Installation, Operation, and Administration

Release 2.1
Installation, Operation, and Administration

Release 2.1
Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page v.


This is a major revision of, and obsoletes, SH12-5657-03.

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.
# Contents

**Notices** ................................................................. v  
**Programming Interfaces** ........................................... v  
**Trademarks and Service Marks** ................................... v  

**About This Book** ....................................................... vii  
**Abbreviations and Special Terms** ................................ vii  

## Part 1. Introduction .................................................. 1  

**Chapter 1. What NetView File Transfer Program for MVS Is and Does** .......................... 3  
**File-Transfer Requests** ................................................. 4  
**How NetView FTP MVS Is Structured** .................................. 5  
**How NetView FTP V2.2.1 MVS Works** .................................. 6  
  - Creating and Adding a Request to the Queue .......................... 6  
  - Scheduling a Request ................................................ 7  
  - Transferring a File .................................................. 8  
**Stopping NetView FTP V2.2.1 MVS Running** .................................. 12  

## Part 2. Installation .................................................... 13  

**Chapter 2. Installing, Preparing, and Verifying the Installation of NetView FTP V2.2.1 MVS** 15  
**Installing NetView FTP V2.2.1 MVS** ................................... 16  
**Preparing the NetView FTP V2.2.1 MVS Program** .......................... 19  
  - Defining the Load Libraries ......................................... 19  
  - Installing the SVC Routine ........................................... 21  
  - Performing IPL after Loading ....................................... 22  
  - Authorizing the Started Tasks for RACF ............................ 22  
  - Setting Up ACF/VTAM Resources ..................................... 22  
  - Submitting Jobs to Create the Request Queue and the Checkpoint Data Set 22  
  - Submitting Jobs to Create the Log File and Report File ............... 23  
  - Copying the NetView FTP V2.2.1 MVS Procedures, Parameter Members, and the Sample Server Group Table ................................. 23  
  - Gaining Access to the Interactive Interface Routine .................. 24  
**Preparing the OSI Part of NetView FTP V2.2.1 MVS** .................................. 26  
**Preparing the Japanese Language Feature of NetView FTP V2.2.1 MVS** ......................... 26  
  - Preparing the NetView FTP V2.2.1 MVS Jobs for the Japanese Language Feature ............................................................. 26  
  - Gaining Access to the Interactive Interface Routine of the Japanese Language Feature .......................................................... 27  
**Verifying the Installation of NetView FTP V2.2.1 MVS** .................................. 28  
**Preparing for the Installation Verification Test** .................................. 28  
**Starting the Queue Handler** ............................................. 28  
**Starting Servers** .......................................................... 29  
**Overview of the Installation Verification Tests** .................................. 29  
**Installation Verification Test 1** ......................................... 30  
**Installation Verification Test 2** ......................................... 33  
**Installation Verification Test 3** ......................................... 37  
**Verifying the Access to the Interactive Interface Routine** .................................. 39  

© Copyright IBM Corp. 1988, 1994
<table>
<thead>
<tr>
<th>Chapter 7. Setting Up a Server Group Table</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a Server Group Table</td>
<td>85</td>
</tr>
<tr>
<td>Considerations for Defining Server Groups</td>
<td>87</td>
</tr>
<tr>
<td>Updating a Server Group Table</td>
<td>88</td>
</tr>
</tbody>
</table>

| Chapter 8. SMF Recording                  | 89 |

<table>
<thead>
<tr>
<th>Chapter 9. Administering OSI File Transfers</th>
<th>91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleting Entries from the OSI/File Services Results File</td>
<td>93</td>
</tr>
<tr>
<td>Making Entries in the Local Resource Directory</td>
<td>94</td>
</tr>
<tr>
<td>Recovering OSI File Transfers</td>
<td>94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 10. Writing Startup Procedures for NetView FTP V2.2.1 MVS</th>
<th>95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure to Start the Queue Handler</td>
<td>95</td>
</tr>
<tr>
<td>Specifying Values for the Queue Handler Initialization Parameters</td>
<td>99</td>
</tr>
<tr>
<td>Procedure to Start a Server</td>
<td>105</td>
</tr>
<tr>
<td>Batch Jobs to Start a Server</td>
<td>110</td>
</tr>
<tr>
<td>Specifying Values for the Server Initialization Parameters</td>
<td>113</td>
</tr>
<tr>
<td>Initialization Parameters for an SNA Server</td>
<td>114</td>
</tr>
<tr>
<td>Example of Initialization Parameters for a Server</td>
<td>124</td>
</tr>
<tr>
<td>Initialization Parameters for an OSI Server</td>
<td>124</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 11. Changing the Text of NetView FTP V2.2.1 MVS Messages and the Language They Are Issued In</th>
<th>129</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules to Follow when Changing the Message Texts</td>
<td>130</td>
</tr>
<tr>
<td>Special Considerations for Languages That Require Double-Byte Characters</td>
<td>132</td>
</tr>
</tbody>
</table>

| Part 5. Problem Determination                       | 133 |

<table>
<thead>
<tr>
<th>Chapter 12. Taking Traces</th>
<th>135</th>
</tr>
</thead>
<tbody>
<tr>
<td>When to Take Traces</td>
<td>135</td>
</tr>
<tr>
<td>Types of Traces</td>
<td>136</td>
</tr>
<tr>
<td>Entry and Exit Trace</td>
<td>136</td>
</tr>
<tr>
<td>Specific Module Traces</td>
<td>136</td>
</tr>
<tr>
<td>Activating the Trace Facility</td>
<td>137</td>
</tr>
<tr>
<td>Starting and Stopping a Trace</td>
<td>137</td>
</tr>
<tr>
<td>Module Groups for Traces</td>
<td>138</td>
</tr>
<tr>
<td>Tracing the Queue Handler</td>
<td>139</td>
</tr>
<tr>
<td>Tracing an SNA Server</td>
<td>140</td>
</tr>
<tr>
<td>Tracing an OSI Server</td>
<td>143</td>
</tr>
<tr>
<td>SNA Server and Queue Handler Trace Output</td>
<td>145</td>
</tr>
<tr>
<td>The NetView FTP V2.2.1 MVS Symptom Record as Trace Output</td>
<td>145</td>
</tr>
<tr>
<td>Register Dump as Tracing Output</td>
<td>147</td>
</tr>
<tr>
<td>Amount of Output Produced during Tracing</td>
<td>149</td>
</tr>
<tr>
<td>OSI Server Trace Output</td>
<td>152</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 13. Handling NetView FTP V2.2.1 MVS Abnormal Terminations</th>
<th>153</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovery Routines in NetView FTP V2.2.1 MVS</td>
<td>153</td>
</tr>
<tr>
<td>Error Situations in the Interactive Interface</td>
<td>154</td>
</tr>
<tr>
<td>Abnormal Termination in the User Interface Routines</td>
<td>155</td>
</tr>
<tr>
<td>Abnormal Termination of the Queue Handler</td>
<td>155</td>
</tr>
</tbody>
</table>
Appendix D. Storage Requirements .................................................. 199
Specifying Checkpoint Data-Set Parameters .................................. 199
  Number of Records ...................................................................... 199
Specifying Request Queue Parameters .......................................... 199
  Record Size .............................................................................. 199
Queue Handler ............................................................................. 201
Server .......................................................................................... 202
Batch Job Interface Routine .......................................................... 202
Application Program Interface Routine .......................................... 202
Interactive Interface Routine ......................................................... 202
Common Storage Requirements ...................................................... 203
  SVC Routine ............................................................................. 203
Static Common Service Area Requirements ..................................... 203
Dynamic Common Service Area Requirements ............................... 203

Appendix E. Factors That Affect the Performance of NetView FTP V2.2.1 MVS .............................................................. 209
FSBSLOTS Queue Handler Initialization Parameter Value ............... 209
CSAELIM Queue Handler Initialization Parameter Value .............. 209
TIMECR and BYTECR Server Initialization Parameter Values ........ 210
The Maximum Number of File Transfers That NetView FTP V2.2.1 MVS Can Process Concurrently ............................... 210
The Compression Method Used ...................................................... 211
Server Classes ............................................................................. 211
The OBTINTVL Server Initialization Parameter Value .................... 212
Request Unit ................................................................................. 212
  Request-Unit Size .................................................................... 212
Pacing .......................................................................................... 213
  How Pacing Affects Throughput ............................................... 213
  Pacing Specifications ................................................................ 213
  Rules for Defining Pacing Values in a Logon Mode Table ............ 214
Machine Cycles versus Transfer Time ............................................ 214
Simultaneous File Transfers per Server versus Several Servers ........ 214
Additional Considerations ............................................................. 215

Appendix F. Fixed Parameters and Constants Used in NetView FTP V2.2.1 MVS ................................................................. 217

Appendix G. NetView FTP V2.2.1 MVS Log Files .................................. 219
Batch Job Interface Log File ............................................................ 219
Queue Handler Log File .................................................................. 220
Server Log File .............................................................................. 221

Appendix H. Layout of the SMF Records ........................................... 223
Interpreting SMF Records .............................................................. 224

Appendix I. Using NetView in Conjunction with NetView FTP V2.2.1 MVS ................................................................. 249
Introduction .................................................................................... 249
Automated Operation of NetView FTP V2.2.1 MVS .......................... 249
The Startup Parameter ..................................................................... 250
Description of Multi-line WTOs ....................................................... 251
Events Issued by NetView FTP V2.2.1 MVS ....................................... 251
  Events Issued by the Queue Handler and the SNA Server ............ 251
  Events Issued by the Queue Handler Only .................................... 253
<table>
<thead>
<tr>
<th>Events Issued by the SNA Server Only</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Log Files with NetView Events</td>
<td>260</td>
</tr>
<tr>
<td>Appendix J. Macro List</td>
<td>261</td>
</tr>
<tr>
<td>Glossary</td>
<td>265</td>
</tr>
<tr>
<td>Bibliography</td>
<td>279</td>
</tr>
<tr>
<td>The NetView FTP Library</td>
<td>279</td>
</tr>
<tr>
<td>Related Publications</td>
<td>279</td>
</tr>
<tr>
<td>Index</td>
<td>281</td>
</tr>
</tbody>
</table>
Notices

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

This book is intended to help the customer to install, administer, and operate NetView FTP V2.2.1 MVS.

This book also documents General-Use Programming Interface and Associated Guidance Information and Diagnosis, Modification, and Tuning 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.

This book also documents Diagnosis, Modification, or Tuning Information, which is provided to help the customer to do diagnosis of NetView FTP V2.2.1 MVS.

Warning: Do not use this Diagnosis, Modification, or Tuning Information as a programming interface.

Diagnosis, Modification, or Tuning 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 (*), used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF/VTAM</td>
<td>AIX</td>
</tr>
<tr>
<td>C/370</td>
<td>IBM</td>
</tr>
<tr>
<td>MVS/ESA</td>
<td>MVS/XA</td>
</tr>
<tr>
<td>OS/2</td>
<td>OS/400</td>
</tr>
<tr>
<td>RMF</td>
<td>VM/ESA</td>
</tr>
<tr>
<td>VSE/ESA</td>
<td>VM/XA</td>
</tr>
<tr>
<td>VTAM</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. The book is intended to assist system programmers to install, administer, and operate NetView FTP V2.2.1 MVS. It contains the installation procedure, administration information, and information required to operate NetView FTP V2.2.1 MVS and to trace and report problems.

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 MVS** NetView File Transfer Program for MVS (regardless of version 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 VM** NetView File Transfer Program for VM (regardless of version number)

**NetView FTP VSE** NetView File Transfer Program for VSE (regardless of version number)

**NetView FTP/400** NetView File Transfer Program for OS/400® (regardless of version number)

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

**NetView FTP/2** NetView File Transfer Program Server/2
- NetView File Transfer Program Client/2

**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 AIX
NetView File Transfer Program Server for AIX and
NetView File Transfer Program Client for AIX

NetView FTP Server AIX
NetView File Transfer Program Server for AIX

NetView FTP Client AIX
NetView File Transfer Program Client for AIX

This book sometimes refers to a location with an MVS operating system as an “MVS location,” a location with a VSE operating system as a “VSE location,” and a location with a VM operating system as a “VM location.” If this book refers to NetView FTP server workstations or to NetView FTP client workstations, the description is valid for both, OS/2 and AIX.

In this book:

- The abbreviation MVS refers to the MVS/370, MVS/XA*, and MVS/ESA* operating systems.
- The abbreviation VM refers to VM/XA*, VM/HPO, and VM/ESA* operating systems.
- The abbreviation VSE refers to both VSE/XA and VSE/ESA* operating systems.
- The abbreviation AIX refers to the AIX/6000* operating system.
- The terms data set, file, and cluster are all used as synonyms.
- The terms file handler, user-exit routine, and user-written file handler are used as synonyms.
- The term hex is used as an abbreviation for hexadecimal.

If you come across an unfamiliar word, refer to the “Glossary” on page 265. A bibliography and an index are also provided at the back of this book.
Part 1. Introduction

This part provides general information about NetView FTP V2.2.1 MVS.
Chapter 1. What NetView File Transfer Program for MVS Is and Does

NetView FTP V2.2.1 MVS for MVS is an ACF/VTAM 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 two computer systems at which NetView FTP is installed can transfer files to each other, regardless of their operating systems. NetView FTP transfers files directly from a storage device at one location to a storage device at another location, without intermediate spooling.

NetView FTP V2.2.1 MVS can transfer files to or from a node in:

- An SNA network using SNA LU 6.2 APPC.
- 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 the OSI/File Services General Information Manual.
- A local area network (LAN) via a NetView FTP Server for OS/2 Server gateway using SNA and NetBIOS protocols.

Figure 1. Example of an SNA File Transfer

Because the transfer programs at any two locations work together to transfer files, they can communicate only on the level of the less sophisticated of the two transfer programs.
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. After you submit a request, NetView FTP V2.2.1 MVS puts it in a VSAM file, called the request queue, where it waits to be processed. Request processing takes place independently of request queuing.

To process requests, NetView FTP V2.2.1 MVS uses components called servers. For transfers between nodes in an SNA network, it uses NetView FTP SNA servers (abbreviated to SNA servers in this book). For transfers within an OSI network, it uses NetView FTP OSI servers (abbreviated to OSI servers in this book).

Each SNA server has its own logical unit (LU) name and runs independently of any other server. An SNA server obtains a request from the queue and initiates a conversation with an SNA server at the remote location specified in the request. The two SNA servers work together to transfer the file specified in the request, and then terminate the conversation.

An OSI server is registered as an end user of OSI/File Services. An OSI server obtains a request from the queue, and passes it to OSI/File Services, which conducts the file transfer. An OSI server does not transfer files itself.

How transfers are handled is described in more detail in “How NetView FTP V2.2.1 MVS Works” on page 6.
How NetView FTP MVS Is Structured

The way that the components of NetView FTP are arranged in an MVS system is shown in Figure 2.

You communicate with NetView FTP MVS by using one of the following:

- An interactive interface (II) routine
- A batch job interface (BJI) routine
- An application program interface (API) routine.

You use the interface routines to add requests to the queue and to query, modify, and delete requests that are in the queue. The interface routines reside in the address spaces of the persons or programs that use them.

Figure 2. NetView FTP MVS Structure
The main NetView FTP MVS component, the *queue handler*, resides in its own address space; each server also resides in its own address space. The interface routines and servers communicate with the queue handler by using MVS common storage.

A NetView FTP V2.2.1 MVS installation can have up to 99 servers. This can be a combination of SNA and OSI servers. Each SNA server can handle up to 32 transfer requests simultaneously. Each SNA server contains a component that takes checkpoints during a file transfer.

An OSI server does not transfer files, it only passes on the requests to OSI/File Services, which handles the transfer. No checkpoints are taken by NetView FTP V2.2.1 MVS during the transfer.

The queue handler has an interface to the operator console, from which the operator can view information about, and can control, NetView FTP V2.2.1 MVS. The queue handler also has a *server group table*, which contains information about the SNA servers in your network.

NetView FTP V2.2.1 MVS allows you to specify as much common storage as you need, and allocates all its common storage above the 16MB boundary (1MB is 1 048 576 bytes).

---

**How NetView FTP V2.2.1 MVS Works**

The following gives a step-by-step description of what happens after NetView FTP V2.2.1 MVS has been started and the queue handler and the servers are ready to work.

**Creating and Adding a Request to the Queue**

1. To create and add a request to the queue, a user can do one of the following:
   - Run a batch job
   - Start an application program
   - Use the interactive interface.

   How to write such batch jobs and application programs is described in the *NetView FTP V2 MVS User’s Guide*. The batch job or application program specifies values for parameters that tell NetView FTP V2.2.1 MVS things such as:
   - Whether to transfer a file to or from another location
   - Which location to transfer the file to or from
   - Which file is to be transferred from the sending location
   - What name the file is to have at the receiving location
   - When to process the request
   - Whom to notify once the file has been transferred
   - What class the server that processes the request must belong to.

2. The interface routine that runs in the same address space as the batch job, a TSO user, or application program, checks the parameters and their values for completeness and consistency.

3. The interface routine creates a request.
4. The interface routine passes control to the pre-queuing user-exit routine. You can modify this routine so that it checks the parameters of the request, changes some of their values, or performs other processing. This routine is described in the NetView FTP Customization guide.

5. The pre-queuing user-exit routine passes control back to the interface routine.

6. The interface routine tells the queue handler component to assign a request number to the request, give it the status waiting, and add it to the request queue.

7. After a user submits one request, that user can immediately submit another—the user does not have to wait until NetView FTP V2.2.1 MVS has processed the first request.

**Scheduling a Request**

1. *Scheduling* a file-transfer request means deciding which request is to be selected from the queue and passed to a server for processing. Scheduling begins when a server asks the queue handler to obtain a request for it to process.

2. The queue handler notes which classes are assigned to the server.

3. The queue handler searches through all requests in the queue for those that require a remote check. For SNA transfers, the parameters specified for the file-transfer request are then checked at the remote location as described in point 7.

4. The queue handler searches through all dispatchable requests in the queue for those that specify classes assigned to the server. Dispatchable requests are waiting requests that are not held and whose not-before time and date has expired.

5. Of those, it selects the request with the highest priority. If there is more than one request with the highest priority, it selects the one with the highest class. If there is more than one request with the highest class, it selects the one that was submitted earliest.

6. After the queue handler selects a request, it checks if that request’s not-after time has expired. If so, it changes the request’s status to finished and repeats step 4 until it finds a suitable request. If there are no suitable requests in the queue, the queue handler tells this to the server. The server waits for the amount of time specified in the OBTINTVL server initialization parameter, then asks the queue handler again to obtain a request for it. This parameter is described in “Specifying Values for the Server Initialization Parameters” on page 113. The server continues to ask for a request until the queue handler locates a dispatchable request for the server or until the server is stopped.
7. After the queue handler finds a request that it can pass to the server, it checks whether the request specifies the LU name of a single server or the name of a server group. If the request specifies the name of a server group, the queue handler retrieves the following information from the server group table and inserts it into the request:

- The LU names of the first eight servers in the server group that have the running mode specified in the request.
- The number of servers in the group whose LU names were not inserted.

