2.1 MVS displays the PDS List Type panel.

29. Select Option 2 (Exclusion). NetView FTP V2.2.1 MVS displays the PDS Members panel.

30. Type an X next to the member MOCHA and then press the ENTER key.
Data Set Name: ICE.CREAM.INFO 94/03/18 16:17
Actions: X to exclude  PDS Option ===> INSERT (INSERT,UPDATE,REPLACE)

<table>
<thead>
<tr>
<th>Act Name</th>
<th>VV.MM Created Last Modified Size Init Mod ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>VANILLA</td>
<td>01.29 93/11/04 94/03/16 14:04 156 269 87 YOURID</td>
</tr>
<tr>
<td>MOCHA</td>
<td>01.09 93/09/10 94/03/17 15:16 445 873 3 JOHN</td>
</tr>
<tr>
<td>FUDGE</td>
<td>01.22 93/10/16 94/03/18 09:33 632 645 67 PAUL</td>
</tr>
</tbody>
</table>

Figure 78. Example 3: PDS Members Panel

31. Press the END PF key. NetView FTP V2.2.1 MVS adds the request to the queue and returns to the Current Request Task Selection panel.

Example 4: Receiving a Physical Sequential DASD Data Set as a VSAM Cluster

Assume the following:

- Your system requires the data contained in a physical sequential data set with the name DAT1.SEQDATA. This data set is stored on a DASD at the remote system ROME5.
- Your system requires the data to be in the form of a VSAM cluster with the name DAT1.VDATA. At present this VSAM cluster does not exist. The attributes of DAT1.VDATA are to be the same as the attributes of the VSAM cluster DAT7.VDATA, except that the new cluster must be defined on the volume US0005.
- Make sure that:
  - The transfer program processes the first (initial) transfer successfully.
  - In the future, the VSAM cluster is regularly updated with the data from the file at the remote system.

To do the two tasks mentioned previously, you need to create two separate requests:

- To transfer a file to a new VSAM cluster (a cluster with the file status MUSTNOTEXIST). This request must also contain information about the structure the new cluster is to have. Because you only need to submit this request once, you need not save it.
- To transfer a file to an existing VSAM cluster (a cluster with the file status MUSTEXIST). Because you will be updating the cluster regularly, you should save this request.
This is how you create the requests for these file transfers:

1. Select Option 1 from the NetView FTP Main Task Selection panel. NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel.

2. Fill it in as shown in Figure 79.

3. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 80).

4. Fill it in as shown in Figure 80.
5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 81).

6. Fill it in as shown in Figure 81.

```plaintext
DVGREBAE ------------ Receiving File Parameters (remote,MVS) ---------------
Command ===> 
Data Set Name ===> 'dat1.vdata'
DD name .......... ===> for job allocation only
File organization . ===> vsam (VSAM,PS,PO)
Label type ........ ===> (Sl=Standard label, NL=No label for tapes)
Coded Character Set ID ..... ===> (5 digits)
Access Security parameters
User ID .......... ===> 
Password ......... ===> 
Group ID ........ ===> 
Specify additional parameters for the receiving file? ... y (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

Figure 81. Example 4: Receiving File Parameters Panel

7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Parameters for the Receiving File panel (Figure 82).

8. Fill it in as shown in Figure 82.

```plaintext
DVGREBCE -- Additional Parameters for the Receiving File (MVS;VSAM) ------------
Command ===> 
Cluster Update Password .... ===> 
File Status Option .......... ===> mne (MAY=May, ME=Must, MNE=Must not exist)
File Access Option .......... ===> (SHR=Shared, EXC=Exclusive)
File Processing Option ...... ===> r (M=Merge, R=Replace)
End-of-Processing Option ... ===> (K=Keep, C=Catlg)
in the successful case
End-of-Processing Option ... ===> (K=Keep, C=Catlg, D=Delete)
in the unsuccessful case
KSDS option ................ ===> (MERGE,ADDBEG,ADDKEY,REPKET,DELKEY)
Note: KSDS option will be ignored if file is not key-sequenced
Specify SMS parameters to define a new receiving file? ... N (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

Figure 82. Example 4: Additional Parameters for the Receiving File Panel
9. Press the ENTER key. Because the receiving file does not already exist at the local system, NetView FTP V2.2.1 MVS displays the Parameters for a New Receiving File panel (Figure 83).

10. Fill it in as shown in Figure 83.

![Figure 83. Example 4: Parameters for a New Receiving File Panel](image)

11. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 53 on page 98). You have told NetView FTP V2.2.1 MVS everything it needs to know to conduct the file transfer.

12. Select Option 1 (Submit). NetView FTP V2.2.1 MVS adds the request to the queue and returns to the Current Request Task Selection panel. Now you are ready to create the second request by changing the current request.

13. Go to the Additional Parameters for the Receiving File panel (Figure 82), and change the disposition from MUSTNOTEXIST to MUSTEXIST.

14. Go to the Current Request Task Selection panel. The request is now ready for you to save.

15. Select Option 3 (Save). NetView FTP MVS asks you to specify a name and description for your request. After you do this, press the ENTER key. NetView FTP V2.2.1 MVS returns to the Current Request Task Selection panel.

16. Use the Saved Requests panel (Figure 58 on page 100) to submit the request whenever necessary.
Example 5: Receiving an OS/2 Text File as a PS Data Set

Assume the following:

- You want to receive a file from an OS/2 workstation.
- The LU alias of the NetView FTP Server for OS/2 in the LAN is NFTPOS2.
- The APPC conversation security parameters for the NetView FTP Client for OS/2 are the user ID JOHN and the password FTPACC.
- The file has the name ACCNT10.TXT and is stored in the directory \ACCOUNTS\YEAR1994\ on drive D: at the remote system.
- You want to receive the file to a physical sequential data set stored on an IBM 3380 disk drive that has the volume serial number MS92FP.
- The file's data is to be converted from ASCII to EBCDIC characters.
- You want to give the file the name ACCOUNTS.YEAR1994.OCTOBER.
- You want NetView FTP to have exclusive access to both files during the transfer.
- If the file transfer is successful, both files shall be kept. If it is unsuccessful, the receiving file is to be deleted.

This is how you create a request for this file transfer:

1. Select Option 1 from the NetView FTP Main Task Selection panel (Figure 8 on page 30). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel.

2. Fill it in as shown in Figure 84.

```
DVGME/zerodotAE ------------ NetView FTP Main Transfer Parameters -----------------------
Command ===>
You are changing a request. All previous defined values are still valid
Send a file to or retrieve a file from a remote system
Transfer Mode ............ ===> f (T=To,F=From)
Specify one, but not both, of the following:
Remote Server Group ...... ===> nftpos2
Remote LU Name ........... ===> os2
Remote Operating System ... ===> os2 (MVS,VSE,VM,OS4/zerodot/zerodot,OS2,AIX)

Specify:
Additional transfer parameters? .... y (Y=Yes,N=No)
Request parameters? ............... N (Y=Yes,N=No)
Input for a user-exit routine? ..... N (Y=Yes,N=No)
Post-transfer jobs? ............... N (Y=Yes,N=No)
PFD1=HELP PFD3=END PFD4=RETURN
```

Figure 84. Example 5: Main Transfer Parameters Panel
3. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Transfer Parameters panel (Figure 85).

4. Fill it in as shown in Figure 85.

5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 86).

6. Fill it in as shown in Figure 86.
7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 87).

8. Fill it in as shown in Figure 87.

```
DVGMREBAE ------------ Receiving File Parameters (remote,MVS) ---------------
Command ===>               
Data Set Name ===> 'account.year1994.october'
DD name ............ ===> for job allocation only
File organization . ===> ps (VSAM,PS,PO)
Label type ........ ===> (Sl=Standard label, NL=No label for tapes)
Coded Character Set ID .. ===> (5 digits)
Access Security parameters
User ID ............ ===> 
Password .......... ===> 
Group ID ............ ===> 
Specify additional parameters for the receiving file? ... y (Y=Yes,N=No)
PF/zerodot1=HELP PF/zerodot3=END PF/zerodot4=RETURN
```

Figure 87. Example 5: Receiving File Parameters Panel

9. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Parameters for the Receiving File panel (Figure 88).

10. Fill it in as shown in Figure 88.

```
DVGMREBGE -- Additional Parameters for the Receiving File (MVS;PS) ----------
Command ===>               
File Status Option .......... ===> MNE (MAY=May, ME=Must, MNE=Must not e
File Access Option .......... ===> exc (SHR=Shared, EXC=Exclusive)
File Processing Option ...... ===> R (R=Replace)
End-of-Processing Option ... ===> k (K=Keep, C=Catlg)
in the successful case
End-of-Processing Option ... ===> d (K=Keep, C=Catlg, D=Delete)
in the unsuccessful case
Padding Character .......... ===> (S=Space,N=Null)
Specify:
SMS parameters to define a new receiving file? .... N (Y=Yes,N=No)
Space / DCB / Expiration / Volume parameters? .... y (Y=Yes,N=No)
PF/zerodot1=HELP PF/zerodot3=END PF/zerodot4=RETURN
```

Figure 88. Example 5: Additional Parameters for the Receiving File Panel
11. Press the ENTER key. Because the receiving file does not already exist at the local system, NetView FTP V2.2.1 MVS displays the Parameters for a New Receiving File panel (Figure 89).

12. Fill it in as shown in Figure 89.

![Parameters for a New Receiving File Panel](image)

**Figure 89. Example 5: Parameters for a New Receiving File Panel**

13. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 53 on page 98). You have told NetView FTP V2.2.1 MVS everything it needs to know to conduct the file transfer.

14. Select Option 1 (Submit). NetView FTP V2.2.1 MVS adds the request to the queue and returns to the Current Request Task Selection panel.

**Example 6: Sending a PS Data Set to OS/2**

Assume the following:

- The data set received in example 5 has been updated and you want to transfer it back to the workstation.
- The LU name of the NetView FTP Server for OS/2 in the LAN is **FF1234A**.
- The APPC conversation security parameters for the NetView FTP Client for OS/2 are the user ID **JOHN** and the password **FTPACC**.
- The physical sequential data set has the name **ACCOUNTS.YEAR1994.OCTOBER** and is stored on an IBM 3380 disk drive that has the volume serial number **MS92FP**.
- The name of the OS/2 file is **ACCNT10.TXT** and must already exist in the directory \**ACCOUNTS\*YEAR1994\*\ on drive D: at the workstation.
- If the file transfer is successful, the sending file shall be deleted.
This is how you create a request for this file transfer:

1. Select Option 1 from the NetView FTP Main Task Selection panel (Figure 8 on page 30). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel.

2. Fill it in as shown in Figure 90.

```
DVGME0AE ---------------- NetView FTP Main Transfer Parameters -----------------------
Command """""""

Send a file to or retrieve a file from a remote system
Transfer Mode .......... ===> T (T-To,F-From)

Specify one, but not both, of the following:
Remote Server Group ...... ===>
Remote LU Name ........... ===> ff1234a
Remote Operating System ... ===> os2 (MVS,VSE,VM,OS4,OS4,OS2,AIX)

Specify:
Additional transfer parameters? .... y (Y=Yes,N=No)
Request parameters? .............. N (Y=Yes,N=No)
Input for a user-exit routine? ...... N (Y=Yes,N=No)
Post-transfer jobs? ............... N (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

Figure 90. Example 6: Main Transfer Parameters Panel

3. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Transfer Parameters panel (Figure 91).

4. Fill it in as shown in Figure 91.

```
DVGME0PE ---------------- Additional Transfer Parameters (Non-MVS) -----------------------
Command """""""

APPC security parameters
User ID ..................... ===> john
Password .................... ===> ftpacc

Automatic transfer restart ... ===> Y (Y=Yes,N=No)
Restart Point ............... ===> C (B=Begin,C=Checkpoint)
Compression method .......... ===> A (A=Adaptive,N=None)
Character Data Conversion ... ===> y (Y=Yes,N=No)
Local LU Name ............... ===> 
Local server running mode .... ===> c (C=Continuous,S=Single)
Report recipients
Report from local program: User ID .. ===> 
Node ID ===>
Report from remote program: User ID .. ===> 
Node ID ===>
PF01=HELP PF03=END PF04=RETURN
```

Figure 91. Example 6: Additional Transfer Parameters Panel
5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 92).

6. Fill it in as shown in Figure 92.

![Sending File Parameters Panel](image1)

Figure 92. Example 6: Sending File Parameters Panel

7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Additional Parameters for the Sending File panel (Figure 93).

8. Fill it in as shown in Figure 93.

![Additional Parameters for the Sending File Panel](image2)

Figure 93. Example 6: Additional Parameters for the Sending File Panel
9. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 94).

10. Fill it in as shown in Figure 94.