**Note:** Point 7 does not apply to OSI file transfers.

8. The queue handler passes the request to the server. The request’s status is changed to **active**.

### Transferring a File

The following describes how a file is transferred using an SNA server and an OSI server.

#### SNA Server

A file is transferred by an SNA server as follows:

1. After the local server obtains a request from the queue handler, it attempts to allocate a conversation with the first server listed in the server group specified in the request.

   If an LU name is specified, the server tries to allocate a conversation only with the server represented by this LU name. (If the LU name of a single server was specified, then this server is the only one listed in the request.)

   Note that for NetView FTP/2 and NetView FTP AIX the server group must not contain more than one LU name, as a group of NetView FTP Client programs is represented by the LU name of the NetView FTP Server workstation to which they are attached.

2. If the server at the remote location (the **remote server**) accepts the conversation parameters, the conversation is established and the server proceeds with step 3.

   If the remote server does not accept the conversation parameters, the local server tries to establish a conversation with the next server listed in the server group specified in the request. It continues trying until it either succeeds in establishing a conversation or has tried to establish a conversation with all the servers listed in the server group specified in the request.

   If there are more LUs in the server group than the ones already tried, the server tells the queue handler to retrieve the names of up to eight LUs. The server repeats this process until it either succeeds in establishing a conversation or has tried all the LU names associated with the server group.

   After it has tried all the servers in the group without success, the local server notifies the queue handler and tells it to place the request back into the queue, leave it in waiting status, and set its not-before time to the current time plus the time specified in the server’s REQDELAY initialization parameter. This parameter is described in “Specifying Values for the Server Initialization Parameters” on page 113.
Later, after the not-before time has passed, this request is again eligible for scheduling (dispatchable). The server that then obtains the request might be different from the one that obtained it last.

3. The local server sends information about the file-transfer request to the remote server. The local and remote servers then determine which is the sending server and which is the receiving server. This depends on the direction of the file transfer (to or from the remote location). The transfer direction is specified in the request and is independent of which server initiated the conversation.

4. The sending server passes control to the pre-transfer user-exit routine at its location. You can modify this routine so that it checks the parameters of the request, changes some of their values, or performs any other pre-transfer processing. This routine is described in the NetView FTP Customization guide.

5. If appropriate, the server uses values specified in the request to establish a security environment, which it uses to open the sending file.

6. Unless job allocation is to be used for the sending file, the sending server dynamically allocates it. If job allocation is to be used, the file is allocated in the JCL used to start the server. Dynamic allocation and job allocation are described in the NetView FTP V2 MVS User’s Guide.

7. The sending server opens the sending file, extracts information about the sending file, and sends that information to the receiving server.

8. The receiving server passes control to the pre-transfer user-exit routine at its location where the part of the request that specifies parameters for the remote location can be modified.

9. If appropriate, the server uses values specified in the request to establish a security environment, which it uses to open the receiving file.

10. Unless job allocation is to be used for the receiving file, the receiving server dynamically allocates it. If the receiving file does not already exist, the receiving server creates it before allocating it. If job allocation is to be used, the file is allocated in the JCL used to start the server.

11. The receiving server opens the receiving file, extracts information about the receiving file, and sends that information to the sending server.

12. The local server asks the queue handler to update the file-transfer request in the request queue. The transfer start time is set.

13. The sending server accesses the sending file.

14. The sending server uses a component called a file handler to read records from the sending file. If specified in the file-transfer request, the sending server uses a user-written file handler. User-written file handlers are described in the NetView FTP Customization guide.

15. The sending server compresses the data to be transferred, using the type of compression specified in the request.

16. The sending server encrypts the data if data encryption is specified in the request. The encryption labels in the request must be matching at both the local and remote locations. For information on encryption, refer to the OS/VS1 and OS/VS2 MVS Programmed Cryptographic Facility: Installation Reference Manual.
17. The sending server passes the data to ACF/VTAM, which controls its transport to the receiving location.

18. The receiving server receives the data from ACF/VTAM, decrypts and decompresses it, and, if specified in the request, converts and pads the data before it is written to the receiving file.

19. The receiving server uses a file handler to write the data to the receiving file. If specified in the file-transfer request, the receiving server uses a user-written file handler. User-written file handlers are described in the NetView FTP Customization guide.

20. Steps 14 through 19 are repeated until the entire contents of the sending file are written to the receiving file.

   Periodically during the file transfer the receiving server records how many records have been transferred successfully. This is called taking checkpoints. The checkpoint interval is described in “Specifying Values for the Server Initialization Parameters” on page 113.

21. When the transfer of data is finished, both the sending and the receiving servers close their files.

22. Both the sending and receiving servers call their post-transfer user-exit routines, deallocate the files sent and received, and remove their established security environments. The post-transfer user-exit routines can record statistics or perform any post-transfer processing. For further information on the post-transfer user-exit routine, refer to the NetView FTP Customization guide.

23. One of the two servers terminates the ACF/VTAM conversation.

24. The requesting server calls the post-conversation user-exit routine.

25. Each server sends a file-transfer report file and file-transfer completion message to the user specified in the request.

26. Each server sends transfer end messages to the operator console.

27. The requesting server adds information about the outcome of the file transfer to the request.

28. The requesting server asks the queue handler to put the updated request back into the queue and to change its status to finished. Depending on the outcome of the file transfer, the request is marked as being successfully or unsuccessfully finished. Unsuccessfully finished requests remain on the request queue and can be restarted.

29. If the requesting server was started in continuous mode, it tells the queue handler to get a new file-transfer request, and the process begins again from step 1. If the server was started in single mode, it stops automatically. The server running modes (continuous and single) are described in “Specifying Values for the Server Initialization Parameters” on page 113.

   Note: The preceding discussion considers only one local-request handler on one server. In practice, up to 99 local servers can be started, with each server having up to 32 local-request handlers.
OSI Server

The following describes how a file is transferred by an OSI server.

An OSI server runs in continuous mode and processes requests one at a time. No subtasking is done by an OSI server.

The OSI server uses the OSI/File Services product to perform the transfer, it does not perform the transfer itself. The OSI server does not access any user files, it only passes the information concerning the local user and the requested file transfer to OSI/File Services. OSI/File Services handles the parallel processing of OSI transfers. Only the COPY and MOVE functions of OSI/File Services are supported. For further information on OSI/File Services, refer to the OSI/File Services General Information Manual, or the OSI/File Services System/370 User’s Guide.

For each request, the OSI server performs the following:

1. The OSI server obtains a request from the queue handler, and checks the request and transfer parameters.

2. The OSI server converts the request to the format required by the OSI/File Services application program interface, and then presents it to OSI/File Services. If the function code specified is COPY, the OSI server starts the necessary mechanisms to invoke the COPY function of OSI/File Services. The following actions are performed:

   a. The OSI server checks if the user (the request issuer) is authorized to perform the request. This is done by verifying entries in the Local Resource Directory (LRD) and any RACF* security parameters.

   b. Parameter values and parameter combinations are checked for validity. The parameter values are moved into the corresponding fields of the OSI/File Services interface module.

   c. If the user wants to transfer a file that belongs to the filestore of the OSI server but the file is not yet registered in that filestore, the OSI server temporarily adds the local file name to its filestore. A file belongs to the filestore of the OSI server if the first qualifier of its MVS name is registered in that filestore.

   d. The OSI server uses the VTAM/APPC interface to OSI/IFS to invoke the OSI/File Services function EHIINI to start the function COPY.

   e. OSI/File Services provides its own security mechanisms and performs a number of checks to guarantee the correctness of the request. If the request is accepted, the OSI server is informed and the processing of the COPY request is carried out by OSI/File Services.

   f. The OSI server periodically asks the OSI/File Services interface whether the request has finished and gets the result from OSI/File Services. The OSI server analyzes return and condition codes, and invokes the appropriate actions.

   g. The queue handler is called with the NOTIFY command and a message is sent to the user.
If an error occurred while processing one of these steps, the corresponding error handling mechanisms are started. Specifying the function code COPY means to transfer a file between the local and the remote system. After a successful copy, the file is stored in MVS and registered in the OSI/File Services filestore at both the sending and receiving systems.

3. After OSI/File Services has finished the transfer, the result is available at the OSI/File Services interface (success or failure of the transfer). The OSI server converts the format of the result to the format needed by the queue handler, and then passes the result to the queue handler. Therefore, OSI/File Services terminology is used in some situations rather than NetView FTP terminology.

4. Further processing is done asynchronously. After one request is passed to OSI/File Services, the OSI server is ready to process the next request. This is done independently of completion of the previous request.

5. When an OSI/File Services activity is completed, the result is available via the GET_RESULT command at the OSI/File Services interface. The server keeps the information necessary to obtain the result, and continues processing the current request. For further information, refer to either the OSI/File Services System/370 Programming Guide or the OSI/File Services System/370 User’s Guide.

6. The OSI server asks for the result in the next processing cycle. The result of the transfer includes detailed information about the success or failure of the transfer, and the number of bytes transferred. This information is passed to the queue handler.

If the function specified is MOVE, the same steps as for function COPY are performed to process the MOVE request. After a successful MOVE:

- The file is removed from the sending system.
- The file entry is removed from the filestore at the sending system.

## Stopping NetView FTP V2.2.1 MVS Running

You stop NetView FTP V2.2.1 MVS running by stopping the queue handler using the STOP command.

When you stop the queue handler, all running servers are stopped immediately. The file transfers that are currently being processed are interrupted and checkpoints are taken for the SNA file transfers. When you restart the queue handler, any file-transfer request that specified automatic transfer restart is rescheduled when a server that serves its request class becomes available.

You can also stop the queue handler by stopping the server first, using the MODIFY command, and then stopping the queue handler. The file transfers that are currently being processed are either completed or not, depending on the command options you used. For further information, refer to Chapter 3, “Operating NetView FTP V2.2.1 MVS” on page 43.
Part 2. Installation

This part describes how to install NetView FTP V2.2.1 MVS, how to prepare the system for NetView FTP V2.2.1 MVS, and how to verify the installation.
Chapter 2. Installing, Preparing, and Verifying the Installation of NetView FTP V2.2.1 MVS

The following provides a summary of the installation process, from unloading the distribution tape to verifying the installation:

1. Installing NetView FTP V2.2.1 MVS

   This is described in the Program Directory and consists of the following:
   a. Unloading the job generator from the distribution tape
   b. Reading the README file (DVGMXRDE)
   c. Generating the installation and installation verification jobs
   d. Coding the job statement macro (DVGMXJBC)
   e. Coding the generation job (DVGMXJGN) and running it
   f. Running the SMP/E installation jobs for NetView FTP V2.2.1 MVS:
      1) Allocating the SMP/E data sets for NetView FTP V2.2.1 MVS (DVG01SMP)
      2) Allocating the distribution and target libraries (DVG02ALL)
      3) Initializing SMP/E libraries (DVG03ZON)
      4) Performing the SMP/E RECEIVE processing (DVG04REC)
      5) Performing the SMP/E APPLY processing (DVG05APP)
      6) Performing the SMP/E ACCEPT processing (DVG06ACC).
   g. If you are also installing the OSI part, you must run the job DVGLKOSI.
   h. If you are also installing the Japanese Language Feature, you must run the following SMP/E installation jobs:
      1) Allocating the distribution and target libraries (DVG12ALL)
      2) Initializing SMP/E libraries (DVG13ZON)
      3) Performing the RECEIVE/APPLY processing (DVG14RAP)
      4) Performing ACCEPT processing (DVG15ACC).

2. Preparing NetView FTP V2.2.1 MVS

   This is described in “Preparing the NetView FTP V2.2.1 MVS Program” on page 19 and consists of the following:
   a. Defining the load libraries:
      1) Authorized library
      2) MVS link library list.
   b. Installing the SVC routine
   c. Performing IPL after loading
   d. Assigning RACF authorization to started tasks
   e. Setting up ACF/VTAM resources
   f. Submitting jobs to create the request-queue and checkpoint data sets
   g. Copying the NetView FTP V2.2.1 MVS procedures, parameter members, and the sample server group table
   h. Gaining access to the interactive interface routine.
i. If you want to use NetView FTP V2.2.1 MVS for OSI transfers, you also have to define the corresponding load library as authorized library, and prepare the NetView FTP V2.2.1 MVS OSI servers.

j. If you want to use the Japanese Language Feature, you also need to prepare the NetView FTP V2.2.1 MVS jobs for the Japanese Language Feature, and provide access to the interactive interface of the Japanese Language Feature.

3. Verifying the installation of NetView FTP V2.2.1 MVS

This is described in “Verifying the Installation of NetView FTP V2.2.1 MVS” on page 28 and consists of the following:

a. Preparing the start of the NetView FTP V2.2.1 MVS queue handler and servers
b. Starting the NetView FTP V2.2.1 MVS queue handler
c. Starting the NetView FTP V2.2.1 MVS servers
d. Preparing the verification tests
e. Performing the verification tests 1 to 3
f. Verifying the access to the interactive interface routine
g. Verifying the installation of the NetView FTP V2.2.1 MVS OSI part
h. Verifying the installation of the Japanese Language Feature.

Installing NetView FTP V2.2.1 MVS

To install the NetView FTP V2.2.1 MVS, you must unload the contents of the distribution tape, generate and run the installation jobs. This is described in the Program Directory. When you have installed NetView FTP V2.2.1 MVS, go on to “Preparing the NetView FTP V2.2.1 MVS Program” on page 19.

The following figures show the members contained on the distribution tape and those generated by the installation job generator.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMXJBC</td>
<td>Contains the macro that generates the JOBCLASS statements for all installation and installation verification jobs.</td>
</tr>
<tr>
<td>DVGMXJGN</td>
<td>Contains the job that generates the installation jobs and installation verification jobs.</td>
</tr>
<tr>
<td>DVGMXJG1</td>
<td>Contains the macro that generates the installation jobs.</td>
</tr>
<tr>
<td>DVGMXJG2</td>
<td>Contains the macro that generates jobs for running NetView FTP V2.2.1 MVS and for installation verification.</td>
</tr>
<tr>
<td>DVGCXLMD</td>
<td>Contains the macro that generates VTAM* logon modes for NetView FTP V2.2.1 MVS servers.</td>
</tr>
<tr>
<td>DVGMXNLS</td>
<td>Contains the macro that generates additional jobs for national language support (NLS) installation.</td>
</tr>
<tr>
<td>DVGMXOSI</td>
<td>Contains the macro that generates additional jobs to install and customize the OSI part.</td>
</tr>
<tr>
<td>DVGMXPGD</td>
<td>Contains a copy of the Program Directory.</td>
</tr>
<tr>
<td>DVGMXRDE</td>
<td>Contains some last-minute information about the product.</td>
</tr>
</tbody>
</table>
**Figure 4. Parameters for the SMP/E Installation of NetView FTP V2.2.1 MVS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGINDD1</td>
<td>Allocate the SMP/E CSI cluster.</td>
</tr>
<tr>
<td>DVGINDD2</td>
<td>Initialize the SMP/E CSI cluster.</td>
</tr>
<tr>
<td>DVGINDD3</td>
<td>NetView FTP V2.2.1 MVS SMP/E UCLIN.</td>
</tr>
<tr>
<td>DVGINDD4</td>
<td>SMP/E RECEIVE control statements.</td>
</tr>
<tr>
<td>DVGINDD5</td>
<td>SMP/E APPLY control statements.</td>
</tr>
<tr>
<td>DVGINDD6</td>
<td>SMP/E ACCEPT control statements.</td>
</tr>
<tr>
<td>DVGINDD7</td>
<td>Delete SMP/E data sets (only for newly allocated SMP/E CSI).</td>
</tr>
<tr>
<td>DVGINDD8</td>
<td>Delete NetView FTP V2.2.1 MVS data sets.</td>
</tr>
</tbody>
</table>

**Figure 5. Jobs for the SMP/E Installation of NetView FTP V2.2.1 MVS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVG01SMP</td>
<td>Allocate SMP/E data sets.</td>
</tr>
<tr>
<td>DVG02ALL</td>
<td>Allocate NetView FTP V2.2.1 MVS DISTRIBUTION and TARGET libraries.</td>
</tr>
<tr>
<td>DVG03ZON</td>
<td>Initialize the SMP/E CSI cluster.</td>
</tr>
<tr>
<td>DVG04REC</td>
<td>SMP/E RECEIVE job.</td>
</tr>
<tr>
<td>DVG05APP</td>
<td>SMP/E APPLY job.</td>
</tr>
<tr>
<td>DVG06ACC</td>
<td>SMP/E ACCEPT job.</td>
</tr>
</tbody>
</table>

**Figure 6. Cross Reference Jobs and Control Parameter Members**

<table>
<thead>
<tr>
<th>Job</th>
<th>Parameters Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVG01SMP</td>
<td>DVGINDD1 (only for newly allocated SMP/E CSI), DVGINDD7</td>
</tr>
<tr>
<td>DVG02ALL</td>
<td>DVGINDD8</td>
</tr>
<tr>
<td>DVG03ZON</td>
<td>DVGINDD2, DVGINDD3</td>
</tr>
<tr>
<td>DVG04REC</td>
<td>DVGINDD4</td>
</tr>
<tr>
<td>DVG05APP</td>
<td>DVGINDD5</td>
</tr>
<tr>
<td>DVG06ACC</td>
<td>DVGINDD6</td>
</tr>
</tbody>
</table>

**Figure 7. Jobs for the Installation of the NetView FTP V2.2.1 MVS SVC Routine**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCSVC</td>
<td>Job to install the SVC routine.</td>
</tr>
<tr>
<td>DVGJSMPE</td>
<td>Job used with SMP/E to perform the JCLIN command.</td>
</tr>
</tbody>
</table>

**Figure 8. NetView FTP V2.2.1 MVS Samples for the MVS Setup**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMXAPF</td>
<td>Sample MVS APF definition.</td>
</tr>
<tr>
<td>DVGMXLNK</td>
<td>Sample MVS LNKLST definition.</td>
</tr>
<tr>
<td>DVGMXLP</td>
<td>Sample MVS LPALST definition.</td>
</tr>
<tr>
<td>DVGMXSVC</td>
<td>Sample MVS SVC definition.</td>
</tr>
</tbody>
</table>
**Figure 9. NetView FTP V2.2.1 MVS Samples for the VTAM Setup**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCDRSC</td>
<td>Sample VTAM CDRSC definition.</td>
</tr>
<tr>
<td>DVGMODTB</td>
<td>VTAM logon-mode definition.</td>
</tr>
<tr>
<td>DVGSVAPP</td>
<td>VTAM APPL definitions for the NetView FTP V2.2.1 MVS server.</td>
</tr>
</tbody>
</table>

**Figure 10. NetView FTP V2.2.1 MVS Setup**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMXCRD</td>
<td>IDCAM parameters: delete, define, and initialize the NetView FTP V2.2.1 MVS</td>
</tr>
<tr>
<td></td>
<td>checkpoint data set.</td>
</tr>
<tr>
<td>DVGMXCR1</td>
<td>Job to define and initialize the checkpoint data set.</td>
</tr>
<tr>
<td>DVGMXQR1</td>
<td>Job to define and initialize the NetView FTP V2.2.1 MVS request queue.</td>
</tr>
<tr>
<td>DVGCRREP</td>
<td>Job to create log and report files.</td>
</tr>
</tbody>
</table>

**Figure 11. NetView FTP V2.2.1 MVS Procedures and Startup Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVG230QH</td>
<td>Procedure to start the NetView FTP V2.2.1 MVS queue handler.</td>
</tr>
<tr>
<td>DVGQHP</td>
<td>Startup parameters for the NetView FTP V2.2.1 MVS queue handler.</td>
</tr>
<tr>
<td>DVG230nn</td>
<td>Procedure to start the NetView FTP V2.2.1 MVS server nn.</td>
</tr>
<tr>
<td>DVGSVPnn</td>
<td>Startup parameters for the NetView FTP V2.2.1 MVS server nn.</td>
</tr>
</tbody>
</table>

**Figure 12. NetView FTP V2.2.1 MVS Installation Aids and Utilities**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGSVGRP</td>
<td>Sample server group table.</td>
</tr>
<tr>
<td>DVGCPROC</td>
<td>Job to copy the NetView FTP V2.2.1 MVS procedures to a procedure library.</td>
</tr>
<tr>
<td>DVGCPARM</td>
<td>Job to copy the NetView FTP V2.2.1 MVS parameter library.</td>
</tr>
<tr>
<td>DVGMXCR2</td>
<td>Job to reorganize the NetView FTP V2.2.1 MVS checkpoint data set.</td>
</tr>
<tr>
<td>DVGPRD</td>
<td>Job to print the NetView FTP V2.2.1 MVS checkpoint data set.</td>
</tr>
<tr>
<td>DVGPRQU</td>
<td>Job to print the NetView FTP V2.2.1 MVS request queue.</td>
</tr>
</tbody>
</table>

**Figure 13. Jobs for the NetView FTP V2.2.1 MVS Installation Verification Test**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMBFV</td>
<td>Job to allocate the test data set.</td>
</tr>
<tr>
<td>DVGMBAD1</td>
<td>Job to add a NetView FTP V2.2.1 MVS file-transfer request—verification test 1.</td>
</tr>
<tr>
<td>DVGMBQR1</td>
<td>Job to query a NetView FTP V2.2.1 MVS file-transfer request—verification test 1.</td>
</tr>
<tr>
<td>DVGMBAD2</td>
<td>Job to add a NetView FTP V2.2.1 MVS file-transfer request—verification test 2.</td>
</tr>
<tr>
<td>DVGMBMY2</td>
<td>Job to modify a NetView FTP V2.2.1 MVS file-transfer request—verification test 2.</td>
</tr>
<tr>
<td>DVGMBQR2</td>
<td>Job to a user query and an administrator query for a NetView FTP V2.2.1 MVS</td>
</tr>
<tr>
<td></td>
<td>file-transfer request—verification test 2.</td>
</tr>
<tr>
<td>DVGADM</td>
<td>Job to query the request queue (administrator functions)—verification test 3.</td>
</tr>
</tbody>
</table>
Preparing the NetView FTP V2.2.1 MVS Program

This section describes the definitions, the authorizations, and the allocations that are necessary to run NetView FTP V2.2.1 MVS.

**Note:** DVG.NFTP230 is the default value of the INDX parameter in the job DVGMXJGN and thus the default high-level data-set qualifier. You can change the high-level qualifier by assigning another value to the INDX parameter in the job DVGMXJGN when you run the installation job generator.

Values shown in *lowercase italics* must be replaced by the values appropriate for your installation.

Defining the Load Libraries

This section describes how to define the NetView FTP V2.2.1 MVS libraries as authorized and as link-list libraries.

**Authorized Library**

You have to define the DVG.NFTP230.SDVGLMD0 and the DVG.NFTP230.SDVGLPA load libraries as Authorized Program Facility (APF) authorized libraries. Do this by adding the library names to the currently used member IEAAPFxx of the SYS1.PARMLIB, where xx is a number between 00 and 99 and is depends on the system installation. The sample member DVGMXAPF shows how to do this.

**Note:** You have to define any library that is concatenated to DVG.NFTP230.SDVGLMD0 in a JCL JOBLIB or STEPLIB statement as an APF authorized library.
MVS Link-Library List

Library `DVG.NFTP230.SDVGLMD2` contains modules that perform common functions:

**DVGMCQI**  Invokes the NetView FTP V2.2.1 MVS SVC. This module can reside in a library that is concatenated to the link-list library.

**DVGCIRQX**  Is the pre-queuing user-exit routine. This module must reside in a library that is concatenated to the link-list library.

**Note:** The only way to run multiple NetView FTP MVS systems concurrently is to use a locally defined link-list library, as all NetView FTP MVS systems share the pre-queuing user-exit routine DVGCIRQX.

Both modules, DVGMCQI and DVGCIRQX, should reside in a library that is concatenated to the link-list library. To achieve this you can do either of the following:

- **Use the default NetView FTP V2.2.1 MVS library:**
  1. Define the `DVG.NFTP230.SDVGLMD2` library as a link-list library and add its name to the member of the SYS1.PARMLIB that has the name LNKLSTxx, where `xx` is a number between 00 and 99 and depends on the installation of your system.
  2. Catalog `DVG.NFTP230.SDVGLMD2` in the System Master Catalog.

  **Note:** If you have specified `LINKAUTH=APFTAB` for your current system, you also have to add the library `DVG.NFTP230.SDVGLMD2` to your currently used IEAAPFxx member.

- **Use a locally defined link-list library:**
  - Move or copy DVGCIRQX and DVGMCQI to a library that is already defined in the active LNKLSTxx member of SYS1.PARMLIB.

    **Note:** When doing this you must customize the SMP/E setup, otherwise the modules will not be maintained further and this might result in service problems.

  - Move or copy DVGCIRQX to a library that is already defined in the active LNKLSTxx member of SYS1.PARMLIB.

    **Note:** When doing this you must customize the SMP/E setup, otherwise the module will not be maintained further and this might result in service problems.

For DVGMCQI, concatenate `DVG.NFTP230.SDVGLMD2` to the JOBLIB or STEPLIB statement of the batch job, of the application interface routine, and of the server startup job, and to the ISPLLIB of the interactive interface.

As the server load library `DVG.NFTP230.SDVGLMD0` is APF authorized, ensure that also the library `DVG.NFTP230.SDVGLMD2` is APF authorized in your currently used IEAAPFxx member.
Installing the SVC Routine

To install the NetView FTP V2.2.1 MVS SVC routine, you have to run the following jobs. They are located in the sample library `DVG.NFTP230.INSTLIB`.

**DVGCSVC**  
This job creates the NetView FTP V2.2.1 MVS SVC routine and the load module DVGMCQI.

**DVGJSMPE**  
This job adds the linkage editor JCL (DVGCSVC) to your SMP/E environment and enables the maintenance of the load modules.

The SVC number corresponds to the SVC number you have specified for the parameter SVCNR in DVGMXJGN.

1. Obtain an unused type-3 SVC number from your MVS system programmer. The sample member DVGMXSVC shows how to define the SVC in the currently used IEASVCxx member of the SYS1.PARMLIB.

2. You can use the default NetView FTP V2.2.1 MVS library or an MVS system library to store the SVC routine.

   If you use the default NetView FTP V2.2.1 MVS library `DVG.NFTP230.SDVGLPA`, you have to add it to the currently used LPA library list. The sample member DVGMXLPA shows how to define the SVC in the currently used LPALSTxx member of the SYS1.PARMLIB.

   If you choose to use an already defined LPA library, change the JCL accordingly.

   **Note:** When doing this, you must customize your SMP/E setup, as otherwise the NetView FTP V2.2.1 MVS SVC cannot be maintained further and this might result in service problems.

3. The SVC load-module name must have the form `IGC00nnx`, where:

   - **nn**  
     Are the first two digits of the SVC number.

   - **x**  
     Is one of the following characters, and corresponds to the last digit of the SVC number:

     | Character | Corresponds to |
     |-----------|---------------|
     | A         | 1             |
     | B         | 2             |
     | C         | 3             |
     | D         | 4             |
     | E         | 5             |
     | F         | 6             |
     | G         | 7             |
     | H         | 8             |
     | I         | 9             |
     | {         | 0             |

   For example, if the SVC number is 230, `nnx` is 23{.

4. Submit the job contained in the member DVGCSVC.

   This job does the following:

   **Step 1**  
   Assembles the SVC calling module DVG$$QIS.

   **Step 2**  
   Links the SVC calling module DVGMCQI.

   **Step 3**  
   Links the SVC load module.

   **Note:** This requires that the SMP/E ACCEPT step was successful.
5. Submit the job DVGJSMPE. This job adds DVGCSVC to the SMP/E environment and allows to maintain it. If this job is not suited to your system, modify it and then submit it.

Performing IPL after Loading
Now you must perform the IPL on the MVS machine with the create link pack area (CLPA) option specified according to your local procedures. This refreshes the link pack area and authorizes the new load libraries.

Authorizing the Started Tasks for RACF
If your installation uses RACF, define the queue handler procedure DVG230QH and the server procedures DVG230nn in the started task table before they can run.

In installations of NetView FTP V2.2.1 MVS with the OSI part, the OSI server procedure DVG230SI must also be defined in the started task table.

In most cases, the name of the started task table is ICHRIN03. Contact your system-security administrator for your local conventions.

For information on RACF, refer to the System Programming Library: Resource Access Control Facility (RACF) manual.

Setting Up ACF/VTAM Resources
DVGSVAPP is the definition of the VTAM major node for the NetView FTP V2.2.1 MVS server LUs. Copy this definition to the VTAMLST that is available to ACF/VTAM. It is recommended to add this major node definition to the currently used VTAM configuration member ATCCONnn in the VTAMLST library. This ensures that the server LUs are activated after a VTAM restart.

DVGCDDRSC is a sample member for a VTAM Cross-Domain Resource (CDRSC) definition.

Use the sample data in DVGMODTB to assemble a VTAM logon mode table, and link-edit it into the VTAMLIB library that is available to ACF/VTAM. The default logon mode used by NetView FTP V2.2.1 MVS is FTPBIND. This logon mode is defined with a maximum request-unit size of 256 bytes so that communication with any partner system is possible. If you do not have to use a request-unit size of 256 bytes, select one of the other logon modes defined in the sample file to improve the transfer performance. It is recommended that you use logon mode NFTP2K.

For more information about tuning NetView FTP V2.2.1 MVS see Appendix E, “Factors That Affect the Performance of NetView FTP V2.2.1 MVS” on page 209.

If the samples are not suited to your system, you can modify them. How to do this is described in Chapter 6, “Setting Up and Maintaining the ACF/VTAM Resources Used by NetView FTP V2.2.1 MVS” on page 79.
Submitting Jobs to Create the Request Queue and the Checkpoint Data Set

Use the sample jobs to create the request queue and the checkpoint data set.

If the contents of the sample jobs are not suited to your system, modify the jobs and then submit them. How to do this is described in “Setting Up the Request Queue” on page 67, and “Setting Up the Checkpoint Data Set” on page 73.

The sample jobs are members in the **DVG.NFTP230.INSTLIB**:  

**DVGMXREQ1** Define and initialize the NetView FTP V2.2.1 MVS request queue data set.

**DVGMXCR1** Define and initialize the NetView FTP V2.2.1 MVS checkpoint data set.

Submitting Jobs to Create the Log File and Report File

Use the sample job DVGCRREP to create the queue-handler log file and the server log and report files.

Copying the NetView FTP V2.2.1 MVS Procedures, Parameter Members, and the Sample Server Group Table

Ensure that the sample jobs follow the conventions of your system. The members are located in the **DVG.NFTP230.INSTLIB**:

**DVGCPROC** Copies the NetView FTP V2.2.1 MVS queue-handler and server procedures to a system-defined procedure library, usually SYS1.PROCLIB.

However, you can also define the **DVG.NFTP230.PROCLIB** as a system-known procedure library.

**DVGCPARM** Copies the sample server-group table (DVGSVGRP) and the members containing the startup parameters for the NetView FTP V2.2.1 MVS queue handler and servers to the NetView FTP V2.2.1 MVS parameter library.

**DVGSVGRP** Is the sample server-group table. The server-group name **MVS** is used during the installation verification tests 1 and 2.

Refer to Chapter 7, “Setting Up a Server Group Table” on page 85 for a description of how to set up the server-group table.
Gaining Access to the Interactive Interface Routine

The NetView FTP V2.2.1 MVS interactive interface routine allows TSO users to use Interactive System Productivity Facility (ISPF) panels to add, query, modify, restart, and delete requests. The following describes how to access the interactive interface routine.

**Note:** ISPF must be installed if you want to use the interactive interface. If you want to save requests or use the PDS selection and exclusion facility, you must also have ISPF/PDF installed.

You can gain access to the NetView FTP V2.2.1 MVS interactive interface in one of the following ways:

- Add the NetView FTP V2.2.1 MVS libraries to the TSO logon procedure.

```
//ISPLLIB DD
// DD DSN=DVG.NFTP230.SDVGLMD1,DISP=SHR
//ISPMLIB DD
// DD DSN=DVG.NFTP230.SDVGMSG0,DISP=SHR
//ISPPLIB DD
// DD DSN=DVG.NFTP230.SDVGPNL0,DISP=SHR
//ISPTLIB DD
// DD DSN=userid.ISPF.ISPTLIB,DISP=SHR
//ISPTABL DD
// DD DSN=userid.ISPF.ISPTLIB,DISP=SHR
```

*Figure 16. Access to the Interactive Interface Using the TSO Logon Procedure*

- Add the NetView FTP V2.2.1 MVS libraries to the TSO startup command list (CLIST).

```
ALLOC F(ISPLLIB) DA('DVG.NFTP230.SDVGLMD1' + other libraries ) SHR
ALLOC F(ISPPLIB) DA('DVG.NFTP230.SDVGPNL0' + other libraries ) SHR
ALLOC F(ISPMLIB) DA('DVG.NFTP230.SDVGMSG0' + other libraries ) SHR
ALLOC F(ISPTLIB) DA('&SYSUID..ISPF.ISPTLIB' + other libraries ) SHR
ALLOC F(ISPTABL) DA('&SYSUID..ISPF.ISPTLIB' + other libraries ) SHR
```

*Figure 17. Access to the Interactive Interface Using the TSO Startup CLIST*
Use the ISPF LIBDEF statement.

A sample member DVGMXRII is shipped in the sample library SDVGSAR0/ADVGSAR0.

PROC 0
/*-----------------------------------------------*/
/* Sample Member DVGMXRII */
/*-----------------------------------------------*/
ISPEXEC LIBDEF ISPPLIB DATASET ID('DVG.NFTP230.SDVGPNL0')
ISPEXEC LIBDEF ISPLLIB DATASET ID('DVG.NFTP230.SDVGMSG0')
ISPEXEC LIBDEF ISPTLIB DATASET ID('DVG.NFTP230.SDVGLMD1')
ISPEXEC LIBDEF ISPTABL DATASET ID('&SYSUID..ISPF.ISPTLIB')
ISPEXEC SELECT PGM(DVGIFII) PARM(E)
EXIT

Figure 18. Access to the Interactive Interface Using the ISPF LIBDEF Statement

Here, the NetView FTP V2.2.1 MVS Interactive Interface is immediately invoked and the libraries are deallocated after leaving the primary NetView FTP V2.2.1 MVS panel. The parameter PARM(E) indicates that the panels are in English.

Note: It is required that the &SYSUID..ISPF.ISPTLIB exists already for each TSO user.

When adding the libraries to the TSO logon procedure or to the TSO startup CLIST you must also add the following to your primary ISPF panel or any local selection panel:

F +NetView FTP - NetView File Transfer Program
.
.
&ZSEL = TRANS( TRUNC (&ZCMD,'.')
.
 F,'PGM(DVGIFII) PARM(E)'
 , ,
 X,'EXIT'
 *,?'
 )

Figure 19. Access to the Interactive Interface from the Primary ISPF Panel

Note: The parameter PARM(E) indicates that the panels are in English.

The performance of the NetView FTP V2.2.1 MVS interactive interface can be improved when the NetView FTP V2.2.1 MVS load library DVG.NFTP230.SDVGLMD1 is added to the link-list library concatenation.
Preparing the OSI Part of NetView FTP V2.2.1 MVS

If you want to use NetView FTP V2.2.1 MVS for OSI transfers, you also must define DVG.NFTP230.SDVGLMD3 as an APF authorized library and prepare the NetView FTP V2.2.1 MVS OSI server.

Defining SDVGLMD3 as an Authorized Library

The library DVG.NFTP230.SDVGLMD3 has to be defined as an APF authorized library.

Note: You must define any library that is concatenated to DVG.NFTP230.SDVGLMD3 in a JCL JOBLIB or STEPLIB statement as an APF authorized library.

Preparing the NetView FTP V2.2.1 MVS OSI Server

1. Submit the job DVGCROSI to create the NetView FTP V2.2.1 MVS OSI server log, report and recovery files.
2. Copy the OSI server procedure DVG230SI to a system-defined procedure library.
3. Copy the member DVGSVPSI containing the OSI server startup parameters to the NetView FTP V2.2.1 MVS parameter library.
4. Install OSI/File Services and the OSI/Communications Subsystem and define each OSI server as a user of OSI/File Services with the necessary Local Resource Directory (LRD) entries.

For more information on how to do this, refer to the OSI/File Services Installation and Customization Guide, MVS.
5. Ensure that the IBM C/370 runtime library is available.

Preparing the Japanese Language Feature of NetView FTP V2.2.1 MVS

This section describes the definitions, the authorizations, and the allocations necessary to run the Japanese Language Feature of NetView FTP V2.2.1 MVS.

Preparing the NetView FTP V2.2.1 MVS Jobs for the Japanese Language Feature

The Japanese Language Feature requires that the preparation of NetView FTP V2.2.1 MVS has been finished successfully. This is described in “Preparing the NetView FTP V2.2.1 MVS Program” on page 19.

Change the NetView FTP V2.2.1 MVS jobs as follows:

- Specify LANG = JAPANESE instead of LANG = ENGLISH in the queue handler procedure DVG230QH and in the server procedures DVG230nn.
- Specify PARM='JAPANESE' in the EXEC statements of the jobs for the installation verification test.
Gaining Access to the Interactive Interface Routine of the Japanese Language Feature

For the Japanese version of the interactive interface routine you must first make the same allocations as for the NetView FTP V2.2.1 MVS interactive interface routine, described in “Gaining Access to the Interactive Interface Routine” on page 24, and you must allocate additionally the ISPF panel and message libraries for the Japanese Language Feature.