```
Command  ===> 
File Specification  ===> d:\account\year1994\acct10.txt <-
---->  
---->  
---->  
File Handling Mode .......... ===> r (R=Record, S=Stream)
File Status Option .......... ===> me (MAY=May-, ME=Must-, MNE=Must-not-exist)
File Processing Option ..... ===> r (A=Append, R=Replace)
End-of-Processing Option ... ===> d (K=Keep, D=Delete)
  in the unsuccessful case
Coded Character Set ID ..... ===> (5 digits)
Specify additional parameters for the receiving file? ... N (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

**Figure 94. Example 6: Receiving File Parameters OS/2 Panel**

11. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 53 on page 98). You have told NetView FTP V2.2.1 MVS everything it needs to know to conduct the file transfer.

12. Select Option 1 (Submit). NetView FTP V2.2.1 MVS adds the request to the queue and returns to the Current Request Task Selection panel.

---

**Example 7: Sending a PS Data Set Using OSI/File Services**

Assume the following:

- There is a physical sequential data set at your system; it has the name `yourid.NETWORK.DATA`, where *yourid* is your user ID. In this example, *yourid* is PATTI.

- You want to transfer it to a remote system using the OSI server `OSISRV1`, where a file with the same name does not already exist. The first qualifier (filestore subset) of the MVS file is in the filestore of the OSI server, so the filename is automatically added to that filestore by the OSI server.

- The remote FTAM application has the AET PARISOSI.

- FTAM creates the receiving file name using the name you specify for the OSI file name.
This is how you create a request for this file transfer:

1. Select Option 5 from the NetView FTP Main Task Selection panel (Figure 8 on page 30). NetView FTP V2.2.1 MVS displays the Main Transfer Parameters panel.

2. Fill it in as shown in Figure 95.

```
DVGMOE1E ----------------- NetView FTP OSI Transfer Parameters -------------------
Command ===> 
Send a file to or retrieve a file from a remote system 
Transfer mode ...... ===> T (T=To,F=From)
Copy or move a file 
Initiator function .. ===> C (C=Copy,M=Move)
Issuer identity ...... ===> patti
Destination effect ... ===> C (C=Create,A=Append,R=Replace,
CA=Create/Append,CR=Create/Replace)
Server name ........... ===> OSISRV1 name of OSI server to be used
Report recipient ...... ===> PATTI user-id at ===> node
Specify post-transfer jobs? ... N (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

Figure 95. Example 7: OSI Transfer Parameters Panel

3. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Request Parameters panel (Figure 96).

4. Fill it in as shown in Figure 96.

```
DVGMOE0E ---------------- Request Parameters -------------------------------
Command ===> 
Server class ........... ===> 0 (0 to 9 , A to Z)
Request priority ...... ===> 0 (0 to 9)
Request password ...... ===> 
Not-before date and time ===> 94 / 10 / 31 (yy/mm/dd) 17 : 25 (hh:mm)
Not-after date and time ===> / / (yy/mm/dd) : (hh:mm)
Hold request ........... ===> N (Y=Yes,N=No)
Remote check ........... ===> N (Y=Yes,N=No)
PF01=HELP PF03=END PF04=RETURN
```

Figure 96. Example 7: Request Parameters Panel
5. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Sending File Parameters panel (Figure 97).

6. Fill it in as shown in Figure 97.

<table>
<thead>
<tr>
<th>Command</th>
<th>Filestore owner</th>
<th>Filestore nickname</th>
<th>Filestore password</th>
<th>Filestore subset</th>
<th>Local file name</th>
<th>Local OSI file name</th>
<th>Access passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filestore owner ...</td>
<td>===&gt; OSISRV1</td>
<td>Filestore nickname</td>
<td>===&gt; OSISRV1</td>
<td>Filestore password</td>
<td>===&gt; PATTI</td>
<td>Local file name</td>
<td>===&gt; NETWORK.DATA</td>
</tr>
<tr>
<td>Local OSI file name</td>
<td>===&gt; LOCAL.NETWORK.DATA</td>
<td>Read .............</td>
<td>===&gt;</td>
<td>Read attributes ..</td>
<td>===&gt;</td>
<td>Specify attributes for the receiving file?</td>
<td>Y (Y=Yes,N=No)</td>
</tr>
</tbody>
</table>

PF01=HELP PF03=END PF04=RETURN

---

7. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Receiving File Parameters panel (Figure 98).

8. Fill it in as shown in Figure 98.

<table>
<thead>
<tr>
<th>Command</th>
<th>Filestore</th>
<th>Password</th>
<th>Create password</th>
<th>OSI file name</th>
<th>Access password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filestore</td>
<td>AET ..............</td>
<td>===&gt; PARISOSI</td>
<td>Password</td>
<td>OSI file name</td>
<td>===&gt; patti.network.data</td>
</tr>
<tr>
<td>Create password</td>
<td>===&gt;</td>
<td></td>
<td></td>
<td>OSI file name</td>
<td>===&gt; patti.network.data</td>
</tr>
<tr>
<td>Access password</td>
<td>Read attributes ..</td>
<td>===&gt;</td>
<td></td>
<td>Specify attributes for the receiving file?</td>
<td>Y (Y=Yes,N=No)</td>
</tr>
</tbody>
</table>

PF01=HELP PF03=END PF04=RETURN
9. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Attributes for the Receiving File panel.

10. Fill it in as shown in Figure 99.

```
DVGMEGCE ---- Attributes for the Receiving File (Transfer Mode TO) ------------
Command ---->
FTAM document type .. ===> F1 (F1=FTAM-1,F3=FTAM-3)
Universal class ..... ===> G (G,H,V,I)
Maximum string length ===> 80 (1-19448 bytes)
String significance . ===> F (V=Variable,F=Fixed,U=Unbounded)
Permitted actions: (Y=Yes,N=No)
Read .. ===> Y Delete ........... ===> N
Replace ===> Y Read attributes . ===> Y
Extend ===> Y Change attributes ===> Y
Set passwords for permitted actions: (Y=Yes,N=No)
Y/N Password Y/N Password
Read .. ===> Y Delete ........... ===> N
Replace ===> Y Read attributes . ===> Y
Extend ===> Y Change attributes ===> Y
Future file size .... ===> 3000 (1-2097151 kbytes)
File availability ... ===> I (I=Immediate,D=Deferred)
PFD1=HELP PFD3=END PFD4=RETURN
```

Figure 99. Example 7: Attributes for the Receiving File (Transfer Mode TO) Panel

11. Press the ENTER key. NetView FTP V2.2.1 MVS displays the Current Request Task Selection panel (Figure 53 on page 98). You have told NetView FTP V2.2.1 MVS everything it needs to know to conduct the file transfer.

12. Select Option 1 (Submit). NetView FTP V2.2.1 MVS adds the request to the queue and returns to the Current Request Task Selection panel.
Chapter 7. Sample NetView FTP V2.2.1 MVS Batch Jobs

The batch jobs shown here do the following:

- Add a request to the queue and query all the requests added by the user that submitted the job
- Modify the class of a request and query all the requests added by the user that submitted the job
- Release a request and query all the requests added by the user that submitted the job
- Restart a request
- Force deletion of an unsuccessfully finished request from the queue
- Delete a request from the queue and query all the requests added by the user that submitted the job
- Retrieve information about the queue and about all waiting requests with class S.

Each batch job is followed by a sample of the output it produces. Batch jobs similar to the ones shown here are provided on the NetView FTP V2.2.1 MVS distribution tape and can be obtained from your system programmer.

Sample Batch Job That Adds a Request and Queries All a User’s Own Requests

Figure 100 shows a sample batch job that adds a request and queries all a user’s own requests:

```
//ADD JOB (ACCN,N), 'YOURID-X', MSGCLASS=X, MSGLEVEL=(1,1),
//        CLASS=A, NOTIFY=YOURID, USER=YOURID
.isNull*-------------------------------------------------------------------------------------------*
isNull* JOB THAT ADDS A REQUEST AND QUERIES ALL MY REQUESTS *
isNull*-------------------------------------------------------------------------------------------*
isNull*/STEP1 EXEC PGM=DVGIFBI
isNull*/STEPLIB DD DSN=STOR.XLIB.LOAD, DISP=SHR
isNull*/SYSPRINT DD SYSOUT=*                 
isNull*/SYSUDUMP DD SYSOUT=*                 
isNull*/DVGLOG DD SYSOUT=*                   
isNull*/SYSIN DD *
isNull* FUNCTION=ADD
isNull* XMODE=T
isNull* RMTNODE=NEWYORK
isNull* CLASS=P
isNull* SFILEID='SENDING.DATA.SET'
isNull* RFILEID='RECVING.DATA.SET'
isNull* QSTAT=HOLD
isNull* REQNAME=ADDQRY
isNull*/EOR
isNull* FUNCTION=QRYALL
isNull*/
```

Figure 100. Sample Batch Job That Adds a Request and Queries All a User’s Own Requests
Figure 101 shows the output from the sample batch job that adds a request and queries all a user's own requests:

```
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/10/16 AT 08:18:13; MAINTENANCE LEVEL =
DVG550I NFTP MVS INITIAL VERSION
DVG520I FUNCTION=ADD
DVG520I XMODE=T
DVG520I RMTNODE=NEWYORK
DVG520I CLASS=P
DVG520I SFILEID='SENDING.DATA.SET'
DVG520I RFILEID='RECVING.DATA.SET'
DVG520I QSTAT=HOLD
DVG520I REQNAME=ADDQRY
DVG520I /EOR
DVG563I PARAMETER WITH KEYWORD SFTYPE NOT SPECIFIED; NETVIEW FTP ASSUMES PS
DVG520I
DVG552I QUEUE HANDLER COMMAND ADD CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME =
DVG552I 08:18:16; REQUEST NUMBER = 210
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/10/16 AT 08:18:16; RETURN CODE = 0 ; REASON CODE = 0
DVG520I *************
DVG520I ****************
DVG520I FUNCTION=QRYALL
DVG520I
DVG551I REDNUM=00041;ST=F;PRTY=0;RC=00;RSN=0000;CL=0;FINIS=93/09/14,14:31;ORIG=YOURID ;REC=233 ;REQNAME=RED1
DVG551I REDNUM=00043;ST=W;PRTY=0;RC=00;RSN=0000;CL=0;ADDED=93/09/14,14:31;ORIG=YOURID ;REC= 0 ;REQNAME=RED2
DVG551I REDNUM=00010;ST=F;PRTY=0;RC=00;RSN=0000;CL=8;FINIS=93/09/14,17:32;ORIG=YOURID ;REC=456 ;REQNAME=RED3
DVG551I REDNUM=00210;ST=H;PRTY=0;RC=00;RSN=0000;CL=P;ADDED=94/10/16,08:18;ORIG=YOURID ;REC= 0 ;REQNAME=ADDQRY
DVG520I *************
DVG558I QUEUE HANDLER COMMAND QRYALL CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME = 08:18:16
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/10/16 AT 08:18:16; RETURN CODE = 0 ; REASON CODE = 0
DVG520I *************
DVG520I
```

Figure 101. Sample Output of a Batch Job That Adds a Request and Queries All a User’s Own Requests. The batch job that produced this output is shown in Figure 100. Note that the status of the request that was added (number 25) is H, for waiting and on hold. This is because the QSTAT=HOLD control statement in the batch job caused the request to be held as soon as it is added to the queue.
Sample Batch Job That Modifies a Request’s Class and Queries All a User’s Own Requests

Figure 102 shows a sample batch job that modifies a request's class and queries all a user’s own requests:

```plaintext
//MOD JOB (ACCN,N), 'YOURID-X', MSGCLASS=X, MSGLLEVEL=(1,1),
// CLASS=A, NOTIFY=YOURID, USER=YOURID
#
# JOB THAT MODIFIES THE CLASS OF A REQUEST AND QUERIES ALL
# MY REQUESTS
#
//STEP1 EXEC PGM=DVGIFBI
//STEPLIB DD DSN=STOR.XLIB.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DVLOG DD SYSOUT=*
//SYSIN DD *
FUNCTION=MODIFY
REQNUM=21
CLASS=1
/EOR
FUNCTION=QRYALL
/)
```

Figure 102. Sample Batch Job That Modifies a Request’s Class and Queries All a User’s Own Requests

Figure 103 shows the output from the sample batch job that modifies a request's class and queries all a user's own requests:

```plaintext
DVG020I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/09/17 AT 09:37:18; MAINTENANCE LEVEL =
DVG020I NETVIEW FTP INITIAL VERSION
DVG020I FUNCTION=MODIFY
DVG020I REQNUM=210
DVG020I CLASS=1
DVG020I /EOR
DVG020I *********
DVG552I QUEUE HANDLER COMMAND MODIFY CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME
DVG552I = 09:37:19; REQUEST NUMBER = 210
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 09:37:19; RETURN CODE = 0 ; REASON CODE = 0
DVG020I *********
DVG020I FUNCTION=QRYALL
DVG020I *********
DVG560I REQNUM=00041;ST=F;PRTY=0;RC=0;RSN=0000;CL=0;FINIS=93/09/14,14:31;ORIG=YOURID ; REC=233 ; REQNAME=REQ1
DVG551I REQNUM=00041;ST=W;PRTY=0;RC=0;RSN=0000;CL=0;ADDED=93/09/14,14:31;ORIG=YOURID ; REC= 0 ; REQNAME=REQ2
DVG560I REQNUM=00101;ST=F;PRTY=0;RC=0;RSN=0000;CL=8;FINIS=93/09/14,17:32;ORIG=YOURID ; REC=456 ; REQNAME=REQ3
DVG551I REQNUM=00210;ST=H;PRTY=0;RC=0;RSN=0000;CL=1;ADDED=94/10/16,08:18;ORIG=YOURID ; REC= 0 ; REQNAME=ADDQRY
DVG020I *********
DVG558I QUEUE HANDLER COMMAND QRYALL CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME = 09:37:19
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 09:37:19; RETURN CODE = 0 ; REASON CODE = 0
DVG020I *********
DVG020I *********
```

Figure 103. Sample Output of a Batch Job That Modifies a Request’s Class and Queries All a User’s Own Requests. The batch job that produced this output is shown in Figure 102. Note that the class or the request with number 25 has changed from P to 1.
Sample Batch Job That Releases a Request and Queries All a User’s Own Requests

Figure 104 shows a sample batch job that releases a request and queries all a user’s own requests:

```
//MOD JOB (ACCN,N), 'YOURID-X', MSGCLASS=X, MSGLEVEL=(1,1),
// CLASS=A, NOTIFY=YOURID, USER=YOURID
//*---------------------------------------------*
//* JOB THAT RELEASES A REQUEST AND QUERIES ALL MY REQUESTS *
//*---------------------------------------------*
//STEP1 EXEC PGM=DVGIFBI
//STEPLIB DD DSN=STOR.XLIB.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=/c5197
//SYSUDUMP DD SYSOUT=/c5197
//DVGLOG DD SYSOUT=/c5197
//SYSIN DD /
FUNCTION=MODIFY
REQNUM=1/zerodot
QSTAT=RELEASE
/EOR
FUNCTION=QRYALL
```

Figure 104. Sample Batch Job That Releases a Request and Queries All a User’s Own Requests

Figure 105 shows the output from the sample batch job that releases a request and queries all a user’s own requests:

```
DVG5501 NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/09/17 AT 09:38:26; MAINTENANCE LEVEL -
DVG5501 NFTP MVS INITIAL VERSION
DVG5201 FUNCTION=MODIFY
DVG5201 REQNUM=210
DVG5201 QSTAT=RELEASE
DVG5201 /EOR
DVG5201 ********
DVG5521 QUEUE HANDLER COMMAND MODIFY CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME
DVG5521 = 09:38:26; REQUEST NUMBER = 210
DVG5541 NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 09:38:26; RETURN CODE = 0; REASON CODE = 0
DVG5201 **************
DVG5201 ********
DVG5201 FUNCTION=QRYALL
DVG5201 ********
DVG5601 REQNUM=00841;ST=F;PRTY=0;RC=00;RSN=0000;CL=0;FINIS=93/09/14,14:31;ORIG=YOURID ;REC=233 ;RENAME=RED1
DVG5601 REQNUM=00841;ST=W;PRTY=0;RC=00;RSN=0000;CL=0;ADDED=93/09/14,14:31;ORIG=YOURID ;REC=0 ;RENAME=RED2
DVG5601 REQNUM=00841;ST=W;PRTY=0;RC=00;RSN=0000;CL=0;ADDED=93/09/14,14:31;ORIG=YOURID ;REC=0 ;RENAME=ADDQRY
DVG5201 ********
DVG5581 QUEUE HANDLER COMMAND QRYALL CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME = 09:38:26
DVG5541 NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 09:38:26; RETURN CODE = 0; REASON CODE = 0
DVG5201 **************
DVG5201 ********
```

Figure 105. Sample Output of a Batch Job That Releases a Request and Queries All a User’s Own Requests. The batch job that produced this output is shown in Figure 104. Note that the status of the request with number 25 has changed from H to W.
Sample Batch Job That Restarts a Request

Figure 106 shows a sample batch job that restarts a request:

```plaintext
//RESTART JOB (ACCN,N), 'YOURID-X', MSGLEVEL=(1,1), MSGCLASS=X,
// CLASS=A, USER=YOURID, NOTIFY=YOURID
// Query

///c5197-----------------------------------------------------------/c5197
///c5197 QUERY
///c5197-----------------------------------------------------------/c5197
// BI EXEC PGM=DVGIFBI, REGION=2/zerodot48K
// STEPLIB DD DSN=STOR.XLIB.LOAD, DISP=SHR
// DVLOG DD SYSOUT=*
// SYSUDUMP DD SYSOUT=*
// SYSPRINT DD SYSOUT=*
// SYSPRINT DD SYSOUT=*
// FUNCTION=RESTART
// REQNUM=2
// CLASS=X, QSTAT=HOLD, PRIORITY=5
// RSTPNT=B
/*
```

Figure 106. Sample Batch Job That Restarts a Request

Figure 107 shows the output from the sample batch job that deletes a request:

```plaintext
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/03/03 AT 11:00:16; MAINTENANCE LEVEL -
DVG550I FTP V2 R2 M1
DVG020I FUNCTION=RESTART
DVG020I REQNUM=2
DVG020I CLASS=X, QSTAT=HOLD, PRIORITY=5
DVG020I RSTPNT=B
DVG020I ***************
DVG550I QUEUE HANDLER COMMAND RESTRT CARRIED OUT FOR USER YOURID; RETURN CODE = 0; REASON CODE = 0; TIME = 11:00:17
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/03/03 AT 11:00:17; RETURN CODE = 0; REASON CODE = 0
DVG020I ***************
DVG020I ***************
```

Figure 107. Sample Output of a Batch Job That Restarts a Request. The batch job that produced this output is shown in Figure 106.
Sample Batch Job That Forces Deletion of a Request

Figure 108 shows a sample batch job that restarts a request:

```
//FORCE JOB (ACCN,N), 'YOURID-X', MSGLEVEL=(1,1), MSGCLASS=X, *
// CLASS=A, USER=YOURID, NOTIFY=YOURID
//******************************************************************************
// QUERY
//******************************************************************************
//BI EXEC PGM=DVGIFBI,REGION=2/zerodot48K
//STEPLIB DD DSN=STOR.XLIB.LOAD,DISP=SHR
//DVGLOG DD SYSOUT=/c5197
//SYSUDUMP DD SYSOUT=/c5197
//SYSPRINT DD SYSOUT=/c5197
//SYSIN DD /c5197
FUNCTION = FORCEDEL
REQNUM = 2
/*
```

Figure 108. Sample Batch Job That Forces Deletion of a Request

Figure 109 shows the output from the sample batch job that forces deletion of a request:

```
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/03/03 AT 12:28:53; MAINTENANCE LEVEL =
DVG550I FTP V2 R2 M1
DVG020I FUNCTION = FORCEDEL
DVG020I REQNUM = 2
DVG020I **************
DVG558I QUEUE HANDLER COMMAND FORCE CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME = 12:28:54
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/03/03 AT 12:28:54; RETURN CODE = 0 ; REASON CODE = 0
DVG020I **************
DVG020I **************
```

Figure 109. Sample Output of a Batch Job That Forces Deletion of a Request and Queries All a User's Own Requests. The batch job that produced this output is shown in Figure 108.
Sample Batch Job That Deletes a Request and Queries All a User's Own Requests

Figure 110 shows a sample batch job that deletes a request and queries all a user's own requests:

```plaintext
//DEL JOB (ACCN,N), 'YOURID-X', MSGCLASS=X, MSGLEVEL=(1,1),
// CLASS=A, NOTIFY=YOURID, USER=YOURID
// *---------------------------------------------------------------------*
// * JOB THAT DELETES A REQUEST AND QUERIES ALL MY OWN REQUESTS        *
// *---------------------------------------------------------------------*
//STEP1 EXEC PGM=DVGIFBI
//STEPLIB DD DSN=STOR.XLIB.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=/c5197
//SYSUDUMP DD SYSOUT=/c5197
//SYSIN DD /c5197
//FUNCTION=DELETE
//REQNUM=21
/EOR
//FUNCTION=QRYALL
/**
```

Figure 110. Sample Batch Job That Deletes a Request and Queries All a User's Own Requests

Figure 111 shows the output from the sample batch job that deletes a request and queries all a user's own requests:

```plaintext
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/09/17 AT 11:13:10; MAINTENANCE LEVEL =
DVG550I NFTP MVS INITIAL VERSION
DVG502I FUNCTION=DELETE
DVG502I REQNUM=210
DVG502I /EOR
DVG552I QUEUE HANDLER COMMAND DELETE CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME
DVG552I = 11:13:10; REQUEST NUMBER = 210
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 11:13:10; RETURN CODE = 0 ; REASON CODE = 0
DVG502I **************
DVG502I FUNCTION=QRYALL
DVG502I **************
DVG560I REQNUM=00041;ST=F;PRTY=0;RC=0;RSN=0;CL=0;FINIS=93/09/14,14:31;ORIG=YOURID ;REC=233 ;REQNAME=REQ1
DVG551I REQNUM=00043;ST=W;PRTY=0;RC=0;RSN=0;CL=0;ADDED=93/09/14,14:31;ORIG=YOURID ;REC= 0 ;REQNAME=REQ2
DVG560I REQNUM=00101;ST=F;PRTY=0;RC=0;RSN=0;CL=8;FINIS=93/09/14,17:32;ORIG=YOURID ;REC=456 ;REQNAME=REQ3
DVG502I **************
DVG502I QUEUE HANDLER COMMAND QRYALL CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME = 11:13:10
DVG554I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 11:13:10; RETURN CODE = 0 ; REASON CODE = 0
DVG502I **************
DVG502I **************
```

Figure 111. Sample Output of a Batch Job That Deletes a Request and Queries All a User's Own Requests. The batch job that produced this output is shown in Figure 110. Note that the request with number 25 is no longer shown.
Sample Batch Job That Retrieves Information about the Queue and about All Waiting Requests with Class S

Figure 112 shows a sample batch job that retrieves information about the queue and about all waiting requests with class s:

```
//ADMIN JOB (ACCN,N),'YOURID-X',MSGCLASS=X,MSGLEVEL=(1,1),
//   CLASS=A,NOTIFY=YOURID,USER=YOURID
//*---------------------------------------------*
//* JOB THAT RETRIEVES INFORMATION ABOUT THE REQUEST QUEUE AND *
//* ABOUT ALL THE WAITING REQUESTS OF A SINGLE CLASS *
//*---------------------------------------------*
//STEP1 EXEC PGM=DVGIFBI
//STPLIB DD DSN=STOR.XLIB.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//DVLOG DD SYSOUT=* 
//SYSIN DD *
FUNCTION=QRYADM
QSTAT=GEN
PASSWORD='MASTER'
/EOR
FUNCTION=QRYADM
QSTAT=WAITING
CLASS=S
PASSWORD='MASTER'
/*
```

Figure 112. Sample Batch Job That Retrieves Information about the Queue and about All Waiting Requests with Class S
Figure 113 shows the output from the sample batch job that retrieves information about the queue and about all waiting requests with class S:

```
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/09/17 AT 11:41:43; MAINTENANCE LEVEL =
DVG550I NFTP MVS INITIAL VERSION
DVG500I FUNCTION=QRYADM
DVG500I QSTAT-GEN
DVG500I PASSWORD=MASTER
DVG500I /EOR
DVG500I ********
DVG544I REQUEST QUEUE STATISTICS: QUEUE HANDLER = XQXSTRTO; START DATE = 94/10/16; START TIME = 14:44:55;
DVG544I QUEUE SIZE = 499
DVG544I WAITING = 283; ACTIVE = 1; FINISHED = 5; RESTARTED = 2; UNUSED = 210
DVG544I ********
DVG558I QUEUE HANDLER COMMAND QRYADM CARRIED OUT FOR USER YOURID ; RETURN CODE = 0 ; REASON CODE = 0 ; TIME = 11:41:44
DVG558I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE ENDED ON 94/09/17 AT 11:41:44; RETURN CODE = 0 ; REASON CODE = 0
```

Figure 113. Sample Output of a Batch Job That Retrieves Information about the Queue and about All Waiting Requests with Class S. The batch job that produced this output is shown in Figure 112.
Chapter 8. Sample NetView FTP V2 MVS Application Programs

This chapter contains General-Use Programming Interface and Associated Guidance Information.

The NetView FTP V2 MVS distribution tape contains sample application programs. Ask your system programmer how you can obtain copies of these programs.

The programs are written in PL/I, Assembler, C or REXX. They contain subroutines that:

- Initialize the APL
- Add a request to the queue
- Query the request periodically until the corresponding file transfer is complete (that is, until the request’s status is **finished**)
- Delete the finished request from the request queue.

In the sample programs:

- The DVGAPL control block is coded without the BASED attribute in PL/I and as CSECT in Assembler.
- The DVGQSR control block is coded with the BASED attribute in PL/I and as DSECT in Assembler.
- It is assumed that the storage for the APL is filled with zeros ('X'00') by the system when it passes control to the application program. If your system does not do this (or if you are not sure whether it does this), initialize the APL as described in "Initializing and Assigning Values to the Fields of the APL and APX" on page 68.

The sample programs check and act upon the return code supplied by NetView FTP V2 MVS after the application program interface routine (DVGIFAI) is called and has run. They also declare space for both the query response area and the message area.

Figure 114 shows an example of the output from an application program that adds a request to the request queue, queries all requests, and deletes one request.

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>NAME</th>
<th>CL</th>
<th>PR</th>
<th>ST</th>
<th>RET</th>
<th>RSN</th>
<th>RECORDS</th>
<th>ORIGINATOR</th>
<th>DATE</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>238</td>
<td>FTPSENDW</td>
<td>2</td>
<td>3</td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td>MARY</td>
<td>94/10/15</td>
<td>17:01</td>
</tr>
<tr>
<td>367</td>
<td>FTPPROF1</td>
<td>1</td>
<td>3</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td>JOBNAMEA</td>
<td>94/10/15</td>
<td>17:13</td>
</tr>
<tr>
<td>412</td>
<td>FTPPROF2</td>
<td>1</td>
<td>3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>JOBNAMEA</td>
<td>94/10/15</td>
<td>17:13</td>
</tr>
<tr>
<td>521</td>
<td>MYPROF0X</td>
<td>5</td>
<td>7</td>
<td>F</td>
<td>12</td>
<td>302</td>
<td>30348</td>
<td>JOBNAMEX</td>
<td>94/10/15</td>
<td>12:33</td>
</tr>
<tr>
<td>1344</td>
<td>FTPRECVS</td>
<td>4</td>
<td>9</td>
<td>F</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>YOURID</td>
<td>94/10/15</td>
<td>13:44</td>
</tr>
<tr>
<td>1678</td>
<td>FTPPROF2</td>
<td>2</td>
<td>3</td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td>JOB222</td>
<td>94/10/15</td>
<td>15:55</td>
</tr>
<tr>
<td>1867</td>
<td>MYPROF14</td>
<td>2</td>
<td>3</td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td>CARLB</td>
<td>94/10/15</td>
<td>16:01</td>
</tr>
</tbody>
</table>

**Figure 114. Example of an Output Created by an Application Program**
Appendix. How NetView FTP Modifies Submitted Jobs

Before submitting a job, NetView FTP can override the values of some of the job’s JCL parameters. For example, if the job is to process the receiving data set, NetView FTP can insert the name of the receiving data set into the job.

If you specify the variables that NetView FTP can replace in the JCL code for a job that is to be submitted, NetView FTP replaces the variable with the corresponding value from the file-transfer request before submitting the job to the job entry subsystem. Figure 115 contains a list of all variables that NetView FTP can replace, together with a description of the associated parameter and the names of the DVGAPL fields that the transfer program uses to pass the corresponding values for the sending and receiving systems.

Figure 115. NetView FTP Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Parameter Description</th>
<th>DVGAPL Field for</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sending System</td>
</tr>
<tr>
<td>DVGACUI</td>
<td>Security system user ID of request originator</td>
<td>APLSACUI</td>
</tr>
<tr>
<td>DVGACPW</td>
<td>Security system user password of request originator</td>
<td>APLSACPW</td>
</tr>
<tr>
<td>DVGACGI</td>
<td>Security system group ID of request originator</td>
<td>APLSACGI</td>
</tr>
<tr>
<td>DVGDYFD</td>
<td>Name of data set to be transferred</td>
<td>APLSDYFD</td>
</tr>
<tr>
<td>DVGDYMM</td>
<td>Name of a PDS member to be transferred</td>
<td>APLSDYMM</td>
</tr>
<tr>
<td>DVGDYLI</td>
<td>First volume serial number</td>
<td>APLSDYLI</td>
</tr>
<tr>
<td>DVGRETCD</td>
<td>File transfer return code</td>
<td>APLRC</td>
</tr>
<tr>
<td>DVGRSNCD</td>
<td>File transfer reason code</td>
<td>APLRSN</td>
</tr>
<tr>
<td>DVGTRST</td>
<td>File transfer status</td>
<td>APLRSTAT</td>
</tr>
</tbody>
</table>

The rules for specifying the variables in the JCL statements are the same as the MVS JCL rules.

Variables can be specified anywhere in columns 1 to 71, except in comment lines (lines with the characters /// in columns 1 to 3).

As NetView FTP does not split records, the user must ensure that there is enough space in each input record to hold the replacement value. It is good practice to have a complete JCL record for each variable.
Examples of How NetView FTP Replaces Variables in a JCL Input Member

Assume that NetView FTP is invoked at the receiving system and that it is to submit the following job:

```jcl
//&DVGACUI.A JOB (1234,5),USER=&DVGACUI,PASSWORD=&DVGACPW,
// GROUP=&DVGACGI,NOTIFY=&DVGACUI,MSGCLASS=A,CLASS=A
//STEP EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=&DVGDYFD.(&DVGDYMM),
//       DISP=SHR
//SYSUT2 DD DSN=&DVGACUI..&DVGDYFD..&DVGDYMM,
//       DCB=(DSORG=PS,RECFM=FB,BLKSIZE=3120),
//       UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),DISP=(NEW,CATLG,DELETE)
//SYSIN DD DUMMY
```

Assume also that the following fields of the APL contain these values:

- APLRACUI: 'USER1'
- APLRACPW: 'USERPW1'
- APLRACGI: 'GROUP1'
- APLRDYFD: 'FTP.RECEIVED.PDSABCD'
- APLRDYMM: 'MEMBERXY'

When NetView FTP is called, it replaces the variables in the job with the values from the APL, thereby creating the following job, which it submits:

```jcl
//&USER1A JOB (1234,5),USER=USER1,PASSWORD=USERPW1,
// GROUP=GROUP1,NOTIFY=USER1,MSGCLASS=A,CLASS=A
//STEP EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=FTP.RECEIVED.PDSABCD(MEMBERXY),
//       DISP=SHR
//SYSUT2 DD DSN=USER1.FTP.RECEIVED.PDSABCD.MEMBERXY,
//       DCB=(DSORG=PS,RECFM=FB,BLKSIZE=3120),
//       UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),DISP=(NEW,CATLG,DELETE)
//SYSIN DD DUMMY
```

Notes:

1. Normally, it is not necessary to replace the user ID and password in the job. When NetView FTP submits a job, it is still running in the RACF environment that was created for the file transfer. The submitted job will use the user ID and password of the RACF environment under which it is submitted.

2. When NetView FTP V2.2.1 MVS retrieves a password from RACF, the password is encrypted by RACF and cannot be used to replace a job variable. NetView FTP V2.2.1 MVS retrieves a password from RACF when the user specifies that NetView FTP V2.2.1 MVS is to do so.
Another example:

```assembler
//&DVGACUI.B JOB (1234,5),USER=&DVGACUI,PASSWORD=&DVGACPW,
// GROUP=&DVGACGI,NOTIFY=&DVGACUI,MSGCLASS=A,CLASS=A
//DELETE  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD /c5197
  DELETE '&DVGACUI..&DVGDYFD..&DVGDYMM',PURGE
/*
```

Assuming the same APL field settings, the job created by NetView FTP V2.2.1 MVS looks as follows:

```assembler
//USER1B JOB (1234,5),USER=USER1,PASSWORD=USERPW1,
// GROUP=GROUP1,NOTIFY=USER1,MSGCLASS=A,CLASS=A
//DELETE  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD /c5197
  DELETE 'USER1.FTP.RECEIVED.PDSABCD.MEMBERXY',PURGE
/*
```

### Using a Return Code in a Post-Transfer Job

NetView FTP V2.2.1 MVS provides a program DVGCXAPR, which can be used to set a job step return code in the post-transfer job according to the return code, reason code, or status of the NetView FTP file transfer. Subsequent steps can then be executed under control of the job step return code.

Symbolic parameters can be used in the post-transfer job to represent the file transfer return code (&DVGRETCD), reason code (&DVGRSNCD), and the file transfer status (&DVGTRST).

You invoke DVGCXAPR in the following way:

```assembler
//STEPRC EXEC PGM=DVGCXAPR,PARM='&DVGTRST'
```

DVGCXAPR does the following:

1. Returns code zero when no parameter is provided.
2. Checks that parameter, and returns code 9999 when the parameter is longer than 4 bytes, or if it contains nonnumerics and contains more than one byte.
3. Converts the numeric parameter to fixed binary and loads the character parameter as a return code. For example, 198 for character F, or 230 for character W.
4. Returns the parameter value in register 15 to the caller.
The following is an example of a job that calls DVGCXAPR:

```plaintext
//uid JOB (accn,n),uid,MSGLEVEL=(1,1),MSGCLASS=X,
// CLASS=D,NOTIFY=uid,USER=uid
//-----------------------------------------------------------------------------------------
// translate NFTP return code to step return code
//-----------------------------------------------------------------------------------------
//FTPRETC EXEC PGM=DVGCXAPR,PARM=&DVGRETCD
//SYSPRINT DD SYSOUT=* 
// * INPUT: FTP Returncode
// * OUTPUT: Step return code
//-----------------------------------------------------------------------------------------
// translate NFTP reason code to step return code
//-----------------------------------------------------------------------------------------
//FTPRSN EXEC PGM=DVGCXAPR,PARM=&DVGRSNCD
//SYSPRINT DD SYSOUT=* 
// * INPUT: FTP Reasoncode
// * OUTPUT: Step return code
//-----------------------------------------------------------------------------------------
// translate NFTP queuing status to step return code
//-----------------------------------------------------------------------------------------
//FTPQST EXEC PGM=DVGCXAPR,PARM=&DVGTRST
//SYSPRINT DD SYSOUT=* 
// * INPUT: FTP Reasoncode
// * OUTPUT: Step return code
// * Transfer status finished 'F' generates return code : 198
// * Transfer status deleted 'D' generates return code : 196
// * Transfer status on hold 'H' generates return code : 200
// * Transfer status waiting 'W' generates return code : 230
// * Transfer status active 'A' generates return code : 193
//-----------------------------------------------------------------------------------------
// now start processing depending on NFTP return or reason code
//-----------------------------------------------------------------------------------------
//user1 EXEC PGM=user prog,COND=((/zerodot,GE,FTPRETC),(8,LT,FTPRETC))
//SYSPRINT DD SYSOUT=* 
//-----------------------------------------------------------------------------------------
//user2 EXEC PGM=user prog,COND=((8,GE,FTPRETC),(nnn,EQ,FTPRSN))
//SYSPRINT DD SYSOUT=* 
//-----------------------------------------------------------------------------------------
```
Glossary

This glossary defines many of the terms and abbreviations used with NetView FTP. If you do not find the term you are looking for, refer to the Dictionary of Computing, New York: McGraw-Hill, 1994.

Access Method Services. A utility program that defines VSAM data sets and allocates space for them, converts indexed sequential data sets to key-sequenced data sets with indexes, modifies data set attributes in the catalog, facilitates data set portability between operating systems, creates backup copies of data sets and indexes, helps make inaccessible data sets accessible, and lists data set records and catalog entries.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method.

active request. A request that is presently being scheduled or for which the corresponding file transfer is in progress.

adaptive compression. A method by which the amount of storage required for data can be reduced by replacing character strings that are repeated with references to a directory of such character strings.

added request. A request that has been added to the request queue.

administrator query command. A type of command that causes NetView FTP to retrieve information about the request queue or the requests in it; the information retrieved is intended for the NetView FTP administrator. Contrast with user query command.

administrator query record. A record containing the information NetView FTP retrieves when an administrator issues an administrator query command. Contrast with user query record.

adopted authority. When a program is created, it can specify that the program always runs under the program owner’s user profile. A user does not need authority specifically given to him for the objects used by the program, but uses (adopts) the program owner’s authority. The user has authority for the objects used by the program only when he is running the program and other programs called by the program.

Advanced Communications Function for the Virtual Telecommunications Access Method. An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. VTAM runs under MVS, VSE, and VM/SP, and supports direct control application programs and subsystems such as NetView FTP and VSE/POWER.

Advanced Function Feature. A set of extra functions that extend the capabilities of the NetView FTP V1 MVS base product.

advanced peer-to-peer networking. Data communications support that routes data in a network between two or more APPC systems that are not directly attached.

advanced program-to-program communication. An implementation of the SNA/SDLC LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

AET. Application Entity Title.

AFF. Advanced Function Feature.

alias. An alternative name for a member of a partitioned data set.

AMS. Access Method Services.

APAR. Authorized program analysis report.

APF. Authorized Program Facility.

APL. Application program parameter list.

APPC. Advanced program-to-program communication.

Application Entity Title. The name by which an OSI application (and filestore) can be addressed by remote users. By contrast, the local name of the filestore is the filestore nickname.

application program parameter list. A control block used by application programs to pass parameter values to NetView FTP.

application program parameter list extension. A control block used by application programs to pass parameter values to NetView FTP; a supplement to the application program parameter list.

APPN. Advanced peer-to-peer networking.

APX. Application program parameter list extension.

attended mode. An operating mode of NetView FTP for Workstations that assumes that a user at a workstation expects to be informed about the success of transfers and is available to load devices such as diskettes or tapes. Contrast with unattended mode.
**automatic logon retry.** NetView FTP’s method for eventually establishing a conversation with a remote system. It is used when a server cannot initiate a conversation with a server at a remote system because:

- All the servers at the remote system are busy with other transfers
- None of the servers at the remote system has been started
- ACF/VTAM is temporarily unable to find a path between the two servers.

The server at the local system automatically changes the status of the request from **active** back to **waiting**, and goes on to process the next request in the request queue (MVS, VSE, or VM) or request database (OS/400). Later, NetView FTP tries again to process the request. It keeps trying until it succeeds in initiating a conversation.

**automatic transfer restart.** NetView FTP’s method for automatically restarting a file transfer that was interrupted. In the following situations, NetView FTP is not able to recover a conversation:

- The queue handler or server at either system is terminated
- The server at either system cannot allocate the file being transferred
- A pre-transfer user-exit routine at either system rejected the file transfer
- There is a severe or prolonged conversation outage.

However, when you create a request you can specify that if one of these situations arises, NetView FTP is to change the status of the request back to **waiting**. The servers at your system then periodically reprocess the request until one of them succeeds in restarting the transfer.

**basic partitioned access method.** An access method that can be used to create program libraries in direct-access storage for convenient storage and retrieval of programs.

**basic sequential access method.** An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access storage device.

**BIND.** Bind Session.

**Bind Session.** In SNA, a request to activate a session between two logical units.

**blocked request.** A waiting request that is trapped in the queue. The request was passed to the queue handler at a time when there was enough CSA storage available to NetView FTP to do so. However, in the meantime, the limit of the amount of CSA storage available to NetView FTP was reduced and is not enough to allow the queue handler to pass the request to a server for processing. The request cannot be processed until the limit of the amount of CSA storage available to NetView FTP is raised.

**BPAM.** Basic partitioned access method.

**BSAM.** Basic sequential access method.

**callback.** In the AIX operating system, a procedure that is called if and when certain specified conditions are met. This is accomplished by specifying the procedure in a callback list.

**callback list.** (1) A list of procedures that are called if and when certain specified conditions are met. (2) In the program IBM AIXwindows Environment/6000, individual widgets can define callback lists.

**CCB.** Command control block.

**CCI.** Common control block insertion.

**CCSID.** Coded character set identifier.

**CCW.** Channel command word.

**CDC.** Character data conversion.

**CDRA.** Character Data Representation Architecture.

**CDRM.** Cross-domain resource manager.

**CDRS.** Cross-domain resource.

**character data.** Data in the form of letters, numbers, and special characters, such as punctuation marks, that can be read by the human eye.

**Character Data Representation Architecture.** An IBM architecture that defines a set of identifiers, services, supporting resources, and conventions to achieve a consistent representation, processing, and interchange of graphic character data in SAA environments.

**character literal.** A symbol, quantity, or constant in a source program that is itself data, rather than a reference to data.

**checkpoint.** A point at which information about the status of a file transfer is recorded. If the file transfer is interrupted, NetView FTP can use this information to resume the file transfer from a point near where the interruption occurred instead of from the beginning of the file.

**checkpoint data set.** A data set that contains information about the current status of an active file.
transfer. If the file transfer is interrupted, NetView FTP can use this information to resume the file transfer from a point near where the interruption occurred instead of from the beginning of the file.

**checkpoint file.** A synonym for *checkpoint data set*.

**checkpoint record.** A record of a checkpoint data set or file. One checkpoint record contains the information needed to restart one file transfer from a checkpoint.

**checkpoint/restart data set.** A deprecated term for *checkpoint data set*.

**checkpoint/restart record.** A deprecated term for *checkpoint record*.

**class.** See *server class*.

**client.** On a local area network, a workstation that requests service from a server workstation.

**CLIST.** Command list.

**Coded character set identifier.** In NetView FTP, an identifier that represents a set of graphic characters and their code point assignment. The coded character set identifier defines how characters are mapped to decimal values.

**command control block.** A control block that contains details of the queue handler command to be carried out.

**command list.** A list of commands and control statements that is assigned a name. When the name is invoked (as a command) the commands in the list are executed.

**common service area.** In MVS, a part of the common area that contains data areas that are addressable by all address spaces. During use, these areas are protected by the key of the requester.

**completion user exit.** Deprecated term for *post-transfer user exit*.

**compression.** A technique for converting data into a form that requires less storage space and less transmission time than its original form. Contrast with *decompression*. See also *SNA compaction* and *SNA compression*.

**condition code.** A 4-digit decimal value derived from the value a server places in register 15 before returning control to the operating system. The digits of the condition code consist of the server return code and, for a server running in single mode, the file-transfer return code.

**continuous mode.** A server running mode in which a server continues running after it has transferred a file.

**control language.** The set of all commands with which a user requests system functions.

**control point.** A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network.

**control statement.** A statement that controls or affects the running of a program. For example, \texttt{NOTAFTER=('21:34','94/12/25')} is a control statement that assigns the value \texttt{('21:34','94/12/25')} to the parameter represented by the keyword \texttt{NOTAFTER}.

**conversation.** In SNA, a connection between two transaction programs over an LU-LU session that lets them communicate with each other while processing a transaction.

**CP.** Control point.

**CP-CP sessions.** The parallel sessions between two control points, using LU 6.2 protocols.

**Cross-domain resource.** In VTAM programs, synonym for *other-domain resource*.

**Cross-domain resource manager.** The functions of the system services control point (SSCP) that control initiation and termination of cross-domain resources.

**cryptographic key.** A value used to encrypt and decrypt data transmitted in an LU-LU session that uses cryptography.

**cryptography.** The transformation of data to conceal its meaning.

**CSA.** Common service area.

**current request.** The request currently being created or changed by the user of the NetView FTP panels.

**daemon.** In the AIX operating system, a program that runs unattended to perform a standard service.

**daemon process.** In the AIX operating system, a process begun by the root user or the root shell that can be stopped only by the root user. Daemon processes generally provide services that must be available at all times.

**DASD.** Direct-access storage device.

**data control block parameters.** The following parameters: Record Format, Logical Record Length, and Physical Block Size.
**data set control block.** A control block containing specifications for data sets that are to be created.

**data transfer message unit.** The message unit used to send the data object.

**DCB parameters.** Data control block parameters.

**decompression.** A technique for converting compressed data back into its original form. Contrast with compression.

**default first qualifier.** Server initialization parameter. Sometimes referred to as the **GID initialization parameter.** The server uses this parameter as the first qualifier when it creates a name for the data set.

**default value.** The value that is assigned to a parameter by a program if no value is specified by a user.

**deferred request.** A waiting request that is temporarily trapped in the queue. CSA storage was obtained for NetView FTP, but not enough for the queue handler to pass the request to a server for processing. The request is processed later, when more CSA storage is available to NetView FTP.

**delay time.** The amount of time a finished request stays in the request queue before rebuilding the request queue causes it to be deleted automatically.

**direct transfer.