The libraries for the national language support (NLS) of NetView FTP V2.2.1 MVS must be allocated additionally. You can do this in one of the following ways:

- Add the NetView FTP V2.2.1 MVS NLS libraries to the TSO logon procedure:

```bash
//ISPPALT DD DSN=DVG.NFTP23.SDVGPNL1,DISP=SHR
// DD
//ISPMALT DD DSN=DVG.NFTP23.SDVGMSG1,DISP=SHR
// DD
```

Figure 20. Access to the NLS Interactive Interface Routine Using the TSO Logon Procedure

- Add the NetView FTP V2.2.1 MVS NLS libraries to the TSO startup CLIST:

```bash
ALLOC F(ISPPALT) DA('DVG.NFTP23.SDVGPNL1' + other libraries) SHR
ALLOC F(ISPMALT) DA('DVG.NFTP23.SDVGMSG1' + other libraries) SHR
```

Figure 21. Access to the NLS Interactive Interface Routine Using the TSO Startup CLIST

- Use the ISPF LIBDEF statement:

```bash
ISPEXEC LIBDEF ISPPALT DATASET ID('DVG.NFTP23.SDVGPNL1')
ISPEXEC LIBDEF ISPMALT DATASET ID('DVG.NFTP23.SDVGMSG1')
ISPEXEC SELECT PGM(DVGIFII) PARM(J)
```

Figure 22. Access to the NLS Interactive Interface Routine Using the ISPF LIBDEF Statement

Here, the NetView FTP V2.2.1 MVS Interactive Interface routine is immediately invoked and the libraries are deallocated after leaving NetView FTP V2.2.1 MVS.

Note: The parameter PARM(J) indicates that the NLS language Japanese is used.

Any of these ways allocates the libraries that contain the Japanese panels and the interactive interface routine. They are concatenated as alternate (ALT) libraries. When a user now invokes the interactive interface routine, ISPF checks whether the user’s terminal supports DBCS and, if so, uses the ALT libraries automatically. If the user’s terminal does not support DBCS, ISPF automatically uses the libraries that contain the default language version of the panels and messages.
Verifying the Installation of NetView FTP V2.2.1 MVS

When you have completed the steps described in “Preparing the NetView FTP V2.2.1 MVS Program” on page 19, follow the steps described here to verify that NetView FTP V2.2.1 MVS is functioning correctly. If any of the sample jobs supplied do not suit your system, you can modify them.

Preparing for the Installation Verification Test

The following list contains some prerequisites for the installation verification tests.

1. Ensure that the queue handler procedure DVG230QH is stored in a system-known procedure library, such as SYS1.PROCLIB.

2. Ensure that the server procedures DVG23001 and DVG23002 are stored in a system-known procedure library, for example, SYS1.PROCLIB.

3. If your installation uses RACF, ensure that the queue handler procedure DVG230QH and the server procedures DVG23001 and DVG23002 have been authorized.

4. Activate the VTAM major node (sample DVGSVAPP) for the server LUs and verify that the resource status of the LUs is connectable (CONCT).

5. The queue handler must be invoked as a started task using a cataloged procedure. A sample procedure, DVG230QH, is contained in library DVG.NFTP230.INSTLIB. If the contents of the sample procedure do not suit your system, modify it and then submit it. How this is done is described in “Procedure to Start the Queue Handler” on page 95.

6. The NetView FTP V2.2.1 MVS servers can be invoked as a started task or by a batch job. If the contents of the sample procedure do not suit your system, modify it and then submit it. In “Procedure to Start a Server” on page 105 is described how to do this. In this verification test, the servers are invoked as started task and started by the queue handler.

7. The sample server-group table DVGSVGRP is used during the installation verification tests 1 and 2.

Starting the Queue Handler

To start the queue handler do the following:

1. Enter S DVG230QH.QH from the operator console. QH identifies the queue handler to the system. This ID is used for further actions.

   When the queue handler starts, it reads its initialization parameters from the file DVGQHP in library DVG.NFTP230.PARMLIB.

2. Wait until the queue handler issues the ready message DVG402I.

   If the queue handler stops before issuing the ready message, check the messages issued to the DVGLOG. The messages are described in the *NetView FTP Messages and Codes* manual. Correct any errors and start the queue handler again.
Starting Servers

After you have successfully completed “Starting the Queue Handler” on page 28, at least two NetView FTP V2.2.1 MVS servers must be active. Check the MVS system console for these messages:

DVG025I SERVER DVG23001 READY
DVG025I SERVER DVG23002 READY

If only one or none of these messages appear, check the messages issued to server log files. The messages are described in the NetView FTP Messages and Codes manual. Correct any errors and restart the server by the appropriate start command:

- Enter F QH,S DVG23001 to start server 1
- Enter F QH,S DVG23002 to start server 2.

Overview of the Installation Verification Tests

The following samples are created by the installation job generator and are located in the library DVG.NFTP230.INSTLIB. In the samples, the values shown in bold characters are generated by the installation generator. You can modify them to suit your own installation. Values shown in lowercase must be replaced by values appropriate for your installation.

For installation verification test 1 and 2, the queue handler, server 01, and server 02 must be active. For installation verification test 3, only the queue handler must be ready.

The sample job DVGMVBFV creates a test data set. This data set is used for the verification test 2.

For verification test 1 the following members are used:

- DVGMVAD1 To add a file-transfer request
- DVGMVQR1 To query a specific file-transfer request.

For verification test 2 the following members are used:

- DVGMVAD2 To add a file-transfer request
- DVGMVQR2 To query all file-transfer requests of this originator
- DVGMVMY1 To release the request.

For verification test 3 the following member is used:

- DVGMVADM To perform administrator queries.
Installation Verification Test 1

In this test, you add a file-transfer request to the request queue and query that specific request.

Adding a NetView FTP V2.2.1 MVS File-Transfer Request
Submit job DVGMVAD1, which is shown in Figure 23. This job adds a NetView FTP V2.2.1 MVS file-transfer request to the request queue. The request is immediately processed.

```
//DVGMVAD1 JOB (accn,n),'programmer-id',CLASS=a,
    MSGCLASS=x,MSGLEVEL=(1,1),
    NOTIFY=userid,USER=userid.
//JOBLIB DD DSN=\DVG.NFTP23.SDVGLMD1,DISP=SHR
//*
//BIADD EXEC PGM=DVGFIBI,REGION=2M
//SYSUDUMP DD SYSOUT=*
//DVGLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSSIN DD *
* Parameters to add a file transfer request
FUNCTION=ADD
XMODE=TO
CLASS=1
PRIORITY=1
RMTNODE=MVS
SSECURP='\c5197'
SFILEID='\DVG.NFTP23.SDVGSAM/zerodot'
MEMBER=(R,DVGMXSDS)
SFILEORG=PO
RNOTIFY=userid
RSECURP='\c5197'
RFILEID='\DVG.NFTP23.RECEIVE.DATA.SET1'
RSTATOPT=MNE
REOPOK=CATLG
RFILEORG=PO
/*
```

Figure 23. DVGMVAD1 - Batch Job to Add a Request to the Request Queue

When the job has successfully finished, check that the message shown in Figure 24 was written to DVGLOG.

```
DVG552I QUEUE HANDLER COMMAND ADD CARRIED OUT FOR USER userid ; RETURN CODE = 0 ;
REASON CODE = 0 ; TIME
DVG552I = hh :mm ; REQUEST NUMBER = n
```

Figure 24. Report of Batch Job to Add a Request to the Request Queue

A unique request number was assigned to the file-transfer request and it is waiting to be processed. Server 01 obtains the request and establishes a session with server 02. The transfer of the specified file starts.
Query a NetView FTP V2.2.1 MVS File-Transfer Request

Submit job DVGMVQR1, which is shown in Figure 25. This job performs a query of the request added by the previously submitted job DVGMVAD1. It retrieves status information about that file-transfer request.

**Note:** The value of the Request Number parameter must correspond to the request number reported in message DVG552I in DVGLOG.

```plaintext
//DVGMVQR1 JOB (accn,n),'programmer-id',CLASS=a,
  MSGCLASS=x,MSGLEVEL=(1,1),
  NOTIFY=userid,USER=userid
//JOBLIB DD DSN=DVG.NFTP230.SDVGLMD1,DISP=SHR
//BIOQYALL EXEC PGM=DVGIFBI,REGION=2M
//SYSUDUMP DD SYSOUT=* 
//DVGMLOG DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSINPUT DD SYSOUT=* 
/*------------------------------------------*/
* Parameters to query a request of this originator
*------------------------------------------*
  FUNCTION=QUERY
  REQUEN=1
  /*
```

*Figure 25. DVGMVQR1 - Batch Job to Query a Request of This Originator*

When the job has finished, check the messages written to DVGLOG. When the request has already been processed, querying the request results in the message shown in Figure 26. When the request has not been processed yet, querying the request results in the message shown in Figure 27.

```plaintext
DVG560I REQUEN=000001;ST=F;PRTY=1;RC=00;RSN=0000;CL=1;FINIS=yy/mm/dd,hh:mm:ss;ORIG=userid;REC=00001000;REQNAME=jobname
```

*Figure 26. Message DVG560I for a Finished Request*

```plaintext
DVG551I REQUEN=000001;ST=W;PRTY=1;RC=00;RSN=0000;CL=1;ADDED=yy/mm/dd;ORIG=userid;REC=00000000;REQNAME=jobname
```

*Figure 27. Message DVG551I for a Waiting Request*

You can periodically submit the job DVGMVQR1 to query the status of the request. The request status varies and can be one of the following:

- **W** The request is waiting for a server to process it.
- **A** The request is active; a server is currently transferring the data set.
- **F** The file transfer is finished.
Continue to submit the job until one of the following is true:

- The return code or reason code given in the message is nonzero and the status of the request is W or F.

There was a problem. Analyze the return and reason codes, these are described in the *NetView FTP Messages and Codes* manual, and correct the problem.

A common problem is that one or both servers are not ready.

If the problem was that the session could not be established because the receiving server had not started until after the sending server tried to establish a session, wait five minutes. Five minutes after its initial attempt, the sending server tries again to establish the session.

If this was not the cause of the problem, stop the two servers. How to do this is described in “Stopping a Server” on page 61. You can then read and analyze the messages that were written to the file defined to DVGLOG in the server JCL. After you have corrected the problem, restart the servers. How to do this is described in “Starting a Server” on page 47.

If the status of the transfer request was:

**F**  Resubmit the job shown in Figure 23 on page 30.

**W**  The request is automatically reprocessed when the servers are restarted.

- The status of the request remains W and the return and reason codes remain zero for more than two minutes.

Check whether the two servers are still running and, if so, which classes they serve. Do this by entering `F QH,D` from the operator console. The system displays the system status.

If server 01 does not serve class 1, stop server 01 as described in “Stopping a Server” on page 61, change the startup parameters accordingly, and restart server 01.

If the server classes are correct, stop both servers and check the messages that were written to the file defined to DVGLOG in the server JCL. After you have corrected the problem, restart the servers. How to do this is described in “Starting a Server” on page 47.

- The status of the request is F, and the return and reason codes are both zero. The file transfer was carried out successfully.

For a complete description of how to create and add requests to the request queue refer to the *NetView FTP V2 MVS User’s Guide.*
Installation Verification Test 2

In this test, you add a file-transfer request to the request queue, query requests on the request queue, and modify a specific request.

Adding a NetView FTP V2.2.1 MVS File Transfer Request

Submit job DVGMVAD2, which is shown in Figure 28.

This job adds a NetView FTP V2.2.1 MVS file-transfer request to the request queue. The request is not processed immediately.

```
//DVGMVAD2 JOB (accn,n),'programmer-id',CLASS=a,
// MSGCLASS=x,MSGLEVEL=(1,1),
// NOTIFY=userid,USER=userid
//JOBLIB DD DSN=DVG.NFTP230.SDVGLMD1,DISP=SHR
//*
//BIADD EXEC PGM=DVGIFBI,REGION=2M
//SYSUDUMP DD SYSOUT=* 
//DVGLOG DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
/*
* Parameters to add a file transfer request
FUNCTION=ADD
XMODE=TO
CLASS=1
PRIORITY=1
RMTNODE=MVS
QSTAT=HOLD
SSECURP='*'
SFILEID='DVG.NFTP230.VERIFY.TEST'
SFILEORG=PS
RNOTIFY=userid
RSECURP='*'
RFILEID='DVG.NFTP230.RECEIVE.DATA.SET2'
RSTATOPT=MNE
REOPOK=CATLG
RFILEORG=PS
*/
```

Figure 28. DVGMVAD2 - Batch Job to Add a Request to the Request Queue

When the job has finished, check that the message shown in Figure 29 is written to DVGLOG.

```
DVG552I QUEUE HANDLER COMMAND ADD CARRIED OUT FOR USER userid;RETURN CODE = 0 ;
REASON CODE = 0 ; TIME =
DVG552I hh:mm:ss;REQUEST NUMBER = 2
```

Figure 29. Report of Batch Job DVGMVAD2 to Add a Request to the Request Queue
Query a NetView FTP V2.2.1 MVS File-Transfer Request

Submit job DVGMVQR2, which is shown in Figure 30.

This job performs two different queries of the request added by the previously submitted job DVGMVAD2. It retrieves status information about all file-transfer requests for the user ID that submitted the job and performs an administrator query for all file-transfer requests on the request queue.

```
//DVGMVQR2 JOB (acct,n),'programmer-id',CLASS=a,
// MSGCLASS=x,MSGLEVEL=(1,1),
// NOTIFY=userid,USER=userid
//JCLIB DD DSN=DVG.NFTP230.SDVGLMD1,DISP=SHR
/***---------------------------------------------------
** Query Requests
**---------------------------------------------------
//QRYALL EXEC PGM=DVGIFBI,REGION=2M
//SYSUDUMP DD SYSOUT=/c5197
//DVLOG DD SYSOUT=/c5197
//SYSPRINT DD SYSOUT=/c5197
//SYSIN DD /c5197
*-----------------------------------------------------
* Parameters to query all requests for this originator
*-----------------------------------------------------
FUNCTION=QRYALL
/EOR
*-----------------------------------------------------
* Administrator function: Query all Requests
*-----------------------------------------------------
FUNCTION=QRYADM
QSTAT=ALL
PASSWORD='SRTF5'
/*
```

Figure 30. DVGMVQR2 - Batch Job to Query a Request of This Originator

The NetView FTP V2.2.1 MVS master password is generated by the installation generator. The default value is SRTF5. Change the value in the PASSWORD control statement of this job if you defined another password in the queue handler startup parameters.

When the job has finished, check that the messages shown in Figure 31 and Figure 32 on page 35 are written to DVGLOG.

```
DVG5601I REQNUM=00001;ST=F;PRTY=1;RC=00;RSN=0000;CL=1;FINIS=yy/mm/dd,hh:mm:ss
ORIG=userid;REC=00001000;RENAME=jobname
DVG551I REQNUM=00002;ST=H;PRTY=1;RC=00;RSN=0000;CL=1;ADDED=yy/mm/dd,hh:mm:ss
ORIG=userid;REC=00000000;RENAME=jobname
```

Figure 31. Messages when Querying All Requests for This Originator
Modifying a NetView FTP V2.2.1 MVS File-Transfer Request

Submit job DVGMVMY2, which is shown in Figure 33. It performs the following steps:

1. Releases the request with the request number \( n \).
2. Queries all requests for the originator.

Note: The value in the REQNUM control statement must correspond to the value reported in message DVG552I in the DVGMVAD2 log.

```bash
//DVGMVMY2 JOB (accn,n),'programmer-id',CLASS=a,
// MSGCLASS=x,MSGLEVEL=(1,1),
// NOTIFY=userid,USER=userid
//JOBLIB DD DSN=DVG.NFTP23.SDVGLMD1,DISP=SHR
//BMODIFY EXEC PGM=DVGIFBI,REGION=2M
//SYSUDUMP DD SYSOUT=* 
//DVGLOG DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *  

*-----------------------------------------------------  
* Release the previously added request  
*-----------------------------------------------------  
FUNCTION=MODIFY  
REQNUM=2  
QSTAT=RELEASE  
/EOR  

*-----------------------------------------------------  
* Parameters to query all requests for this originator  
*-----------------------------------------------------  
FUNCTION=QRYALL  
*/
```

Figure 33. DVGMVMY2 - Batch Job to Modify and Query a File-Transfer Request
When the job has finished, check that the message shown in Figure 34 is written to DVGLOG.

![Log for Job DVGMVMY2](image)

**Figure 34. Log for Job DVGMVMY2**
Installation Verification Test 3

Administrator Function Queries
Submit job DVGMVADM shown in Figure 35. It performs the following:

- Queries the NetView FTP V2.2.1 MVS file-transfer request queue status
- Queries all requests on the request queue
- Queries all finished requests with data set name.

```plaintext
//DVGMVADM JOB (accn,n), 'programmer-id', CLASS=a,
//   MSGCLASS=x, MSGLEVEL=(1,1),
//   NOTIFY=userid, USER=userid
//JOBLIB DD DSN=DVG.NFTP230.SDVG_LMD1, DISP=SHR
//QADM EXEC PGM=DVGIFBI, REGION=2M
//SYSUDUMP DD SYSOUT=*
//DVGLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

*---------------------------------------------------------------
* Administrator function: Query Request Queue
*---------------------------------------------------------------
FUNCTION=QRYADM
QSTAT=GEN
PASSWORD='SRTF5'
/EOR

*---------------------------------------------------------------
* Administrator function: Query all Requests
*---------------------------------------------------------------
FUNCTION=QRYADM
QSTAT=ALL
PASSWORD='SRTF5'
/EOR

*---------------------------------------------------------------
* Administrator function: Query finished Requests
*---------------------------------------------------------------
FUNCTION=QRYADM
QSTAT=(FINISHED, DSN)
PASSWORD='SRTF5'
/
```

Figure 35. DVGMVADM - Administrator Function Queries

The NetView FTP V2.2.1 MVS master password is generated by the installation generator. The default value is SRTF5. Change the value in the PASSWORD control statement of this job if you defined another password in the queue handler startup parameters.
When the job has finished, check that the messages shown in Figure 36 are written to DVGLOG.

| DVG0201 | *---------------------------------------------------------------
| DVG0201 | * Administrator function: Query Request Queue
| DVG0201 | *---------------------------------------------------------------
| DVG5401 | REQUEST QUEUE STATISTICS: QUEUE HANDLER = DVGQHPRO; START DATE = yy/mm/dd; START TIME = hh:mm; QUEUE SIZE = 99
| DVG5411 | WAITING = 0; ACTIVE = 0; FINISHED = 2; RESTARTED = 0; UNUSED = 99
| DVG0201 | *---------------------------------------------------------------
| DVG0201 | * Administrator function: Query all Requests
| DVG0201 | *---------------------------------------------------------------
| DVG5451 | REQUEST NUMBER = 1; CLASS = 1; SUBMITTED = yy/mm/dd, hh:mm; STARTED = yy/mm/dd, hh:mm; FINISHED
| DVG5451 | RETURN CODE = 00; REASON CODE = 0; RECORDS = 1000; ORIGINATOR = userid;
| DVG5451 | REQUEST NUMBER = 2; CLASS = 1; SUBMITTED = yy/mm/dd, hh:mm; STARTED = yy/mm/dd, hh:mm; FINISHED
| DVG5451 | RETURN CODE = 00; REASON CODE = 0; RECORDS = 25; ORIGINATOR = userid;
| DVG0201 | *---------------------------------------------------------------
| DVG0201 | * Administrator function: Query finished Requests
| DVG0201 | *---------------------------------------------------------------
| DVG5451 | REQUEST NUMBER = 1; CLASS = 1; SUBMITTED = yy/mm/dd, hh:mm; STARTED = yy/mm/dd, hh:mm; FINISHED
| DVG5451 | RETURN CODE = 00; REASON CODE = 0; RECORDS = 1000; ORIGINATOR = userid;
| DVG5461 | NAME OF SENDING DATA SET = DVG.NFTP23.SDVGSAM
| DVG5451 | REQUEST NUMBER = 2; CLASS = 1; SUBMITTED = yy/mm/dd, hh:mm; STARTED = yy/mm/dd, hh:mm; FINISHED
| DVG5451 | RETURN CODE = 00; REASON CODE = 0; RECORDS = 25; ORIGINATOR = userid;
| DVG5461 | NAME OF SENDING DATA SET = DVG.NFTP23.VERIFY.TEST

Figure 36. Log of Administrator Function Queries

This completes the verification procedure.
Verifying the Access to the Interactive Interface Routine

The following steps verify the access to the interactive interface routine.

1. Ensure that the NetView FTP V2.2.1 MVS queue handler is running.

2. If the NetView FTP V2.2.1 MVS interactive interface has been allocated via the TSO logon procedure or with a TSO startup CLIST, do the following:
   • From a terminal that is in TSO READY mode, enter:

   **ISPSTART PGM(DVGIFII) PARM(E)**
   The letter you specify as a parameter is the language the panels are written in. E indicates English.

   If the NetView FTP V2.2.1 MVS interactive interface has been allocated via the ISPF LIBDEF statement, call the DVGMXRII CLIST from the ISPF command line.

   NetView FTP V2.2.1 MVS displays the main NetView FTP V2.2.1 MVS selection panel.

3. Enter **PANELID** on the command line. The panel ID shown in the upper left corner of the screen must be DVGME00E.

4. Request help by either entering **HELP** or pressing the HELP PF key. The first NetView FTP V2.2.1 MVS help panel is displayed. It shows the current maintenance level of NetView FTP V2.2.1 MVS.

5. Return to the main selection panel.

This completes the access verification of the interactive interface.

Verifying the Installation of the OSI Part of NetView FTP V2.2.1 MVS

To perform the installation verification test of the OSI part of NetView FTP V2.2.1 MVS do the following:

1. Start the NetView FTP V2.2.1 MVS queue handler as described in “Starting the Queue Handler” on page 28.

2. Make sure that OSI/File Services is active.

3. Start the NetView FTP V2.2.1 MVS OSI server.

   Enter from the operator console:

   **F DVG230QH,S DVG230SI**

4. Let the OSI server run for about ten minutes.

5. Stop the NetView FTP V2.2.1 MVS OSI server.

   Enter from the operator console:

   **F DVG230QH,P DVG230SI**

6. Allow three minutes for termination, then check that the DVG230SI address spaces have terminated.
7. Check the NetView FTP V2.2.1 MVS OSI server log for the following:
   a. The server start message (DVG001I)
   b. The correct OSI server startup parameters
   c. Message DVG471I, confirming that the server session values have been accepted
   d. Request recovery information (messages DVG909I and DVG911I), stating that no requests were to be recovered
   e. The OSI server termination messages DVG312I and DVG002I; DVG002I must state return code 0.

   The NetView FTP V2.2.1 MVS OSI installation verification test is complete.

   Administering NetView FTP V2.2.1 MVS OSI file transfers is described in Chapter 9, “Administering OSI File Transfers” on page 91.

---

**Verifying the Installation of the Japanese Language Feature of NetView FTP V2.2.1 MVS**

After the preparation of the Japanese Language Feature as described in “Preparing the Japanese Language Feature of NetView FTP V2.2.1 MVS” on page 26 has been successfully finished, you are ready to start the installation verification test for the Japanese Language Feature. You can rerun the tests described in “Verifying the Installation of NetView FTP V2.2.1 MVS” on page 28.
Part 3. Operation

This part describes how to operate NetView FTP V2.2.1 MVS.
Chapter 3. Operating NetView FTP V2.2.1 MVS

At the operator console, you can enter commands that:

- Start NetView FTP V2.2.1 MVS (by starting the queue handler)
- Start a server:
  - Automatically
  - By submitting a batch job
  - By using the MODIFY command
  - By using a cataloged procedure.
- Display information about NetView FTP V2.2.1 MVS
- Modify a server's classes
- Modify the maximum number of simultaneous sessions that a server can handle
- Modify a request's class or priority
- Initiate a request-queue rebuild
- Hold and release servers
- Start and stop an internal trace
- Stop:
  - A specific server
  - All servers.
- Stop NetView FTP V2.2.1 MVS (by stopping the queue handler).

Syntax diagrams are used to describe the operator commands. To use a syntax diagram, simply follow a path from left to right, top to bottom, adding elements as you go. In the diagrams, all spaces and other characters are significant.

Each diagram begins with a double right arrow and ends with a right and left arrow pair. Lines beginning with single right arrows are continuation lines.

The following conventions are used:

- Keywords are all in uppercase and should be entered exactly as shown.
- Variable values that you provide are shown in italics, and are usually in lowercase.
- Commas (,), periods (.), equal signs (=), parentheses ((())), and intermediate blanks are significant.

```
  >>> KEYWORD=variable_value_1,variable_value_2
```

- Optional syntax elements are shown below the main line.

```
  >>> COMMAND
        KEYWORD=value
```

© Copyright IBM Corp. 1988, 1994
A repeatable operand is shown like so:

```
KEYWORD=variable_value
```

The following is a sample syntax diagram and the possible valid expressions that it indicates you can form.

Possible valid versions of the START command are:

```
START proname
START proname.id
START proname,SUB=subsystem
START proname.id,SUB=subsystem
START proname,SUB=subsystem,keyword1=option1
START proname.id,SUB=subsystem,keyword1=option1
START proname,SUB=subsystem,keyword1=option1,keyword2=option2
START proname.id,SUB=subsystem,keyword1=option1,keyword2=option2
START proname,keyword1=option1
START proname.id,keyword1=option1
S proname
S proname.id
S proname,SUB=subsystem
S proname.id,SUB=subsystem
S proname,SUB=subsystem,keyword1=option1
S proname.id,SUB=subsystem,keyword1=option1
S proname,SUB=subsystem,keyword1=option1,keyword2=option2
S proname.id,SUB=subsystem,keyword1=option1,keyword2=option2
S proname,keyword1=option1
S proname.id,keyword1=option1
```

Note that the space before the `proname` value is significant.

The key to using the syntax diagrams is to follow them mechanically.
Operator Commands

There are three types of operator commands:

- START operator command
- MODIFY operator command
- STOP operator command.

With the START operator command you can:

- Start the queue handler
- Start a server.

With the MODIFY operator command you can:

- Start a server
- Stop a server
- Stop all servers
- Display the NetView FTP V2.2.1 MVS system status
- Change a server’s server class
- Change the maximum number of simultaneous file transfers that a server can handle
- Change a request’s class
- Change a request’s priority
- Suspend and resume the processing of local requests for one server
- Rebuild the request queue
- Suspend processing of local requests for all servers
- Resume the processing of local requests for all servers
- Start and stop an internal trace for the queue-handler component.

With the STOP operator command you can:

- Stop the queue handler and all servers immediately.

Starting the Queue Handler

Start the queue handler from the operator’s console via the START command. If it is started as a batch job, subsequent MODIFY queue handler commands are not accepted.

The START command starts the queue handler by invoking a cataloged procedure.

The syntax of the START command is:

```
START procname .id SUB=subsystem keyword=option
```

`procname`

The name of a cataloged procedure, residing in a procedure library, that defines the job to be used to start the queue handler. You can use the sample procedure, DVGQHPRO, or you can write your own procedure. The startup procedure for the queue handler is described under “Procedure to Start the Queue Handler” on page 95.
id  This is an optional user-defined name that identifies the queue handler to be started. It can be up to eight characters long. The first character must be a letter. The default value is *procname*. Although this parameter is optional, you should specify it so you can easily identify the queue handler in subsequent MODIFY or STOP commands.

If you do not use the id with the START command, the system automatically uses the procedure name as an identifier.

**SUB**=subsystem

The name of the subsystem that selects the job for processing. The subsystem must be active and must support job selection. The default value is the primary job entry system.

**keyword**=option

If the sample startup procedure, DVGQHPRO, is used to start the queue handler, this parameter is not used. If a user-written procedure containing symbolic parameters is used to start the queue handler, you can use **keyword**=option to override the value of any specified keyword in the cataloged procedure.

**Example:** To start the queue handler using the procedure DVGQHPRO from the NetView FTP V2.2.1 MVS sample library **DVG.NFTP230.INSTLIB**, enter:

```
S DVGQHPRO.QH
```

Once the queue handler has started it issues a message to the operator's console. The queue handler then scans the request queue and builds the request queue directory. The time required depends on the size of the request queue and the performance of your system. It may take several minutes.

Once the queue handler has finished its initialization processing and is ready to accept queue handler commands, it issues a ready message.
Starting a Server

Before you start a server you must decide whether you want job allocation or dynamic allocation (refer to the NetView FTP V2 MVS User’s Guide for more information).

You can start a server in any of the following ways. In the last case you must have a startup job for the server. In the other cases you must have a cataloged startup procedure for the server:

- Automatically, when the queue handler is started by the STARTSVn initialization parameter
- Using the MODIFY command
- Using the START command
- By submitting a batch job.

**Note:** When a server dynamically allocates a data set, it uses the name DVGFIxx in the DD statement (xx is a 2-digit number in the range 01 through 32 that represents the server subtask number that processes the corresponding request). These are reserved names and must not be used in any other context.

Automatically during Queue-Handler Initialization

NetView FTP can be made to start servers automatically during queue handler initialization by using the STARTSVx keywords. See “Specifying Values for the Queue Handler Initialization Parameters” on page 99 for details of how to specify values for these parameters.

Using the START Command

You can use the START command to call a cataloged procedure to start a server.

The syntax of the START command is:

```
| START | procname | .id | ,SUB=subsystem | ,keyword=option |
```

**procname**

The name of the cataloged procedure, residing in a procedure library, that defines the job to be used to start the server. You can use the sample procedure, DVGSTSV, or you can write your own procedure. The startup procedure for the server is described under “Procedure to Start a Server” on page 105.

**id**

An optional user-determined name identifying the server to be started. It can be up to eight characters long. The first character must be alphabetic. If you do not use the *id* with the START command, the system automatically uses the procedure name, *procname*, as an identifier. Although this parameter is optional, you should specify it so you can easily identify the server in subsequent MODIFY or STOP commands.
**SUB=** *subsystem*

The name of the subsystem that is to select the job for processing. The subsystem specified must be active and must support job selection. The default value is the primary job entry subsystem.

For more information see *MVS/ESA Operations Systems Commands* or *MVS/XA Operations Systems Commands*.

**keyword=** *option*

Any appropriate keyword specified to override the corresponding parameter in the cataloged procedure.

For more information see *MVS/ESA Operations Systems Commands* or *MVS/XA Operations Systems Commands*.

**Example:** To start a server using a cataloged procedure named FTPSRV03, enter the following:

```
START FTPSRV03.SV03
```

### Using the MODIFY Command

You can use the MODIFY command to call a cataloged procedure to start a server.

The following command modifies the queue handler to start a server:

```
MODIFY QH,START /03
```

**id** The identifier that you used in the START command to start the queue handler.

**m** One of the following:

- A number (0 through 9 or 00 through 99) that is the suffix of the name of the cataloged procedure of a server. These digits are added to the prefix specified for the server in the initialization parameters for the queue handler. Refer to `SRVPREF` described in “Specifying Values for the Queue Handler Initialization Parameters” on page 99.

- A name of up to eight characters that is the complete name of the cataloged procedure for the server. If a server is started using this method, it must be stopped using the same name.

**Examples:**

```
MODIFY QH,START 03
F QH,START FTPSRV03
```

### Using a Batch Job

You can start a server by submitting a startup job. An example of a startup job is the DVGSTSV member in `DVG.NFTP230.INSTLIB`.
Displaying Information about NetView FTP V2.2.1 MVS

You can display the status of either all NetView FTP V2.2.1 MVS components, the queue handler, or a specific SNA or OSI server, using the MODIFY command. The syntax of the command is as follows:

```
MODIFY id,DISPLAY ALL
```

*id* The identifier that you used in the START command to start the queue handler.

**ALL**

Displays the status of all NetView FTP V2.2.1 MVS components.

**QH**

Displays the status of the queue handler.

**suffix**

The suffix of the procedure name of the server whose status is to be displayed.

**jobname**

The *jobname* of the server whose status is to be displayed. This format must be used if the server has not been started via the queue handler.

Displaying the Status of All NetView FTP V2.2.1 MVS Components

When you enter `DISPLAY ALL`, the queue handler retrieves information about its status and all started servers. Once all the information is available, the data is displayed at the console from which the command was entered.

If a specified server does not pass its status information within 30 seconds the NetView FTP V2.2.1 MVS system status is displayed without that server’s information.

The operator can use the displayed information to help him decide whether to use the MODIFY subcommands to perform actions such as the following:

- Increase or decrease the number of local-request handlers or remote-request handlers, or both
- Hold or release a specific server
- Reassign server classes to a specific server
- Hold or release all file-transfer requests of a request class
- Hold or release all servers.
Figure 37 shows an example of this type of display.

<table>
<thead>
<tr>
<th>NAME</th>
<th>STARTED</th>
<th>LU NAME</th>
<th>ACT LRH</th>
<th>ACT RRH</th>
<th>ST</th>
<th>RM</th>
<th>CLASSES SERVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTPQH</td>
<td>94/10/11</td>
<td>---------</td>
<td>-</td>
<td>-</td>
<td>A</td>
<td>-</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>10:13:12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVJ0801</td>
<td>94/10/11</td>
<td>XQXSTR14</td>
<td>12</td>
<td>10</td>
<td>A</td>
<td>C</td>
<td>1234567890ABCDE</td>
</tr>
<tr>
<td></td>
<td>11:13:12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVJ0802</td>
<td>94/10/11</td>
<td>XXEASV2</td>
<td>2</td>
<td>6</td>
<td>A</td>
<td>S</td>
<td>19A</td>
</tr>
<tr>
<td></td>
<td>11:14:23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSISERV</td>
<td>94/10/11</td>
<td>OSISERV</td>
<td>/zerodot</td>
<td>/zerodot</td>
<td>A</td>
<td>C</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>11:15:46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DVG448I</td>
<td>3 of 3 SERVERS DISPLAYED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 37. Example of a NetView FTP V2.2.1 MVS Display

The display contains one row of queue handler information and one row of information for each active server.

The meaning of the column headers is as follows:

**NAME**
This is the job name or STC name under which the queue handler or the servers are started.

**STARTED**
The startup time of the address space.

**LUNAME**
The LU name or OSI server name of a server within the local NetView FTP V2.2.1 MVS.

**ACT LRH**
This is the number of currently active local-request handlers for the server.

**ACT RRH**
This is the number of currently active remote-request handlers for the server. This value is always zero for an OSI server.

**ST** Status. Can be one of the following:
- A Active
- T Terminating delayed
- H Held
- C Closed.

**RM**
Running mode. Can be one of the following:
- S Single mode
- C Continuous mode
- G Continuous mode with GDG support.

**CLASSES SERVED**
All server classes that are served by this server are listed.

For more details, the DISPLAY commands shown in the following sections can be invoked.
Displaying the Status of the Queue Handler

When you enter DISPLAY QH, NetView FTP V2.2.1 MVS displays the status of the queue handler and request queue. The operator can use this information to help decide whether to perform actions such as the following:

- Initialize queue rebuild
- Start or stop one or more servers
- Reassign server classes to a specific server
- Increase or decrease the number of local-request handlers or remote-request handlers for one or more servers.