** Transfer of data from one file to another file without first storing the data in an intermediate file.

**direct-access storage device.** A storage device for which access time is effectively independent of the location of the data being accessed.

**directory file.** In the AIX operating system, a file that contains information the system needs to access all types of files.

**distribution-service component.** A component of NetView FTP VM that handles communication with the queue handler (such as retrieving requests to be processed), and with the network (such as establishing conversations and transferring files). In NetView FTP VM, each server consists of one distribution-service component and up to 32 file-service components.

**distribution-service machine.** With NetView FTP VM, a virtual machine in which a distribution-service component runs.

**DSCB.** Data set control block.

**DTMU.** Data transfer message unit.

**dynamic allocation.** The allocation of a file when it is needed, not in advance. Contrast with job allocation.

**encrypt.** To scramble data or convert data to a secret code that masks the meaning of the data to any unauthorized recipient.

**entry sequence.** The order in which records are physically arranged in auxiliary storage.

**entry-sequenced data set.** A data set whose records are loaded without respect to their contents, and whose relative byte addresses cannot change.

**ESDS.** Entry-sequenced data set.

**ESTAE.** Extended specify task abnormal exit.

**exceptional checkpointing.** To take a checkpoint when certain types of errors occur.

**exchange identification.** The ID that is exchanged with the remote physical unit when an attachment is first established.

**exclude members.** To choose those members of a PDS that are not to be transferred. Contrast with select members.

**exit.** A point in a program at which control is passed to another program.

**exit routine.** A routine that receives control when a specified event occurs.

**Exit(n) Message Unit.** The message unit used to convey information provided by the Post-Transfer User Exit routine n (where n is 1 or 2) of the sender to the receiver.

**extended specify task abnormal exit.** A macroinstruction that allows a user to intercept a scheduled abnormal termination.

**FAT.** File allocation table.

**FBA.** Feedback area.

**feedback area.** An area of storage containing information related to a queue handler command. For example, an FBA can contain a request control block, a query data area, or a server data area.

**file allocation table.** A table used by DOS and OS/2 to allocate space on a disk for a file and to locate and chain together parts of the file that may be scattered on different sectors so that the file can be used in a random or sequential manner.

**file group.** One or more files that reside on one system. For example, all files that are stored in the
same directory or whose file names consist of partly matching character strings are considered a file group.

**file pool.** A collection of minidisks managed by SFS. It contains user files and directories and associated control information. Many user's files and directories can be contained in a single file pool.

**file transfer.** The sending and receiving of the contents of a file.

**File Transfer Access and Management.** A set of programs, such as OSI/File Services, which conforms to FTAM standards to manage and transfer files over an OSI network.

**file-service component.** A component of NetView FTP VM that handles file access and the taking of checkpoints. In NetView FTP VM, each server consists of one distribution-service component and up to 32 file-service components.

**file-transfer completion message.** A message, sent by a server to a user after a file transfer, which describes the outcome of a file transfer.

**file-transfer report.** A file, sent by a server to a user after a file transfer which describes the outcome of a file transfer.

**file-transfer request.** A list of parameters and their values that tell NetView FTP (1) that it is to transfer a file from one system to another, and (2) about the file transfer and the sending and receiving data sets.

**filestore.** See local filestore and remote filestore.

**filestore nickname.** The name of the filestore at the local level. It is defined by the filestore owner when registering the filestore in the LRD, and is used by authorized local users to access that filestore.

**filestore owner.** The single user, local or remote, who has created the filestore and who controls the passwords for accessing it and the filestore accessibility. Each OSI/File Services user owns one filestore.

**filestore subset.** A subdivision of the local filestore. It is the first qualifier of the MVS data-set name. Each local file is registered in the LRD under a related filestore subset, which in turn belongs to a local filestore.

**filter.** In the AIX operating system, a command that reads standard input data, modifies the data, and sends it to the display screen.

**finished request.** A request for which the corresponding file transfer has finished, whether successfully or unsuccessfully.

**FIU.** File Interchange Unit.

**FTAM.** File Transfer Access and Management.

**FTP level.** A character that represents the level of sophistication of an FTP or NetView FTP program.

**FTP V2.** File Transfer Program Version 2 Release 2.

**FSB.** NetView FTP shared block.

**FSBX.** NetView FTP shared block extension.

**GDG.** Generation data group.

**generation data group.** A collection of data sets kept in chronological order; each data set is a generation data set.

**generation data set.** One generation of a generation data group.

**GETVIS area.** Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

**GID initialization parameter.** See default first qualifier.

**GUI.** The graphical user interface of NetView FTP for Workstations.

**handle.** (1) In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and a file so that the file can be found and opened. (2) In the AIX operating system, a data structure that is a temporary local identifier for an object.

**HDAM.** Hierarchic direct access method.

**hex.** Abbreviation of hexadecimal.

**hierarchic direct access method.** A database access method that uses algorithmic addressability of records in a hierarchic direct organization.

**hierarchic indexed sequential access method.** A database access method that uses indexed access to records in a hierarchic sequential organization.

**High Performance File System.** A file organization available under OS/2.

**HISAM.** Hierarchic Indexed Sequential Access Method.

**HPFS.** High Performance File System.

**ICCF.** Interactive computing and control facility.

**ICF.** Intersystem communications function.
IMS/VSe Information Management System/Virtual Storage.

independent LU. A logical unit (LU) that does not receive an ACTLU over a link. Such LUs can act as primary logical units (PLUs) or secondary logical units (SLUs) and can have one or more LU-LU sessions at a time.

INI file. See initialization file.


input field. An area on a panel in which data is entered.

instance. In the AIX operating system, the concrete realization of an abstract object class. An instance of a widget or gadget is a specific data structure that contains detailed appearance and behavioral information that is used to generate a specific graphical object on-screen at run time.

Internet. A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses /IP as the standard for transmitting information.

Internet Protocol. A protocol used to route data from its source to its destination in an Internet environment.

Intersystem communications function. Communications between application programs on an AS/400 system and an application program on a remote system are accomplished using the AS/400 system intersystem communications function (ICF) and the underlying support.

IP. Internet Protocol.

ISPF. Interactive System Productivity Facility.

JCL. Job control language.

JES. Job entry subsystem.

job allocation. The allocation of a file by a server startup job. The allocation takes place when a server is started, which is before (sometimes long before) the file transfer takes place. Contrast with dynamic allocation.

job control language. A control language used to identify a job to an operating system and to describe the job's requirements.

key sequence. In VSAM, the collating sequence of data records as determined by the value of the key field in each record.

key-sequenced data set. A VSAM data set whose records are loaded in key sequence and controlled by an index.

keyword. A part of a control statement that consists of a specific character string.

KSDS. Key-sequenced data set.

LAN. Local area network.

LAN gateway. A functional unit that connects a local area network with another network using different protocols.

LDS. Linear data set.

LEN node. Low-entry networking node. That is a node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

linear data set. A VSAM data set that contains data but no control information. A linear data set can be accessed as a byte-string in virtual storage. A linear data set has no records and a fixed control interval size of 4096 bytes.

local. Refers to one's own system.

local area network. A data network located on the user's premises in which serial transmission is used for direct data communication among workstations.

local filestore. A collection of local files. Each local filestore is registered in the LRD with a filestore nickname for local access, and a filestore AET for remote access.

Local Resource Directory. The file containing information on local users, local filestores, filestore subsets, and local files necessary for OSI/File Services to run initiator and responder functions.

local-request handler. A server subtask that can process a request submitted at the local system and can initiate a conversation.

log file. A file to which a NetView FTP component writes messages.

logical unit. In SNA, a port through which an end user accesses an SNA network. Each NetView FTP server is a logical unit.

logical unit name. A name used to represent the address of a logical unit.

LRD. Local Resource Directory.
LU. Logical unit.

LU name. Logical unit name.

LU 0 conversation. The type of conversation NetView FTP uses for file transfers between a node where NetView FTP V2.1 MVS, NetView FTP V1 VM, or NetView FTP V1 VSE is installed and a node where NetView FTP V1 MVS or FTP V2 is installed.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network.

manual transfer restart. NetView FTP's method for allowing a user to restart a file transfer that was interrupted by submitting a restart request for that file transfer.

master password. A password, set by the NetView FTP system programmer that lets those who specify it query, modify, or delete any request in the request queue, regardless of whether or not it is password-protected, and regardless of who added it to the queue.

message area. The area of storage to which NetView FTP writes the messages it issues to an application program.

mode. The session limits and common characteristics of the session associated with advanced program-to-program (APPC) devices managed as a unit with a remote location.

mode description. A system object created for advanced program-to-program (APPC) devices that describes the session limits and the characteristics of the session, such as the maximum number of sessions allowed, maximum number of conversations allowed, and other controlling information for the session.

MVS node. A node with MVS as its operating system.

MVS system. A system with MVS as its operating system.

NETBIOS. Network Basic Input/Output System. An operating system interface for application programs used on IBM personal computers that are attached to the IBM Token-Ring Network.

NetView FTP administrator. Someone who knows the master password. A NetView FTP administrator can query, delete, modify, hold, or release any request regardless of whether it is password-protected, and regardless of who submitted it.

NetView FTP application program. An application program that adds, queries, modifies, or deletes a request, or that retrieves information about NetView FTP.

NetView FTP batch job. A batch job that adds, queries, modifies, or deletes a request, or that retrieves information about NetView FTP.


NetView FTP MVS. NetView File Transfer Program for MVS.


NetView FTP partition. A VSE partition that contains the main components of NetView FTP VSE.

NetView FTP shared block. An area of CSA storage that is used to pass data between the components of NetView FTP. Any data that does not fit in the FSB is put in the NetView FTP shared block extension.

NetView FTP shared block extension. An area of ECSA storage that is used to pass data between the components of NetView FTP. It contains any data that does not fit in the NetView FTP shared block.

NetView FTP VM. NetView File Transfer Program for VM.

NetView FTP VSE. NetView File Transfer Program for VSE.

network. An interconnected group of nodes.

network drive. With NetView FTP, it is a shared resource that can be accessed from each workstation in the LAN.

network job entry facility. A facility that uses the network job-interface (NJI) protocols to allow a computer system to communicate with other computer systems in a network.

NFTP directory. The directory that contains the NetView FTP/2 product files.

NFTPWORk directory. The directory that contains all NetView FTP/2 work files, for example, the NetView FTP message and log files.

NJE. Network job entry.

NJI. Network job-interface.
node.  An endpoint in a link, or a junction common to two or more links in a network. A deprecated term for server group.

node ID.  Deprecated term for server group.

node ID table.  Deprecated term for server group table.

not-after time.  The time after which NetView FTP is not to process a request.

not-before time.  The time before which NetView FTP is not to process a request.

numeric literal.  A numeric character or string of numeric characters whose value is implicit in the characters themselves; for example, 777 is the literal as well as the value of the number 777.

octal.  Pertaining to a selection, choice, or condition that has eight possible different values or states.

OEM.  Original equipment manufacturer.

Open Systems Interconnection.  The seven-layer communications architecture used for the definition of protocol standards for networks.

operation mode.  See attended mode and unattended mode.

operational key.  Deprecated term for cryptographic key.

Original equipment manufacturer.  A manufacturer of equipment that may be marketed by another manufacturer.

originator ID.  A string of characters that identifies the job, started task, or user that added a request to the request queue.

OSI.  See Open Systems Interconnection.

other-domain resource.  A recommendation for a logical unit that is owned by another domain and is referenced by a symbolic name, which can be qualified by a network identifier.

panel.  A predefined image displayed on a terminal screen.