Figure 38 shows this type of display.

```
DVG458I DISPLAY OF QUEUE HANDLER INFORMATION

<table>
<thead>
<tr>
<th>QH NAME</th>
<th>STARTED</th>
<th>SIZE</th>
<th>WAIT</th>
<th>ACT</th>
<th>FIN</th>
<th>%FULL</th>
<th>DLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTPQH</td>
<td>12:27:46</td>
<td>9999</td>
<td>458</td>
<td>14</td>
<td>33</td>
<td>5%</td>
<td>4</td>
</tr>
</tbody>
</table>

NUMBER OF WAITING REQUESTS IN EACH CLASS

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>CLASSES</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C</td>
<td>22</td>
</tr>
<tr>
<td>1</td>
<td>D</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>E</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>G</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>H</td>
<td>15</td>
</tr>
</tbody>
</table>

Figure 38. Panel Layout when the Queue Handler is Displayed

The display contains one row with general queue information and several rows of request class information.

The columns contain the following:

**QH NAME**

Name used to start the queue handler.

**STARTED**

Startup time of the queue handler.

**SIZE**

Maximum number of file-transfer requests that can be queued.

**WAIT**

Number of waiting file-transfer requests on the request queue.

**ACT**

Number of active file-transfer requests on the request queue.

**FIN**

Number of finished file-transfer requests on the request queue.

**%FULL**

Percentage of the slots on the request queue that are filled.

**DLY**

Minimum number of hours that finished file-transfer requests are to be left on the queue.
Displaying the Status of a Specific Server

When you enter **DISPLAY suffix** or **DISPLAY jobname**, the queue handler requests the server specified by either **suffix** or **jobname** to display its current status.

**suffix**

The suffix of the procedure name of the server to be displayed.

**jobname**

The **jobname** or STC name of the server. This format must be used if the server has not been started via the queue handler.

This information enables you to answer questions such as the following:

- How many simultaneous sessions and of which type can be served?
- If a server must be stopped, how should it be stopped (delayed or immediately)?
- Who originated the file-transfer request?
- Which LUs are involved in the transfer?
- When did the transfer start?

Figure 39 shows an example of the display for an SNA server. An example of the display for an OSI server is shown in Figure 40 on page 54.

<table>
<thead>
<tr>
<th>LU NAME</th>
<th>TRANSFER ORIGIN</th>
<th>TRANSFER STARTED</th>
<th>REQUESTED</th>
<th>REMOTE ORIGIN</th>
<th>REMOTE STARTED</th>
<th>RHH</th>
<th>RECORDED</th>
<th>DATA SET NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>XQXSTR14</td>
<td>KEVIN</td>
<td>11:05:00</td>
<td>12</td>
<td>NFTMVS01</td>
<td>SDFMVS24</td>
<td>63846</td>
<td>DAT.PDS01.TRSV614</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FRED</td>
<td>11:12:22</td>
<td>3</td>
<td>XXX33</td>
<td>HHH123</td>
<td>12</td>
<td>DAP.DFS01.C614</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CARL</td>
<td>11:08:52</td>
<td>125</td>
<td>X34T3</td>
<td>TH</td>
<td>1112</td>
<td>GFPDFA1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MARIE</td>
<td>11:10:53</td>
<td>D</td>
<td>2225</td>
<td>FDSSE</td>
<td>332</td>
<td>GFP.DFA1.DNBV614</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PETER</td>
<td>11:12:52</td>
<td>E</td>
<td>5</td>
<td>QWERT</td>
<td>22</td>
<td>IMSFH01</td>
<td></td>
</tr>
</tbody>
</table>

Figure 39. Panel Layout when the Status of a Server Is Displayed

The display contains one row with general server information and a row for each currently active file-transfer request.
The columns contain the following:

**LU NAME**
The LU name of the currently displayed server.

**STARTED**
The startup time of the address space.

**MAX SES**
Number of sessions that can be handled by this server. This limit cannot be changed.

**AVL SES**
Maximum number of sessions that are currently possible. This limit can be changed.

**AVL LRH**
Maximum number of local-request handlers that are currently possible. This limit can be changed.

**ACT LRH**
Number of currently active local-request handlers.

**AVL RRH**
Maximum number of remote-request handlers that are currently possible. This limit can be changed.

**ACT RRH**
Number of currently active remote-request handlers.

**CLASSES SERVED**
All server classes that are served by this server.

**ORIGIN. UID/JOB**
The originator ID of the file-transfer request.

**TRANSFER STARTED**
The time when the file transfer for the file-transfer request was started.

**CL**
The request class of the file-transfer request.

**RH TYPE**
Request handler type. Can be one of the following:

- **LRH**
  Indicates a local file-transfer request handler

- **RRH**
  Indicates a remote file-transfer request handler.

**REQNUM**
The request number that is currently being served. If no request number is available, N/A is displayed. This can happen for remote request handlers when the transfer partner is NetView FTP/400.
REMOTE LU NAME
The LU name of the remote server with which a conversation is to be established.

For transfers to or from a NetView FTP Client program, it is the LU name of the workstation running the NetView FTP Server program.

REMOTE SERVER GROUP
The server group specified in the file-transfer request.

RECORDS TRANSFERRED
The number of records of the data set that have already been transferred.

DATA SET NAME
The name of each data set currently being transferred is also displayed, as follows:

- If the data-set name was specified in the file-transfer request, the data-set name is displayed.
- If the data-set name was specified via a server DD statement, and the file is referenced via the filename statement in the file-transfer request, the ddname is displayed.
- If the file type of the data set is USER, the name of the user-written file handler is displayed.

Figure 40 shows an example of the display for an OSI server.

<table>
<thead>
<tr>
<th>LU NAME</th>
<th>STARTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSISERV</td>
<td>94/10/11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ORIGIN</th>
<th>TRANSFER</th>
<th>RH</th>
<th>REQ</th>
<th>AET</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOHN</td>
<td>11:14:00</td>
<td>0</td>
<td>LRH</td>
<td>12345 REMAET1</td>
</tr>
<tr>
<td>FILE NAME:</td>
<td>OSI.DAT01.TRSV614</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRED</th>
<th>TRANSFER</th>
<th>RH</th>
<th>REQ</th>
<th>AET</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE NAME:</td>
<td>OSI.DFS01.C614</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 40. Panel Layout when the Status of an OSI Server Is Displayed

The display contains one row with general server information and a row for each currently active file-transfer request.

The columns contain the following:

LU NAME
The name of the currently displayed OSI server.

STARTED
The date and time when the OSI server was started.
MAX SES
Maximum number of transfer requests the OSI server is allowed to handle.
This limit cannot be changed, it is fixed at startup time.

AVL SES
Maximum number of transfer requests the OSI server is currently able to handle.

AVL LRH
Number of remotely initiated file-transfer requests the OSI server can handle.
This value is always zero.

ACT LRH
Number of currently active FTAM requests in the OSI server. This value can be different to the number of transfers already started by OSI/File Services.

AVL RRH
This value is always zero.

ACT RRH
This value is always zero for an OSI server.

CLASSES SERVED
Classes that are served by this OSI server. Only those classes allowed for OSI file transfers can be specified.

ORIGIN. UID/JOB
The originator ID of the file-transfer request.

TRANSFER STARTED
The time when the file transfer is handled by OSI/File Services.

CL
The server class specified in the OSI file-transfer request.

RH TYPE
Request-handler type. For an OSI server, this is always type LOCAL.

REQNUM
The request number that is currently being served.

AET
The remote Application Entity Title of the partners FTAM application.

FILE NAME
The local file name, the MVS file name without the first qualifier if the transfer is local-to-remote, or the OSI file name, the name by which the file is identified and accessed by the remote FTAM application if the transfer is remote-to-local. This is the information specified by the user when initiating the request.
Modifying a Server’s Classes

You can modify the server classes of a server using the MODIFY command. The syntax of the command is:

```
MODIFY id, MODIFY suffix, CLASS=class
```

- **id**: The identifier that you used in the START command to start the queue handler.
- **suffix**: The suffix of the server to be modified.
- **jobname**: The job name of the server. This format must be used if the server was not started via the queue handler.
- **ALL**: Specifies that the server is to serve all classes (0 through 9 and A through Z).
  - **Note**: Do not specify this value if both SNA and OSI servers are used.
- **class**: Specifies the class or classes that the server is to serve. Up to 15 classes can be specified.

**Examples:**

The following command specifies that the server **XY** is to be modified to serve all classes (0 through 9 and A through Z):

```
F QH,F XY,CLASS=ALL
```

The following command specifies that server **YZ** is to be modified to serve classes 1 and 2:

```
F QH,F YZ,CLASS=(1,2)
```

- **Note**: If you modify the class of an OSI server, make sure classes dedicated to perform OSI requests are assigned. The OSI server cannot distinguish between OSI classes and other classes, therefore it accepts any class assigned. Requests obtained by the OSI server are checked that they are OSI requests. SNA requests are rejected.
Modifying the Maximum Number of Simultaneous Sessions That a Server Can Handle

You can tell the queue handler to modify the maximum number of simultaneous sessions that a server can hold, using the MODIFY command. The syntax of the command is:

```
MODIFY id suffix,SRV=(avlsess,avllrh,avlrrh)
```

- **id**  The identifier that you used in the START command to start the queue handler.
- **suffix**  The suffix of the server to be modified.
- **jobname**  The job name of the server. This format must be used if the server has not been started via the queue handler.
- **avlsess**  For SNA file transfers, the maximum number of concurrently active sessions on the specified server. It can be any numerical value from 0 up to the value specified in the server MAXSESS initialization parameter. When 0 is specified, the specified server no longer serves any file-transfer request.

  For OSI file transfers, the maximum number of active transfer requests the OSI server can handle. If the number of available server sessions (that is, FTAM requests) is less than the maximum number of parallel FTAM sessions, the request for a file-transfer session from a remote system is handled at the same time as, and is not blocked by, a locally initiated file transfer.

- **avllrh**  The maximum number of concurrently active local file-transfer requests the specified server can handle. It can be any numerical value from 0 up to the value specified by the AVLSESS parameter.

  When 0 is specified, this server no longer serves any local file-transfer request. When the value is equal to the AVLSESS value, the situation could arise where the server serves only local file-transfer requests and no remote file-transfer requests.

  For an OSI server, this number must be equal to the AVLSESS value because an OSI server can only handle local transfer requests.

- **avlrrh**  The maximum number of concurrently active remote file-transfer requests that the specified server can handle. It can be any numerical value from 0 up to the value specified by the AVLSESS parameter.

  When 0 is specified, this server no longer serves any remote file-transfer request. When the value is equal to the AVLSESS value, the situation could arise that the server serves only remote file-transfer requests and no local file-transfer requests.

  For an OSI server, this number must be set to zero because an OSI server can only handle local transfer requests.
With this MODIFY command the queue handler changes the number of simultaneous sessions that the specified server can handle.

A NetView FTP V2.2.1 MVS server has control statements that the system programmer can use to specify the startup parameters for simultaneous sessions (see also “Specifying Values for the Server Initialization Parameters” on page 113).

These values, except for the value specifying the total maximum number of the servers’ simultaneous sessions can be changed by the operator via this MODIFY command.

The next time the specified server obtains a file-transfer request from the queue handler, the file-transfer request with the highest request class of all server classes this server now serves is selected.

**Example:**

```
  F QH,F XY,SRV=(3,2,2)
```

This command specifies that server XY is to be modified to have a maximum of three concurrently active sessions, and a maximum of either two local sessions or two remote sessions.

---

### Modifying a Request’s Class or Priority

You can use the MODIFY command to change the class or the priority of a specific file-transfer request. The syntax of the command is:

```
MODIFY id,MODIFY REQNUM=n,CLASS=c
```

- `id` The identifier that you used in the START command to start the queue handler.
- `n` The request number of the file-transfer request.
- `c` The new server class to be assigned to the file-transfer request.
- `p` The new request priority to be assigned to the file-transfer request.

This MODIFY command asks the queue handler to change the request class, or the request priority, of the file-transfer request with the request number `n` to `c`, or `n` to `p`. Only the user who submits a file-transfer request, the request queue administrator, and the operator can change the request class of a file-transfer request.
**Note:** The operator can change the class or the priority of a request, even if the pre-queuing user-exit routine prevents the system administrator or the user from doing so.

The modification is accepted if the file-transfer request specified by the request number is still waiting.

---

### Initiating a Request-Queue Rebuild

You can use the MODIFY command to rebuild the request queue.

![COMMAND EXAMPLE]

:id The identifier that you used in the START command to start the queue handler.

:hhh Number of hours that a file-transfer request must have had the status **finished** before it is deleted from the queue. If you omit this value, the value specified in the DELAY startup parameter is used.

Although an automatic request-queue rebuild takes place when the queue is full, the operator can perform a queue rebuild at a noncritical time, for example, in the morning. During a queue rebuild, the queue handler does not allow any file-transfer requests to be added to the queue, nor any file-transfer requests to be scheduled. Note that only successfully finished requests are deleted.

---

### Holding and Releasing Servers

You can use the MODIFY command to hold or release one or all servers.

#### One Server

The syntax of the command is:

![COMMAND EXAMPLE]

:id The identifier that you used in the START command to start the queue handler.

:suffix The suffix of the server to be modified.

:jobname The job name of the server to be modified. This format must be used if the server was not started via the queue handler.

**HOLD**

The queue handler does not schedule any local file-transfer requests for the server. However, the server is still able to serve remote file-transfer requests.
An OSI server can only serve local file-transfer requests. This command allows the OSI server to complete all current requests that are registered in the server and then remain inactive until the server is released by the queue handler.

**RELEASE**
The queue handler releases the specified server. The server starts to process local requests again. Any previously established modifications remain in effect.

### All Servers

The syntax of the command is:

```
MODIFY id HOLD
```

- **id** The identifier that you used in the START command to start the queue handler.

**HOLD**
The queue handler does not schedule any waiting file-transfer requests until the servers are released by the operator. The servers are still able to serve remote file-transfer requests.

OSI servers can only serve local file-transfer requests. This command allows the OSI servers to complete all current requests that are registered in them and then remain inactive until they are released by the queue handler.

**RELEASE**
The queue handler releases all servers. The servers start to process local requests again. All previously established modifications are still in effect.

### Starting and Stopping an Internal Trace

You can use the MODIFY command to start an internal trace in the queue handler. The syntax of the command is:

```
MODIFY id MODIFY TRACE=(GROUPm, CLn)
```

- **id** The identifier that you used in the START command to start the queue handler.

**GROUPm**
The group specifies a set of modules for which one or more trace classes are activated. The trace groups are described in Chapter 12, “Taking Traces” on page 135.

---

1 In previous releases, this was referred to as *trace mask position.*
The class specifies what is to be traced in the selected group of modules. You can activate more than one trace class at a time. The trace classes are described in Chapter 12, “Taking Traces” on page 135.

**OFF**

Switches off all traces that are currently active for the queue handler.

This command modifies the queue handler to start an internal trace, to change the trace class or classes for the currently running internal trace, or to deactivate all queue handler traces.

**Examples:**

MODIFY QH,F TRACE=(GROUP3,CL1,CL2)

Invokes an internal trace with the trace classes entry/exit (CL1) and request control block and (if available) extensions (CL2) in the module group that handles queue-handler commands issued by interface modules.

F QH,MODIFY TRACE=OFF

Stops the internal trace in the queue handler.

---

**Stopping a Server**

You can stop any server using the NetView FTP V2.2.1 MVS operator interface, regardless of the way in which it was started. For instance, you can use the operator interface to stop a server that was started by submitting a batch job.

**Note:** If possible, do not use the CANCEL command to stop a server. If the server does not terminate after you have tried the normal methods, you can deactivate its LU by entering the ACF/VTAM VARY command:

```bash
VARY NET,INACT,ID=luname,IMMED
```

If this has no effect, use the CANCEL command.

The following command causes the queue handler to stop either a single server, or all servers:

```bash
MODIFY id,STOP suffix ,P ALL ,IMMED
```

**id** The identifier that you used in the START command to start the queue handler.

**m** One of the following:

- A 2-digit number (0 through 9 or 00 through 99) that is the suffix of the name of the cataloged procedure of a server
• A name of up to eight characters that is the complete name of the cataloged procedure for the server. If a server is started using this method, it must be stopped using the same name.

**ALL**

Stops all running servers and switches the queue handler into a state of delayed termination. It does not allow:

- A batch job to start a new server
- The request queue to schedule a file-transfer request.

You can, however, start a new server via the operator interface routine. This resets the state of delayed termination.

**IMMED**

Specifies an immediate termination of the server.

If you specify **IMMED** for an SNA server, all file transfers that are currently being processed are interrupted, checkpoints are taken, and the server stops immediately. If the corresponding file-transfer requests specify automatic transfer restart, they are rescheduled when a server that serves their request class becomes available.

If you do not specify **IMMED**, the server first finishes all file transfers that are currently being processed, before it stops. It does not start any new file transfers, nor does it accept any request from a remote server to start a session.

If you specify **IMMED** for an OSI server, all file transfers that are currently being processed are canceled, and the requests are returned to the queue handler with the status *waiting*. When a request is already completed and the server is posted for immediate termination, the request is returned to the queue handler with the appropriate status. The recovery file will be empty after successful server termination unless an error occurs while terminating.

If you do not specify **IMMED**, the OSI server waits for all requests to finish before it stops. It does not obtain any new requests from the queue handler nor does it pass a request to OSI/FS.

**Examples:**

```
F QH,STOP 03
Orders the delayed termination of the server FTPSRV03. FTPSRV is the name of the SRVPREF initialization parameter of the queue handler.

F QH,P SERVST,I
Orders the immediate termination of the server SERVST. SERVST can either be a job name or the name of the cataloged procedure with which you started the server.
```

**Examples:**

```
F QH,STOP ALL
Orders the delayed termination of all running servers.

F QH,P ALL,I
Orders the immediate termination of all running servers.
```
Stopping the Queue Handler

You stop the queue handler by entering the STOP command.

The syntax of the STOP command is:

```
STOP id
```

The `id` is the identifier that you used in the START command to start the queue handler.

**Example:** To stop the queue handler that was started with the procedure DVGQHPRO or with the `id` QH, enter:

```
P DVGQHPRO
```

or

```
P QH
```

Do not use the CANCEL system command to stop the queue handler unless the queue handler has not terminated several minutes after you entered the STOP command, and is apparently in a wait state. When you stop the queue handler, all running servers are stopped immediately. The file transfers that are currently being processed are interrupted and checkpoints are taken. When you restart the queue handler, any file-transfer request that specified automatic transfer restart is rescheduled when a server that serves its request class becomes available.

The termination of the queue handler can take several minutes, depending on the system’s performance. If, after a defined time interval, not all servers could be stopped, the queue handler terminates. The success of stopping the servers is indicated by a message. If not all servers are stopped, cancel the servers that are still running.
This part describes how to:

- Set up and maintain the request queue
- Set up and maintain the checkpoint data set
- Set up and maintain ACF/VTAM resources used by NetView FTP V2.2.1 MVS
- Set up a server group table
- Administer OSI file transfers
- Write procedures for starting the queue handler and server
- Change the text of NetView FTP V2 MVS messages.
Chapter 4. Setting Up and Maintaining the Request Queue

The request queue is a VSAM key sequenced data set (KSDS) that must be set up before you run NetView FTP V2.2.1 MVS. This chapter describes how to do this and how to maintain the request queue.

Setting Up the Request Queue

The request queue can have up to 99,999 records, and each record can contain one file-transfer request. NetView FTP V2.2.1 MVS identifies each request by the request number.

Setting up the request queue involves defining and initializing the request queue.

Defining and Initializing the Request Queue

Member DVGMXRQ1 in library DVG.NFTP230.INSTLIB on the NetView FTP V2.2.1 MVS distribution tape contains a batch job that defines a request queue, uses the utility IEBDG to create a sequential data set containing initial values for the request queue, and uses AMS to load the sequential data set into the request queue.

This batch job is shown in Figure 41.

If you require a larger request queue than that provided by job DVGMXRQ1, you can modify the job accordingly.

Note: The values in bold are generated by the job generator. You can modify them to suit your installation. You must change the values coded in lowercase to meet the needs of your installation.
Figure 41. Sample Job to Define the Request Queue (DVGMXRQ1)
The values of the following SYSIN parameters can be changed in or added to the job to suit the needs of your installation:

- BUFFERSPACE
- FREESPACE
- RECORDSIZE
- CONTROLINTERVALSIZE.

How to specify the request queue parameters is described in “Specifying Request Queue Parameters” on page 199.

CLUSTER

The cluster name shown (DVG.NFTP230.REQUEST.QUEUE) is a sample only.

VOLUMES

If you are not using SMS, include the VOLUMES parameter.

DATA

The data name shown (DVG.NFTP230.REQUEST.QUEUE.DATA) is a sample only.

INDEX

The index name shown (DVG.NFTP230.REQUEST.QUEUE.INDEX) is a sample only.

The SEQOUT DD statement defines the sequential data set that IEBDG initializes. The data-set name shown is only a sample.

Change the UNIT parameter to correspond to a volume on your system.

The value for the RECORD parameter depends on the number of records you want to generate.

The two CREATE statements are needed to generate the records. The first statement generates the record with key 0. This record is used for NetView FTP V2.2.1 MVS internal purposes and must always be generated. The second statement generates the records that can later contain file-transfer requests. You must generate these records with ascending keys starting with 1.

The value of the QUANTITY parameter in the second CREATE statement defines the maximum number of requests that the request queue can hold. This value is one less than the total number of records in the request queue (record 0, is used by NetView FTP V2.2.1 MVS for internal purposes). You can also change the values of the following, to suit the needs of your installation:

RQDATA

The sequential data set that is used to load the request queue.

You must use the same data-set name as in the previous SEQOUT DD statement.
Avoiding Request Queue Overflows

Rebuilding the request queue usually eliminates an overflow. However, because rebuiliding the queue can often take several minutes and during this time users cannot use NetView FTP V2.2.1 MVS, it can severely affect the availability of NetView FTP V2.2.1 MVS.

Use the following formula to calculate a size for the request queue so that overflows do not occur:

\[ M \geq (A \times N) + (B \times N) + C \]

where:

- **M** is the size, in number of records, of the request queue.
- **A** is the number of days that the queue handler is to run continuously.
- **B** is the number of days that successfully finished file-transfer requests are to be left in the request queue, so that users can query them.
- **N** is the average number of file-transfer requests that are added to the request queue each day. The formula assumes that the same number of file-transfer requests reach finished status each day.
- **C** is the number of file-transfer requests that you want to reserve as a buffer on the request queue, to allow for special situations.

For example, imagine the following:

- You want to stop and restart the queue handler once every week \((A=7)\).
- You want to leave successfully finished requests in the queue for at least two weeks \((B=14)\).
- An average of 100 file-transfer requests per day are added to the request queue, and the same number reach finished status each day \((N=100)\).
- You want to have as a buffer the number of requests added to the queue in five working days \((C=5 \times N=500)\).

With this formula you define a request queue with the following number of records:

\[ M = (7 \times 100) + (14 \times 100) + 500 = 2600 \]

If you define the buffer large enough to cover all unusual situations, the request queue does not overflow. You can use this result to specify QUANTITY in the second CREATE statement of the request queue initialization job.
What to Do when the Request Queue Overflows

The request queue “overflows” when it receives more requests than it can contain. When this happens the queue handler rejects all attempts to add more file-transfer requests to the request queue, and initiates an automatic rebuild of the request queue.

During a queue rebuild, the queue handler deletes all requests that have finished successfully and have had the status finished for longer than the time specified in the DELAY initialization parameter for the queue handler. During the time the queue handler rebuilds the request queue it does not accept any commands that access the request queue.

Automatic rebuilding can fail if there are no successfully finished file-transfer requests for which the delay time has elapsed. To remedy this, initiate a request queue rebuild with a shorter delay time. How to do this is explained in “Initiating a Request-Queue Rebuild” on page 59. You can then ask your users to delete any file-transfer requests they no longer need, for example:

- Specific file-transfer requests
- All their successfully finished file-transfer requests
- All their file-transfer requests.

Besides that you can ask your users to restart all file-transfer requests that are manually restartable.

If you know the NetView FTP V2.2.1 MVS master password you can do this for other users.

If your request queue overflows frequently, you can increase its size. To do this:

1. Stop NetView FTP V2.2.1 MVS. How to do this is described in “Stopping the Queue Handler” on page 63.
2. Unload the request queue using IDCAMS with the REPRO command.
3. Modify the job used to define the request queue. Increase the value of the RECORD parameter.
   Modify the QUANTITY of the second CREATE statement in the request queue initialization job. How to do this is described in “Defining and Initializing the Request Queue” on page 67.
4. Run the request queue initialization job.
5. Reload the request queue using IDCAMS with the REPRO command.
6. Restart NetView FTP V2.2.1 MVS. How to do this is described in Chapter 10, “Writing Startup Procedures for NetView FTP V2.2.1 MVS” on page 95.

To enable automated operations on the request queue data set, the I/O error messages DVG043I, DVG045I, and DVG670I are written to the operator console.
Printing the NetView FTP V2.2.1 MVS Request Queue

Member DVGPRQU in library DVG.NFTP230.INSTLIB, generated by installing NetView FTP V2.2.1 MVS, contains a batch job that you can use to print the contents of the request queue. You must adapt the job to meet the needs of your installation.
Chapter 5. Setting Up and Maintaining the Checkpoint Data Set

This chapter describes how to set up and maintain the checkpoint data set for SNA-type transfers. The checkpoint data facility does not apply for OSI-type transfers.

During a file transfer the receiving server periodically writes a checkpoint record into the checkpoint data set. The checkpoint record contains all the information that NetView FTP V2.2.1 MVS needs to restart the file transfer. If a file transfer is interrupted NetView FTP can use this information to restart the file transfer from the last checkpoint rather than from the beginning of the file.

All servers at one NetView FTP V2.2.1 MVS installation can share the same checkpoint data set. This allows a server other than the one that began a file transfer to restart an interrupted file transfer. You can also define a checkpoint data set for each server in your installation. If you do this, however, an interrupted file transfer can only be restarted by the same server. The request originator must ensure this by specifying that the request can be processed only by a specific server. How to do this is described in the NetView FTP Parameter Reference.

Setting Up the Checkpoint Data Set

Setting up the checkpoint data set involves doing the following:

- Deleting the old checkpoint data set if necessary
- Defining a new checkpoint data set
- Initializing the checkpoint data set.

Member DVGMXCR1 in library DVG.NFTP230.INSTLIB on the NetView FTP V2.2.1 MVS distribution tape contains a batch job that deletes the old checkpoint data set and defines and initializes a new checkpoint data set. The batch job uses the utility IEBDG to create a sequential file containing initial values for the checkpoint data set and uses AMS to load the sequential file into the checkpoint data set.

This batch job is shown in Figure 42.

Member DVGMCXRD in library DVG.NFTP230.INSTLIB on the NetView FTP V2.2.1 MVS distribution tape contains the IDCAMS parameters for the job DVGMXCR1. Member DVGMCXRD is shown in Figure 43 on page 75.

Note: The values in bold are generated by the installation generator. You can modify them to suit your system. You must insert values for all lowercase variables. All other parameters must be specified as shown in the sample.
Figure 42. Sample Job to Set Up the Checkpoint Data Set (DVGMXCR1)
Figure 43 shows the IDCAMS parameter input for job DVGMXCR1.

```
/*-----------------------------------------------*/
/* NETVIEW FILE TRANSFER PROGRAM VERSION 2.2.1 */
/* GENERATED BY MACRO DVGMXJG2 */
/* MEMBER:    DVGMXRD */
/* PARAMETERS USED: */
/* VSQUAL   DVG.NFTP23 */
/* --------- ONLY IF SMS=NO: */
/* VSOL    US9000 */
/* FUNCTION: */
/* IDCAM PARAMETERS: DELETE, DEFINE AND */
/* INITIALIZE OF THE NFTP CHECKPOINT DATA SET */
/*-----------------------------------------------*/
/*-----------------------------------------------*/
/* CHECK FOR EXISTENCE OF THE VSAM DATA SET */
/*-----------------------------------------------*/
LISTCAT ENTRY (DVG.NFTP23.CHECKPNT.FILE)
   OUTFILE (VSAMCHCK)
IF LASTCC=/zerodot THEN DO
   SET MAXCC=/zerodot
END
ELSE DO
/*-----------------------------------------------*/
/* DELETE THE VSAM CLUSTER */
/*-----------------------------------------------*/
DELETE (DVG.NFTP23.CHECKPNT.FILE)
SET MAXCC=/zerodot
END
/*-----------------------------------------------*/
/* DEFINE THE VSAM CLUSTER */
/*-----------------------------------------------*/
DEFINE CLUSTER( -
   NAME(DVG.NFTP23.CHECKPNT.FILE) -
   RECORDSIZE(992 992) -
   REC(128 128) -
   FREESPACE(20 30) -
   INDEXED -
   KEYS(254 /zerodot) -
   SHR(4) -
   WRITECHECK -
   DATA( -
      NAME(DVG.NFTP23.CHECKPNT.FILE.DATA) -
      CONTROLINTERVALSIZE(4096)) -
   INDEX( -
      NAME(DVG.NFTP23.CHECKPNT.FILE.INDEX) -
      CONTROLINTERVALSIZE(4096)) )
/*-----------------------------------------------*/
/* INITIALIZE THE VSAM CLUSTER */
/*-----------------------------------------------*/
REPRO -
   INFILE(CRRDATA) -
   ODS(DVG.NFTP23.CHECKPNT.FILE)
/*-----------------------------------------------*/
```

Figure 43. IDCAMS Parameters for Job DVGMXCR1 (DVGMXRD)

You can also change or add the following SYSIN parameters to suit the needs of your installation:

- BUFFERSPACE
- FREESPACEx
- CONTROLINTERVALSIZE.

How to specify the checkpoint data set parameters is described in “Specifying Checkpoint Data-Set Parameters” on page 199.
NAME
The cluster name shown is a sample only.

VOLUMES
If you are not using SMS, include the VOLUMES parameter.

SHAREOPTIONS
To allow servers to share the same checkpoint data set, each server startup job (procedure) must specify DISP=SHR in the checkpoint data set's DD statement.

The servers serialize access to the checkpoint data set by using ENQ/DEQ.

The SEQOUT DD statement defines the sequential file that IEBDG initializes with the initial checkpoint record.

CRRDATA Defines the sequential file that is used in this step to load the checkpoint data set. It must be the same name you specified in the SEQOUT DD statement.

What to Do when the Checkpoint Data Set Becomes Full
NetView FTP V2.2.1 MVS automatically deletes each checkpoint record that corresponds to a successful file transfer. However, if many file transfers are interrupted and not restarted, the checkpoint data set can become full. When the checkpoint data set is full, only receiving servers conducting file transfers for which there is already a checkpoint record in the checkpoint data set can take checkpoints—other receiving servers cannot.

The file-transfer reports that the other receiving servers create each contain a message indicating that the checkpoint data set is full. These messages are also sent to the operator console to enable automated operations to commence. Refer to message DVG670I for information about CLISTs.

If NetView FTP V2.2.1 MVS detects an overflow during a file transfer, it does not interrupt the file transfer, but continues the transfer. However, no additional recovery information can be stored. In the case of an interrupt, NetView FTP V2.2.1 MVS restarts the file transfer from an earlier checkpoint or from the beginning of the file. The NetView FTP V2.2.1 MVS report files created for each file transfer contain an I/O error message for the NetView FTP V2.2.1 MVS checkpoint data set, indicating the overflow condition.

When all transfers that are currently being processed have finished, the server that detected the overflow condition automatically terminates. If all of your servers share the same checkpoint data set you should stop NetView FTP V2.2.1 MVS without interrupting the file transfers that are currently being processed. Otherwise you should stop those servers that access the checkpoint data set in error.
You can do any of the following:

1. Set up a new and larger NetView FTP V2.2.1 MVS checkpoint data set.

2. Use AMS to unload the old NetView FTP V2.2.1 MVS checkpoint data set. Delete the old checkpoint data set. Allocate a larger NetView FTP V2.2.1 MVS checkpoint data set. Reload the checkpoint data set.

   Member DVGMXCR2 in library **DVG.NFTP230.INSTLIB** contains a sample batch job for this. This member is shown in Figure 44.

3. Rebuild the NetView FTP V2.2.1 MVS checkpoint data set.

You can now restart NetView FTP V2.2.1 MVS or the stopped servers.

```plaintext
//DVGMXCR2 JOB (accn,n),'programmer-id',
// MSGCLASS=x,MSGLEVEL=(1,1),CLASS=a,
// NOTIFY=userid,USER=userid
 /***********************************************************************
 // NETVIEW FILE TRANSFER PROGRAM VERSION 2.2.1
 // Generated by macro DVGMXJG2
 // MEMBER : DVGMXCR2
 // Parameter:
 // INDEX DVG.NFTP230
 // VSQUAL DVG.NFTP230
 // WRKDISK SYSDA
 // FUNCTION :
 // REORGANIZE THE NFTP CHECKPOINT DATA SET
 /***********************************************************************
 // UNLOAD THE EXISTING CHECKPOINT DATA SET
 // UNLOAD EXEC PGM=IDCAMS,REGION=2M
 // SYSPRINT DD SYSOUT=* // CRRDATA DD DSN=&TEMP,UNIT=SYSDA,
 // DISP=(NEW,PASS),SPACE=(TRK,(15,15)), // DCB=(RECFM=FB,LRECL=992,BLKSIZE=1984)
 // SYSIN DD +

 // UNLOAD THE VSAM CLUSTER
 REPRO -
   INFILE(CRRDATA) -
   ODS(DVG.NFTP230.CHECKPNT.FILE)

 /***********************************************************************
 // DELETE AND DEFINE AND INIT THE CHECKPNT DATA SET
 /***********************************************************************
 //DEFREP EXEC PGM=IDCAMS,REGION=2M
 //SYSPRINT DD SYSOUT=* //VSAMCHCK DD DUMMY,
 // DCB=(RECFM=VBA,LRECL=125,BLKSIZE=629)
 //CRRDATA DD DSN=&TEMP,DISP=(OLD,DELETE)
 //SYSIN DD DSN=DVG.NFTP230.INSTLIB(DVGMXCRD),DISP=SHR
```

Figure 44. Sample Job DVGMXCR2 to Reorganize the Checkpoint Data Set
Avoiding a Checkpoint Data Set Overflow

To avoid an overflow you should delete and re-create the checkpoint data set at regular intervals. This is called rebuilding the checkpoint data set.

Before doing this, however, you should list the contents of the checkpoint data set, to ensure that there are no checkpoint records belonging to file transfers that must be restarted.

Printing the NetView FTP V2.2.1 MVS Checkpoint Data Set

Member DVGPCRD in library DVG.NFTP230.INSTLIB, generated by installing NetView FTP V2.2.1 MVS, contains a batch job that you can use to print the contents of the checkpoint data set. You must adapt the job to meet the needs of your installation.
Chapter 6. Setting Up and Maintaining the ACF/VTAM Resources Used by NetView FTP V2.2.1 MVS

This chapter describes how to set up and maintain the ACF/VTAM resources used by NetView FTP V2.2.1 MVS.

Before using NetView FTP V2.2.1 MVS, you must:

- Define each local server as an ACF/VTAM application program system.
- Create one or more logon mode table entries for your servers.

In addition, you are recommended to define each remote server with which you want to exchange data as a cross-domain resource.

When NetView FTP V2.2.1 MVS communicates with its partner file-transfer programs it uses the LU type 6.2 (APPC) protocol.

Defining an SNA Server as an ACF/VTAM Application Program Node

You must define each server at your location\(^2\) to ACF/VTAM as part of a logical set of one or more ACF/VTAM application programs (APPLs). This logical set is called an application program major node. An application program major node is defined by coding and filing one VBUILD statement for the major node, and one APPL definition statement for the network name of each server known to ACF/VTAM at each location.

For further information refer to VTAM Installation and Resource Definition Version 3.

**Note:** As with any APPL definition, you can define NetView FTP V2.2.1 MVS as one or more separate major nodes, or you can add NetView FTP V2.2.1 MVS to one or more existing application program major nodes. Figure 45 shows sample definitions for two servers in an application program major node.

![Figure 45. Defining Servers as an Application Program Major Node](image-url)

\(^2\) In this book, the term "location" is used as a synonym for "domain," "host node," and "host processor" as used in other SNA-related publications.
**APPL name and ACBNAME**

Use these parameters to define the minor node names for your servers.

In this example, the names of the APPL statements for the server are NFTP001 and NFTP002. The names must be unique in the domain in which the server resides, and within the network if the server is to engage in cross-domain sessions.

If you do not code ACBNAME, the name of the APPL statement must be the same as the name in the APPLID operand of the ACB macroinstruction for the server. The server uses the value of the LUNAME initialization parameter as the APPLID operand.

If you code ACBNAME, it must be unique within the domain in which the server resides. The name must also be the same as the value of the LUNAME initialization parameter for the server. If ACBNAME is not coded, the name of the APPL statement is used for the ACBNAME.

**AUTH**

The value ACQ must be specified for the AUTH operand. This gives the server the authority to use the OPNDST macroinstruction with the ACQUIRE option. For further information refer to *VTAM: Programming*.

Specifying the value PASS is optional.

**VPACING**

Specifying a value for the VPACING is optional, but strongly recommended. The value you specify works with values in the BIND image. For more information about pacing, see “Pacing” on page 213.

**APPC**

Because the server uses an LU 6.2 protocol, the value of the APPC operand must be YES.

**DLOGMOD**

The value specified in the DLOGMOD server initialization parameter is used, see “Mode Name” on page 117.

**MODETAB**

The value of the MODETAB operand specifies the name of the ACF/VTAM logon mode table in which an entry for the server has been made.

**PARSESS**

Because each server uses parallel sessions for transferring data, the value of the PARSESS operand must be YES.

**AUTOSES**

It is recommended to set this value to 0.

**DSESLIM**

This value defines the maximum number of sessions to be allowed between the local application LU and a remote LU on a given node. For more details on VTAM session limits see the *VTAM Installation and Resource Definition Version 3* manual.
Defining an SNA Server as a Cross-Domain Resource

You are recommended to define the remote servers that conduct file transfers with the servers at your location as application program cross-domain resources (CDRSCs). You do this by coding and filing one VBUILD statement for the cross-domain resource major node and one CDRSC statement for each network name that a remote server uses when communicating with its session partner.

For more information about defining cross-domain resources, refer to VTAM Installation and Resource Definition Version 3.

Figure 46 shows sample definitions of how cross-domain resources could be defined. In this example, CDRM5 is the name of the Cross-Domain Resource Manager (CDRM) that controls the CDRSC.

<table>
<thead>
<tr>
<th>VBUILD TYPE=CDRSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFTP003 CDRSC CDRM=CDRM5</td>
</tr>
<tr>
<td>NFTP004 CDRSC CDRM=CDRM5</td>
</tr>
</tbody>
</table>

Figure 46. Defining Servers as a Cross-Domain Resource

Example of Combined APPL and CDRSC Definitions for One Location

Assume the following:

- Your location has two servers with the LU names NFTP001 and NFTP002.
- Your location is to communicate with two other locations.
  - One location has a cross-domain resource manager called CDRM12, and one server with the LU name FTP0002.
  - The other location has a cross-domain resource manager called CDRM99 and three servers with the LU names FTP000X, FTP000Y, and FTP000Z.

SECACPT

This value must be set to CONV.

All other operands that can be coded in APPL definitions are optional; they depend on the requirements of your location.

Note: If the PRTCT operand is used to assign a password, this password must also be coded when specifying a value for the LUNAME server initialization parameter. The LUNAME server initialization parameter is described in “Specifying Values for the Server Initialization Parameters” on page 113.
An example of what the entries can look like is shown in Figure 47.