panel flow.  The way in which panels are chained together so that a user can move from one to another.

panel layout.  The way in which the text and the input fields on a panel are arranged.

partitioned data set.  A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

PCF.  Programmed Cryptographic Facility.

PDS.  Partitioned data set.

path information unit.  In SNA, a message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment.

phase.  The smallest unit of executable code that can be loaded into virtual storage.

ping.  The use of the ping command to send an echo request to a network host or gateway.

port.  With NetView FTP, the communication end point in TCP/IP. A port is identified by a port number.

port number.  In TCP/IP, a 16-bit number used to communicate between TCP/IP and a higher-level protocol or application.

post-conversation user exit.  A user exit that passes control to a routine at the system at which the file-transfer request originated. This routine is to run just after the servers terminate their conversation.

post-transfer user exit.  A user exit that passes control to a routine that is to run just after a server closes a file that has been transferred.

pre-queuing user exit.  A user exit that passes control to a routine that is to run just after a request is submitted to NetView FTP and just before NetView FTP adds the request to the queue.

pre-transfer user exit.  A user exit that passes control to a routine that is to run just before a server opens a file that is to be transferred.

preparation user exit.  Deprecated term for pre-transfer user exit.

process (a request).  To obtain and try to carry out (a request).

program temporary fix.  A temporary solution to bypass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of a program.

PTF.  Program temporary fix.

PUBX.  Physical Unit Control Block Extension.

QDA.  Query data area.

QRA.  Query response area.
QSAM. Queued sequential access method.

QSR. Query Status Record.

query (a request). To ask for information about (a request).

query data area. The area of ECSA storage into which NetView FTP places either user query records or administrator query records.

query response area. The area of storage into which NetView FTP places information it retrieves for an application program.

queue handler. A NetView FTP component that controls access to the request queue. In NetView FTP MVS, the queue handler also controls all other NetView FTP components and all communication with the operator.

queue handler command. A command that a component of NetView FTP issues to the queue handler when it wants the queue handler to do something.

queued sequential access method. An extended version of the basic sequential access method (BPAM). When this method is used, a queue is formed of (1) input data blocks that are awaiting processing or (2) output data blocks that have been processed and are awaiting transfer to auxiliary storage or to an output device.

raw device. In the AIX operating system, a device that treats data I/O as a continuous stream, without consideration for the data’s logical structure. For example, I/O for fixed disks and streaming tapes occurs in units of bytes that have no relationship to characters.

RCB. Request control block.

RCE. Request control element.

RDF. Request definition file.

reason code. A value issued by a program that gives additional information about a situation described by a return code.

receiving data set. A data set in which a copy of a file that has been sent using NetView FTP has been placed. Contrast with sending data set.

receiving file. A data set in which a copy of a file is placed that has been sent using NetView FTP. Contrast with sending file.

receiving system. The NetView FTP system that receives the file being transferred.

regular checkpointing. To take a checkpoint at specified regular intervals.

regular file. In the AIX operating system, a file that contains data. A regular file can be a text file or a binary file. Text files contain information readable by the user. This information is stored in ASCII. Binary files contain information readable by the computer.

relative record data set. In VSAM, a data set whose records are loaded into fixed-length slots and are represented by the relative-record numbers of the slots they occupy.

remote. Pertaining to a system other than one’s own.

remote filestore. A collection of remote files. OSI/File Services users can access a file residing in a remote system only if they specify the filestore AET of the remote filestore, and the OSI file name of the related file.

remote-request handler. A server subtask that can accept a conversation initiated by a local-request handler at a remote system.

report. Synonym for file-transfer report.

report recipient. A user to whom a server sends a file-transfer report.

request. Synonym for file-transfer request.

request class. A deprecated term for server class.

request control block. A control block that contains some or all of a file-transfer request. Data that does not fit in the RCB is put in the request control block extension.

request control block extension. A control block that contains data that does not fit in the request control block.

request control element. An element of the request queue directory.

request database. In NetView FTP/400, the database in which NetView FTP stores all requests.

request definition file. A file containing NetView FTP control statements, that is, all NetView FTP parameters with their appropriate values that are necessary for a file transfer.

request handler. A server subtask. See also remote-request handler.

request number. A number that the queue handler assigns to a request when it adds the request to the request queue and that is used to identify the request.
**request password.** A character string, assigned by a user to a request, that prevents users sharing that user's originator ID from deleting or modifying that user's waiting and active requests. If a user specifies a request password for a request, another user with the same originator ID must specify either the request password or the master password to be able to delete or modify the request while it is waiting, or to delete the request while it is active.

**request priority.** A number, assigned by a user to each request, that determines the order in which a server is to process it. When a server is ready to process a request and several requests are eligible, the server processes the request with the highest priority first.

**request queue.** In NetView FTP for MVS, VSE, or VM, the file in which NetView FTP stores requests that have been submitted for processing.

**request queue directory.** A directory of the contents of the request queue.

**request unit.** A message unit that contains control information, end-user data, or both.

**request-queue user exit.** Deprecated term for pre-queuing user exit.

**requesting system.** The system where the file-transfer request has been initiated.

**resident session partner.** An FTP V2 MVS or FTP V2 VSE server that runs continuously and can perform an indefinite number of file transfers, however, these file transfers must all be initiated by servers at remote systems.

**responding system.** The system responding to a file-transfer request.

**return code.** A value issued by a program that describes the outcome of an operation performed by that program.

**root.** In the AIX operating system, the user name for the system user with the most authority.

**RRDS.** Relative record data set.

**RTM.** Recovery Termination Manager.

**RU.** Request unit.

**RXB.** Request control block extension.

**SAF.** System authorization facility.

**SAW.** System Authorization Work Area.

**SAM.** Sequential access method.

**SAS.** Spool Access Services.

**saved request.** A request that has been created with the NetView FTP panels and that has then been saved in an ISPF table data set.

**scheduling a request.** Determining which request is to be obtained, obtaining it, and passing it to a server.

**scroll amount.** The amount that the list on a panel is scrolled up or down when you enter the **UP** or **DOWN** command.

**SDA.** Server data area.

**SDMU.** Source description message unit.

**SDWA.** System Diagnostic Work Area.

**select members.** To choose those members of a PDS that are to be transferred. Contrast with **exclude members**.

**sending data set.** A data set, a copy of which is to be transferred using NetView FTP. Contrast with **receiving data set**.

**sending file.** A file of which a copy is to be transferred using NetView FTP. Contrast with **receiving file**.

**sending system.** The NetView FTP system stores the file that will be transferred to the receiving system.

**sequential access method.** See basic sequential access method.

**server.** (1) A NetView FTP component that establishes or accepts conversations and that transfers files. (2) With NetView FTP for Workstations, a NetView FTP Server program that serves as a gateway for one or more NetView FTP Client programs.

**server class.** A number or letter, assigned by a user to a request, that specifies which servers can process that request.

**server data area.** The area of ECSA into which NetView FTP places information about a server or servers.

**server group.** A group of servers (logical units).

**server group table.** A data set that specifies which servers make up each server group, and that contains information about each server.

**server modification area.** An area of storage that contains the modifications to a server's session parameters.
**session.** In SNA, a logical connection between two network-addressable units.

**SFS.** Shared file system.

**SFS directory.** A group of files. SFS directories can be arranged to form a hierarchy in which one directory can contain one or more subdirectories as well as files.

**shared file pool.** See file pool.

**shared file system.** A part of CMS that lets users organize their files into groups known as directories, and selectively share those files and directories with other users.

**single mode.** A server running mode in which a server stops running after it has transferred a file (or attempted to transfer a file and failed).

**slot.** A space in the request queue directory that is able to hold a request control element (RCE).

**SMA.** Server modification area.

**SMF.** System management facilities.

**SMS.** Storage Management Subsystem.

**SNA.** Systems Network Architecture.

**SNA compaction.** The transformation of data by packing two characters in a byte that normally would only hold one character.

**SNA compression.** The replacement of a string of up to 64 repeated characters by an encoded control byte to reduce the length of the string.

**SNA network.** In SNA, the part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units, boundary function components, and the path control network.

**SNA node.** A node that supports SNA protocols.

**socket.** In the AIX operating system: (1) A unique host identifier created by the concatenation of a port identifier with a TCP/IP address. (2) A port identifier. (3) A 16-bit port number. (4) A port on a specific host; a communications end point that is accessible through a protocol family’s addressing mechanism. A socket is identified by a socket address. See also socket address.

**socket address.** In the AIX operating system, a data structure that uniquely identifies a specific communications end point. A socket address consists of a port number and a network address. It also specifies the protocol family.

**SRMU.** Statistics report message unit.

**special file.** In the AIX operating system, a file that defines a FIFO (first-in, first-out) file or a physical device.

**SSCP.** System Services Control Point.

**statistics report message unit.** A message unit sent by the target (receiver of the file), at the end of the file transfer.

**status.** The state of a request in the request queue (MVS, VSE, or VM) or request database (OS/400). In NetView FTP for MVS, VSE, or VM, the possible statuses are waiting, active, and finished. In NetView FTP/400, the possible statuses are held, waiting, active, finished, and failed.

**status data area.** The area of ECSA into which NetView FTP places details of the status of a server or servers.

**Storage Management Subsystem.** An MVS subsystem that helps automate and centralize the management of DASD storage. SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

**submit (a request).** To give (a request) to NetView FTP so that NetView FTP can add it to the request queue.

**superuser.** In the AIX operating system, the user who has unrestricted authority to access and modify any part of the operating system, usually the user who manages the system.

**supervisor call instruction.** An instruction that interrupts a running program and passes control to the supervisor so that the supervisor can perform the service indicated by the instruction.

**SVA.** Shared Virtual Area.

**SVC.** Supervisor call instruction.

**symbolic constant.** A data item that has an unchanging, predefined value.

**system authorization facility.** At an MVS or VM location, a generic interface to security products that is provided by the operating system. In this way, an installation has the possibility to run a security program, such as IBM RACF.
System management facilities. An optional control program for MVS that provides the means for gathering and recording information that can be used to evaluate system usage.

system services control point. In SNA, a focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Several SSSPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units within its domain.

Systems Network Architecture. The description of the logical structure, formats, protocols, and operating sequences for transmitting information units through, and for controlling the configuration and operation of, networks. The layered structure of SNA allows the origin and ultimate destination of information to be independent of and unaffected by the SNA network services and facilities used to transfer that information.

table display panel. A panel that contains a scrollable list.

Target Information Message Unit. A message unit sent by the receiving node after it receives an SDMU. A TIMU updates the original request, to inform the sender where to start or restart sending, and, to convey statistical information.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol. A set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks.

TIMU. Target Information Message Unit.

token-ring network. A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

Transfer Request Message Unit. A message unit used to send or receive a file, or to restart a previously interrupted file transfer.

Transmission Control Protocol. A communications protocol used in Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

TRMU. Transfer Request Message Unit.

unattended mode. An operating mode of NetView FTP for Workstations that assumes that a user is not available at the workstation, for example, to load devices such as diskettes or tapes. Contrast with attended mode.

unused slot. A slot that does not contain a request control element (RCE).

user exit. A point in an IBM-supplied program at which a user-exit routine is given control.

user interface. Hardware, software, or both that allows a user to interact with and perform operations on a system or program.

user query command. A type of command that causes NetView FTP to retrieve information about the requests in the request queue; the information retrieved is intended for users of NetView FTP. Contrast with administrator query command.

user query record. A record containing the information NetView FTP retrieves when a user issues a user query command. Contrast with administrator query record.

user-exit routine. An exit routine written by a user.

user-written file handler. An exit routine, written by a user, that gains access to a file and passes its records to NetView FTP, or that retrieves the records from NetView FTP and writes them to a file.

virtual storage access method. An access method for indexed or sequential processing of fixed- and variable-length records on direct-access storage devices. The records in a VSAM data set can be organized (1) in logical sequence by means of a key field (key sequence), in the physical sequence in which they are written in the data set (entry sequence), or (2) by means of a relative-record number.


VM node. A node with VM as its operating system.

VM system. A system with VM as its operating system.

VSAM. Virtual storage access method.

VSE node. A node with VSE as its operating system.

VSE system. A system with VSE as its operating system.
VTAM. Virtual Telecommunications Access Method. See *Advanced Communications Function for the Virtual Telecommunications Access Method*.

**waiting request.** A request that is waiting to be processed.

**WAN.** Wide area network.

**Wide area network.** A network that provides communication services to a geographic area larger than that served by a local area network and that may use or provide public communication facilities.

**wildcard character.** Either a question mark (?) or an asterisk (*) used as a variable in a file name or file name extension when referring to a particular file or group of files.

**WTO.** Write-to-operator.

**XID.** Exchange identification.

**XnMU.** Exit(n) message unit.
The NetView FTP Library

This manual is part of a library of publications that describe NetView FTP and explain how to use it. The publications in this library are:

**NetView File Transfer Program for VSE:**
- NetView FTP Licensed Program Specifications, GH12-5485
- NetView FTP VSE Installation, Operation, and Administration, SH12-5674
- NetView FTP Customization, SH12-5482
- NetView FTP Messages and Codes, SH12-5483
- NetView FTP Parameter Reference, SH12-6052

**NetView File Transfer Program for VM:**
- NetView FTP General Information, GH12-5480
- NetView FTP Licensed Program Specifications, GH12-5485
- NetView FTP VM Installation, Operation, and Administration, SH12-5676
- NetView FTP Customization, SH12-5482
- NetView FTP Messages and Codes, SH12-5483

**NetView File Transfer Program for OS/400:**
- NetView FTP General Information, GH12-54808
- NetView FTP V1 for OS/400 Licensed Program Specifications, GH12-5777
- NetView FTP V3 for OS/400 Licensed Program Specifications, GH12-6176
- NetView FTP V1 for OS/400 Installation and User's Guide, SH12-5776

**NetView File Transfer Program Version 2 for MVS:**
- NetView FTP Licensed Program Specifications, GH12-5485
- NetView FTP V2 MVS Installation, Operation, and Administration, SH12-5657
- NetView FTP Customization, SH12-5482
- NetView FTP Messages and Codes, SH12-5483
- NetView FTP Parameter Reference, SH12-6052

**NetView File Transfer Program Server for AIX, NetView File Transfer Program Client for AIX:**
- NetView FTP Parameter Reference, SH12-6052

**NetView File Transfer Program Server/2, NetView File Transfer Program Client/2**
- NetView File Transfer Program Client for DOS and Windows:
  - NetView FTP Parameter Reference, SH12-6052

The unlicensed manuals with prefix SH are also available as softcopy on the following collection kits:

- IBM Networking Softcopy Collection Kit, SK2T-6012
- IBM Online Library Omnibus Edition: MVS Collection, SK2T-0710
- IBM Online Library Omnibus Edition: VM Collection, SK2T-2067
- IBM Online Library Omnibus Edition: VSE Collection, SK2T-0060
- IBM Online Library Omnibus Edition: AIX Collection, SK2T-2066

**Related Publications**

- OSI/File Services for MVS User's Guide, SH19-6638
- OSI/File Services General Information Manual, GH19-6636
- TSO/E Version 2—REXX/MVS Reference, SC28-1883

---

8 NetView FTP/400 V1 only.
Index

Special Characters
/ (slash) 56
*DVGAPL* 69, 83
*DVGAPX* 69, 85

Numerics
16MB boundary 68

A
A-type address 77
access control 23
access method 14
accounting 23
active status 4
adaptive compression 19
address
A-type 77
of the QRA 84
administering the queue 47, 87
administrator query record 72
alias 37
allocation
dynamic 20
file 20
job 20
AMS commands 17
AMS parameters 20
APL
general 67
initializing and assigning values to the fields of 68
mapping for 75
rules for assigning values to the fields of 69
APL control block identifier 83
APLACCR 69
APLFBALN 84
APLFBAP 84
APLID 69, 83
APLMACT 84
APLMAPTR 84
APLMTOT 84
APLQACNT 84
APLQTCNT 84
APLRC 83
APLRQNUM 84
APLRSN 83
APLSMBR# 85
APLSMBRN 85
APLXMBR# 85
APLXMBRN 85
APPC conversation
security 23
application program
creating requests 69
modifying a request 73
NetView FTP V2.2.1 MVS 7
parameter list 67
parameter list extension 67
querying a request 72
restarting a request 73
transferring entire PDSs 70
transferring groups of members 70
transferring members 70
writing 67
APX
control block identifier 85
general 67
initializing and assigning values to the fields of 68
mapping for 75
rules for assigning values to the fields of 69
APXACCR 69
APXID 69, 85
area, message 74
assigning values to the fields of the APL and APX 68, 69
automatic
logon retry 13
transfer restart 25
automatically retrieving values for access security parameters 24

B
batch job
converting a NetView FTP V2.1 MVS 65
creating requests 61
deleting a request 64, 74
general 55
interface routine, log file 65
interface routine, messages 65
modifying a request 63
NetView FTP V2.1 MVS 7
querying a request 62
restarting a request 63
blank suppression 60
boundary, 16MB 68
BPAM 14
BSAM 14
C

calling the external function DVGCXRIA 78
CATLG, end-of-processing option 36
changing
  current request 34
  saved request 37, 39
character data conversion 19
checking requests 21
checkpoint
  taking a 24
class
  request 10
CLIST 23
cluster
  allocation 20
code, reason and return 46, 50
coding job control statements 56
coding NetView FTP V2.2.1 MVS control statements 58
command line 27
communication fields 83
compaction, SNA 19
completion message, file-transfer 5, 21
compression
  general 19
console, operator's 65
constant, symbolic 80
contents of NetView FTP V2.2.1 MVS panels 27
contents of registers (DVGCALL) 76
control block identifier
  APL 83
  APX 85
control statement 55, 58
controlling requests 39, 47, 87
conversation 4
conversion
  character data 19
converting NetView FTP V1 MVS BP, NetView FTP V1 MVS AFF, and NetView FTP V2.1 MVS requests 53
converting NetView FTP V2.1 MVS batch jobs 65
creating a request 31
creation, file 20
current request
  panel 33
  summary 34
  task selection 33

date
  for submitted request 46
  request submitted 49, 50
  transfer finished 50
  transfer started 50
declaring the Assembly-language routine in the C language 77
defaults 51
definition module, message 57
delaying processing 8
deleting requests 3, 39, 47, 50
description of request to be saved 34
DFSUDMP0 17
DFSURDB0 17
direction, file-transfer 4
directory
  information 35, 37
discard, saved request 39
discarding a saved request 37
displaying
  statistics about the queue 47
DL/I databases 17
dump, batch job interface routine 57
DVGAPI 68, 75
DVGAPL macro 80
DVGAPX macro 80
DVGCALL macro 76
DVGCCMDx 57
DVGCXAPR 143
DVGCXCAI 77
DVGCXRIA 78
DVGLLOG 58
DVGMEM00E 30
DVGMEM01E 32
DVGMEM0AE 31
DVGMEM0KE 33
DVGMEM0LE 53
DVGMEM0NE 52
DVGMEM0OE 51
DVGMEM0QE 52
DVGMEM0RE 38
DVGMEM1PE 35
DVGMEMAPE 47
DVGMEMAEQE 48
DVGMEMAXE 49
DVGMEMAEQE 40
DVGMEMEQE 45
DVGMEMETAE 53
DVGMEMETBE 54
DVGQAR macro 80
DVGQSR macro 80
dynamic allocation 20
dynamic file creation 20

data encryption 24
data security 23
data set
  ISPF table 34
databases, DL/I 17
E
end-of-processing option 36
End-of-Request parameter 58
entire PDS, transferring 35
EOR 58
equates 80
error recovery 24
example
  of a batch job 129
  of an application program 139
  of application program output 80
  of using NetView FTP V2.2.1 MVS panels 95
excluding members 36, 37
EXEC job control statement 56
EXPORT command 17

F
fields
  communication 83
  initializing and assigning values to 68
file
  allocation 20
  compression 19
  creation 20
  request definition 58
  type 14
file handler, user-written 17
file organization
  general considerations 16
  supported by NetView FTP for Workstations 16
  supported by NetView FTP V2 MVS 14
  supported by NetView FTP VM 14
  supported by NetView FTP VSE 14
  supported by NetView FTP/400 15
file organizations
  specifying 21
file status option 36
file-transfer
  completion message 5, 21
  direction 4
  reason code 46, 50, 61
  reason codes 22
  report 5, 21
  request 2
  return code 46, 50, 61
  return codes 22
finished status 5
forcing requests 39, 47, 50

G
getting help from NetView FTP V2.2.1 MVS 30
glossary 145

group, server 11

H
HDAM 17
header lines 80
help 30
HISAM 17
holding requests 3, 50

I
ID
  originator 8, 9
  identifying responding system 11
IEHMOVE 17
IMS/VS 17
initializing the APL and APX 68
input
  fields 27
INSERT PDS option 36
invoking NetView FTP V2.2.1 MVS 29
ISPF
  panels 7
  table data set 7, 34

J
job
  allocation 20
  control statement 55
JOBLIB 57

K
KEEP, end-of-processing option 36
key-sequenced data set option 18
keyword 58
KSDS 18

L
languages, message definition 57
layout, panel 27
length
  of the APL 83
  of the QRA 84
library
  link list 57
  link pack area 57
lines, header 80
link
  list library 57
  pack area library 57
  password 23
LINKLIB 57
list of members, displaying 35
listing saved requests 37
local
   LU name 49, 50
locating a request in a list panel 47
log files
   batch job interface 65
   queue handler 23
   server 23
logging and reporting 21
logon retry, automatic 13
LPALIB 57
LU name
   general 11
   local and remote 49
   remote 38, 50

M
macro
   DVGAPL 80
   DVGAPX 80
   DVGCALL 76
   DVGQAR 80
   DVGQSR 80
   general 75
   mapping 17
   NetView FTP V2.2.1 MVS 8
   main transfer parameters 31
   manual transfer restart 25
   mapping 8, 75
   master password 9, 39, 47
   measuring performance 23
   members
      excluding 36, 37
      selecting 36
   message
      area 67, 74, 84
      batch job interface routine 65
      CLIST 23
      definition module 57
      file-transfer completion 5, 21
      line 27
   modification of submitted jobs 141
   modifying requests
      general 3, 50
      holding requests 39, 47
      releasing requests 39, 47
   module, message definition 57
   MUSTNOTEXIST, file status option 36

N
name
   of request to be saved 34
   NetView FTP V2.1 MVS batch jobs, converting 65
   NetView FTP V2.2.1 MVS
      application program 7
      batch job 7
      control statement 55
      master password 39, 47
      panels 7, 27
   new name 36
   notification 21
   number
      members that can be selected or excluded 85
      messages issued 84
      messages moved to the message area 84
      of transferred records 46, 50

O
Open Systems Interconnection 1
operator’s console 65
option for KSDSs 18
option, PDS 36
options for partitioned data sets 18
originator ID 8, 9
originator, request
   active requests 50
   finished requests 50
   submitted requests 46
   to identify 8
   to protect requests 9
   waiting requests 49
OSI 1
OSI network 3
OSI servers 3
OSI transfer parameters 32

P
panel
   Current Request Task Selection 33
   layout 27
   Main Task Selection 30
   main transfer parameters 31
   NetView FTP V2.2.1 MVS 7, 27
   OSI transfer parameters 32
   Request Control 40
   Saved Requests 38
   Submitted Requests 45
   parameter 31
   End-of-Request 58
   transfer 31, 32
   Wait 61
   PARM 57
   partitioned data set option 18
   password
      link 23
      master 9, 39, 47
password (continued)
request 9
VSAM 23

PDS
  displaying a list of members 35
  excluding members 37
  option 18
  selecting members 36
  transferring an entire PDS 35
  transferring members 35

performance 23

POWER/PNET 22

priority, request 10
processing requests 3

Q

QRA
  address and length of the 84
  APLQACNT 84
  APLQTCTNT 84
  APLRONUM 84
  general 67
  mapping for 75

QSAM 14, 17
query record 72
query response area
  address and length of the 84
  APLQACNT 84
  APLQTCTNT 84
  APLRONUM 84
  general 67

querying requests 3, 39, 47
queue
  handler log files 23
  request 2
queue handler command return and reason code 65, 68

R

RDF
  request definition file 58, 62
  transferring entire PDSs 62
  transferring groups of members 62
  transferring members 62
  writing an RDF 62
reason code
  APLRC 83
  APLRSN 83
  batch job 61
  finished requests 50
  general 22
  queue handler 65, 68
  submitted requests 46

recipient of file-transfer report 21
record
  number transferred 46, 50
  SMF 23
recovery, error 24
refreshing contents of a list panel 47, 51
REGION 57
register number 77
registers, contents of (DVGCALL) 76
releasing requests 3, 50
remote
  LU name 38, 49, 50
  name 49
  system 49
REPLACE PDS option 36
report recipient 21
report, file-transfer 5, 21
reporting 5
reporting and logging 21
REPRO command 17
request
  checking 21
  class 10, 46, 49, 50
  controlling 39, 47
  creating 2, 31
  current 33
  defaults 51
  definition file 58
  deleting 39, 47
  file-transfer 2
  forcing 39, 47
  holding 39, 47
  modifying 39, 47
  name 46
  number 40, 49, 50
  originator 40, 49
  password 9, 40
  priority 10, 49
  querying 39, 47
  queue 2
  releasing 39, 47
  restarting 39
  saved 37
  status 46, 49
  submitted 39, 47
request definition file 58
requesting
  server 4
responding
  server 4
  system, identifying 11
restart
  automatic transfer 25
  manual transfer 25
restart information
  submitted requests 46
writing application programs 67
writing batch jobs 55
Your comments, please ...

NetView
File Transfer Program
Version 2 for MVS
User’s Guide
Release 2.1
Publication No. SH12-5656-04

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you mail this form to us, be sure to print your name and address below if you would like a reply.

You can also send us your comments using:

• A FAX machine. The number is: +49–7031–166609.
• Internet. The address is: gadlid@sdfvm1.vnet.ibm.com.
• IBMLink. The address is: SDFVM1(GADLID).
• IBM Mail Exchange. The address is: DEIBM3P3 at IBMMMAIL.

Please include the title and publication number (as shown above) in your reply.

Name

Address

Company or Organization

Phone No.
Program Number: 5685-108

Printed in Denmark by IBM Danmark A/S