```
VBUILD TYPE=APPL
NFTP001 APPL AUTH=(ACQ,PASS),VPACING=16,ACBNAME=NFTP001,APPC=YES, X
   DLOGMOD=FTPBIND,MODETAB=FTPFTF,PARSESS=YES, X
   AUTOSES=0,DSESLIM=128,SECACPT=CONV
NFTP002 APPL AUTH=(ACQ,PASS),VPACING=16,ACBNAME=NFTP002,APPC=YES, X
   DLOGMOD=FTPBIND,MODETAB=FTPFTF,PARSESS=YES, X
   AUTOSES=0,DSESLIM=128,SECACPT=CONV,

VBUILD TYPE=CDRSC
FTP0002 CDRSC CDRM=CDRM12
FTP000X CDRSC CDRM=CDRM99
FTP000Y CDRSC CDRM=CDRM99
FTP000Z CDRSC CDRM=CDRM99
```

Figure 47. Example of Combined APPL and CDRSC Definitions

Defining an OSI Server as an ACF/VTAM Application Program Node

Defining an OSI server as an ACF/VTAM application program node is described in the *OSI/File Services Installation and Customization Guide MVS*.

Creating a Logon Mode Table

There must be a logon mode table entry for NetView FTP V2.2.1 MVS. This entry must be defined in the table specified by the MODETAB keyword in the ACF/VTAM APPL definition, or must be in the ACF/VTAM default logon mode table. The name must not be SNASVCMG because this name is reserved. However, as NetView FTP uses LU 6.2 conventions, an entry with the name SNASVCMG must exist in the ACF/VTAM logon mode table. For different servers, the entries can be in different tables. A sample logon mode table with the name FTPFTF is shown in Figure 48.

**Note:** The values shown may be overwritten by VTAM during conversation establishment.

```
FTPFTF MODETAB
FTPBIND MODEENT LOGMODE=FTPBIND,
   TYPE=X'00',
   FMPROF=X'13',
   TSPROF=X'07',
   PRIPROT=X'80',
   SECPROT=X'80',
   COMPROT=X'00B1',
   PSNDPAC=X'00',
   SRCVPAC=X'00',
   SSNDPAC=X'01',
   RUSIZES=X'8585',
   PSERVIC=X'060200000000000000000000000300'
   MODEEND
END
```

Figure 48. Example of an Entry in a Logon Mode Table
When you have created and named a logon mode table you must assemble and link-edit it into a library that is available to VTAM. How to do this is described in *VTAM: Customizing*.

Most of the parameters that NetView FTP uses for an LU 6.2 type conversation are set by VTAM. They override the values in your logon mode table entry. For more information about conversation parameters refer to *VTAM Programming for LU 6.2*.

The following parameters are used the way they are coded in your logon mode table:

- LOGMODE
- RUSIZES
- PSNDPAC
- SRCVPAC
- SSNDPAC.

A logmode name can be specified as the server initialization parameter DLOGMOD. If a NetView FTP server requests an LU 6.2 type conversation, this logmode name is sent to the node of its conversation partner. At this node the logmode name is translated into conversation parameters and a class-of-service name. Ensure that an entry with this logmode name is included in the logon mode table that is specified in the VTAM APPL statement of the partner LU.

The RU sizes used for an LU 6.2 type conversation are always the values in the logon mode table entry.

How to specify conversation pacing values is described in “Pacing” on page 213.

### VTAM Start Options

VTAM non-native network connection enables NetView FTP MVS to establish communications with NetView FTP/400 residing in an adjacent network.

The non-native network connection is enabled by specifying the VTAM Start option: XNETALS=YES

This function is available in the following VTAM releases:

- VTAM V3 R3 for MVS/XA, APAR OY30488
- VTAM V3 R3 for MVS/ESA, APAR OY30487
- NCP V5 R3, APAR IR88927
- SSP V3 R5 for MVS, APAR IR88946
- SSP V3 R5 for VM, APAR IR88978
- SSP V3 R5 for VSE, APAR IR88979.
Chapter 7. Setting Up a Server Group Table

This chapter describes how to set up a server group table. The server group table only contains information about SNA servers. It contains no information about OSI servers.

A server group table is a data set that assigns each of the servers in your network to a group and contains information about each server. Using a server group table has the following advantages:

- It gives local servers a wider range of potential session partners. The local server that processes a request can establish a session with any of the servers in the group. If one server in the group is busy or not yet started, the local server tries the next server in the group until it succeeds in establishing a session or has tried all the servers in the group. Without a server group table, users must specify the LU name of a single remote server. If this server is not available, the file transfer cannot take place.

Note, however, that server groups are not supported for NetView FTP/2 and NetView FTP AIX. An entry for NetView FTP/2 or NetView FTP AIX represents a NetView FTP Server for OS/2 or NetView FTP Server AIX, respectively.

- You can make the names of the server groups easier to remember than LU names, which often must follow technical naming conventions.

Make sure that the server group table specifications are correct to avoid delays when the transfer starts.

Creating a Server Group Table

The server group table is a sequential data set with the ddname DVGNDT. To use the server group table, specify this ddname in the startup job of the queue handler. This startup job is described in Chapter 10, “Writing Startup Procedures for NetView FTP V2.2.1 MVS” on page 95. The queue handler reads the server group table into its address space. Any changes made to the server group table after the queue handler has started do not become effective until the queue handler has been stopped and restarted.

Member DVGSVGRP in DVG.NFTP230.INSTLIB is a sample server group table that you can inspect and modify for your own use.

Each record of the server group table corresponds to one server. Each record must have the following format:

<table>
<thead>
<tr>
<th>Server Group Name</th>
<th>Operat. sys.</th>
<th>LU name</th>
<th>FTP level</th>
<th>R M</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>1 1 1 1 1 1 1</td>
<td>2 2 2 2 2 2 2 2 2</td>
<td>3 3 3 3 3 3 3 3 3</td>
<td>3 3 3 3 3 3 3 3 3</td>
<td>Description</td>
</tr>
</tbody>
</table>

Figure 49. Layout of the NetView FTP V2.2.1 MVS Server Group Table
The records must be at least 28 bytes long. The records are sorted internally into ascending order, the first 28 bytes are significant. NetView FTP reads only the first 28 bytes of each record; the remaining bytes can be used for comments. An entire record can be used for comments if column 1 contains an asterisk (*). Each value must begin in the first byte of the corresponding field. If a value is shorter than its field, it must be padded with trailing blanks.

Server Group Name
Bytes 1 to 8 of a server group table record are reserved for the names of the server groups in your network. All records that begin with the same server group name define a server group and the LU names of its servers.

Operating System
This field determines the operating system under which all servers of a server group run. The following values can be specified:

- MVS
- OS2
- OS4
- AIX
- VM
- VSE
- Blank.

Each server group can have only one operating system. All servers in one server group must have the same operating system.

If you specify an operating system, the user interfaces can check a file-transfer request that specifies that server group or server for operating system-dependent errors. If you leave this field blank, the user interfaces do not perform this check.

LU name
This field defines the ACF/VTAM LU name of one server belonging to the server group. It can be the LU name of a NetView FTP server.

If the server group contains more than one server, specify each of the servers, using different records in the server group table, so that the server group names and the operating systems for each of these records are the same.

FTP level
This defines the level of a remote transfer program.

It is no longer used.

Running Mode
Defines the server running mode on the remote system associated with the LU name.

The following running modes can be specified:

- S Single
- C Continuous
- G Continuous, update of GDG table. For more information see page 116.

If the running mode is not known, the column can be left blank.
Note: It is recommended that you enter in the server group table all potential remote systems to or from which files can be transferred. The local system can also be defined in the server group table.

A sample server group table is shown in Figure 50.

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWYORK1 VSE NYFTPE1</td>
<td>Small files transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWYORK1 VSE NYFTPE2</td>
<td>Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWYORK1 VSE NYFTPE5</td>
<td>S Single mode Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* New York Sales System

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWYORK2 MVS NYFTPM1</td>
<td>Tape data set transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWYORK2 MVS NYFTPM2</td>
<td>Small data set transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWYORK2 MVS NYFTPM3</td>
<td>Small data set transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWYORK2 MVS NYFTPM4</td>
<td>Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWYORK2 MVS NYFTPM5</td>
<td>G GDG data sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* New York Research System

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS MVS EHNFTPM1</td>
<td>C Tape data set transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARIS MVS EHNFTPM2</td>
<td>C Small data set transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARIS MVS EHNFTPM3</td>
<td>C Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Paris System

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAIROBI VM NBRFTPV1</td>
<td>Tape file transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAIROBI VM NBRFTPV2</td>
<td>Small file transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAIROBI VM NBRFTPV3</td>
<td>Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Nairobi System

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LONDON1 OS4 LONTPE1</td>
<td>Small files transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LONDON1 OS4 LONTPE2</td>
<td>Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* London Sales System

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROME OS2 ROMESV1</td>
<td>Small files transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROME OS2 ROMESV2</td>
<td>Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Rome System

<table>
<thead>
<tr>
<th>* SGN</th>
<th>OPS</th>
<th>LUN</th>
<th>FL</th>
<th>R</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIENNA AIX VIENNAV1</td>
<td>Small files transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIENNA AIX VIENNAV2</td>
<td>Long running transfers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Vienna System

Figure 50. Sample Server Group Table

Considerations for Defining Server Groups

When defining server groups, take the following into consideration:

- Only your local NetView FTP V2.2.1 MVS installation uses the server groups you specify. You can therefore choose any name. Even if the same server group is defined at two different locations, the server groups can be different.

- NetView FTP does not impose any restrictions about which server LU name belongs to which server group.

- You can group servers into nodes according to your requirements. You are also independent from the server group table definitions on other systems in your network.
- A server group does not have to comprise all servers at a system, one server might be sufficient.
- You can group the servers at one system into several server groups.
- You can group servers at different systems into one server group.
- If a server group is defined for transfer programs running on a workstation, the server group must contain only the LU name of the NetView FTP Server workstation.
- A server LU name can be part of more than one server group.
- It is recommended that you sort your server group table in alphabetic order before you start the queue handler. This reduces the amount of time the queue handler needs for its initialization.

### Updating a Server Group Table

If the data set that contains the server group table is accessed exclusively by the queue handler you cannot update the server group table. If the data set is accessed as shared you can update the server group table.

**Note:** Changes that you make to the server group table take effect only when you restart the queue handler.
Chapter 8. SMF Recording

You can specify the SMF record number that NetView FTP V2.2.1 MVS is to use via the queue handler and server initialization parameter SMFREC. Each NetView FTP V2.2.1 MVS action creates a subtype of this SMF record type. A file-transfer request specified by a request number can be tracked from the time the user enters the file-transfer request until the time the file transfer finishes.

A NetView FTP V2.2.1 MVS SMF record consists of the following section types:

- The SMF header section
- The SMF product section
- The SMF data section.

For more information see Appendix H, “Layout of the SMF Records” on page 223.

If the SMF record type is defined in the SMFPRMxx member of the SYS1.PARMLIB data set, the queue handler (respectively the server) creates the SMF buffer and writes the record using the SMFWTM macro. NetView FTP V2.2.1 MVS provides an Assembler macro to map the SMF records.

A data reduction or evaluation program, such as SLR Version 3 Release 2 helps to make efficient use of the SMF records. Some of the key values, such as request number, job name, and request originator, are contained in several data sections. In order to get meaningful data over the lifetime of a request, you should combine all data sections that contain the relevant key value or values.

The queue handler writes SMF records for all NetView FTP activities initiated at the requesting system. The server writes SMF records only for remotely initiated activities.
Chapter 9. Administering OSI File Transfers

There are two ways to allow users to perform OSI file transfers in NetView FTP V2.2.1 MVS:

1. Prevent users from accessing OSI/File Services. This means, all OSI file transfers must go through NetView FTP V2.2.1 MVS, and all the security features of NetView FTP V2.2.1 MVS are used.

2. Allow users access to OSI/File Services. This means, users can make entries in the Local Resource Directory (LRD) themselves and also transfer files using OSI/File Services. This has the disadvantage that the security features provided by NetView FTP V2.2.1 MVS can be bypassed. For more information, see “Data Security for OSI File Transfers” on page 183.

The following diagram shows the layout of the local resource directory of OSI/File Services:

![Local Resource Directory Diagram](image)

**Figure 51. Layout of the Local Resource Directory**

**Local Resource Directory (LRD)**

The file containing information on local users, local filestores, filestore subsets, and local files necessary for OSI/File Services to perform file transfers.

**Filestore Owner**

The 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.

**Local File Names**

The MVS file name without the first qualifier.
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.

Passwords

There are passwords to access a filestore, to create a new file in an existing filestore, and to perform certain functions on a particular file.

OSI file transfers can be performed either automatically or manually:

Automatically

The OSI server that is to process the file transfer automatically adds the name of the file to be transferred to its own filestore, and automatically deletes the name of the file once it has been transferred.

An automatic file transfer can take place if:

- The first qualifier of the MVS file name (filestore subset), usually the TSO user ID of the user, is in the filestore of the OSI server that is to process the request. This can be added using the interactive interface of the OSI/File Services.
- The filestore subset does not exist in any other filestore. Filestore subsets cannot be included into a user’s filestore if they already exist in another filestore.

For example, in Figure 51 an MVS file called BMM.filename5 could be transferred automatically. However, an MVS file called ABC.filename5 would have to be transferred manually, since it already exists in another filestore, and its filestore subset does not already exist in the filestore of the OSI server.

The NetView FTP V2.2.1 MVS OSI server acts as a normal user to OSI/File Services. OSI/File Services accepts requests from the OSI server, and other OSI/File Services users, for file transfers, administers those requests, reads or writes the local file, and transfers files to or from the remote system.

To transfer a file automatically, users must specify the filestore nickname of the OSI server.

Manually

If the filestore subset of the file to be transferred is not in the filestore of the OSI server, users must manually add the name of the file, without the first qualifier, to their own filestore before that file transfer can be initiated. If users do not have access to OSI/File Services, the OSI/File Services administrator must do this. This can be done using the interactive interface of OSI/File Services. There is no facility within NetView FTP V2.2.1 MVS to do this. Users can also use existing entries in another user’s filestore, if the owner of that filestore has given authorization (access and password).
When files are transferred using OSI protocols, only information that complies with the FTAM standard can be sent to the remote system. Because of this, the following is not available for an OSI file transfer:

- Automatic generation of a report about the transfer for a specified TSO user ID on the remote system.
- Automatically submitting a job on the remote system after the file transfer.
- Remote checking of a request before the file transfer is started.
- Pre-transfer and post-transfer user exits and user-written file handlers.

For more information, refer to the *OSI/File Services System/370 User’s Guide*. This guide also describes how to create the LRD and its entries.

## Deleting Entries from the OSI/File Services Results File

OSI/File Services records the results of file transfers in the Activity Control File.

To delete entries from the Activity Control File, results must be retrieved using the OSI/File Services function GET-RESULT. If the originator of the request and the filestore owner are the same, the result is deleted from the file. If the originator of the request and the filestore owner are not the same, the result is only deleted after both the request originator and filestore owner have issued a GET-RESULT.

The filestore owner may not know about requests initiated by other users and may be prohibited from looking into the results file because he has no access to OSI/File Services. In this case, the OSI/File Services administrator must delete these entries on a regular basis. If the result file becomes full, all subsequent file-transfer requests are rejected by OSI/File Services.

When a file transfer is initiated from the remote system and a file is created locally, the OSI server cannot inform the owner of the filestore in which a new file has been created that a result record was written into the Activity Control File. The filestore owner or the administrator must check the Activity Control File regularly and delete such entries.

For more information about administering OSI/File Services, refer to the *OSI/File Services System/370 User’s Guide*. 
Making Entries in the Local Resource Directory

Before an OSI file transfer can be performed, the following must be included in the Local Resource Directory (LRD):

- Information about filestore accessibility. This is done when the OSI/File Services user is defined. For more information, see “Data Security for OSI File Transfers” on page 183.

- Information about user capabilities. This is done when the OSI/File Services user is defined. For more information, see “Data Security for OSI File Transfers” on page 183.

- To transfer a file:
  - The first qualifier of the MVS file name (called the filestore subset in OSI/FS) must be in the filestore of either the user, the OSI server, or another user who has given authorization to use their filestore.
  - The name of the file that is to be transferred is in a filestore, otherwise the file-transfer request is rejected. If the filestore of the OSI server is used, this is done automatically.

The OSI server must have permission to access entries in the filestore (accessibility for other users in OSI/File Services). In addition, the Application Entity Title of the partner FTAM application must have been registered in OSI/Communications Subsystem together with the corresponding addressing information. For more information, refer to the MVS and VM OSI/Communications Subsystem Configuration and Administration Guide.

Recovering OSI File Transfers

All information concerning a request is written to a recovery file. This file contains information for all requests that have been passed to OSI/File Services, and for which NetView FTP V2.2.1 MVS has not yet been notified as being completed.

If the OSI server abends, the current request is set to waiting. When the OSI server is restarted the list of submitted requests in the recovery file is compared with the requests that were passed to the OSI/File Services. Some requests may already have been processed by OSI/File Services before the OSI server was restarted. Otherwise, any requests still outstanding, or any new requests are set to waiting.

If the queue handler terminates, all outstanding requests are set to waiting.
Chapter 10. Writing Startup Procedures for NetView FTP V2.2.1 MVS

To ease the writing of startup procedures for NetView FTP V2.2.1 MVS, the following were automatically prepared during the installation:

- A queue handler procedure (DVGQHPRO)
- Several server procedures (DVGSVnn)
- Parameters for the queue handler (DVGQHPAS)
- Parameters for servers (PDVGnn and DVGCXSPPP)
- A job (DVGCPROC) to:
  - Create a procedure library
  - Copy DVGQHPRO into the procedure library
  - Copy DVGSVnn into the procedure library and rename it SVPRFnn, where SVPRF is the server prefix chosen by the installation generator.
- A job (DVGCPARM) to:
  - Create a parameter library
  - Rename PDVGnn to DVGSVnn and copy the parameters contained in DVGQHPAS, DVGSVnn, and DVGCXSPPP into the parameter library.
- A job (DVGCREP) to:
  - Create a log file for the queue handler
  - Create a report file for each server
  - Create a server group table that has an entry for each of the servers DVGSVnn.

Procedure to Start the Queue Handler

You always need a cataloged procedure that contains the startup JCL statements to start the queue handler. A sample procedure is shown in Figure 52.

```plaintext
//DVGHQHPRO PROC LANG=ENGLISH, DEFAULT LANGUAGE FOR MESSAGES
// SNAP=A, DEFAULT CLASS FOR TRACE/SNAPDUMP
// UDUMP=A DEFAULT CLASS FOR SYSTEM DUMP
//--------------------------------------------------------------------
// START THE NETVIEW FTP QUEUE HANDLER
//--------------------------------------------------------------------
// NOTE: ADJUST ALL VARIABLES WRITTEN IN LOWERCASE TO YOUR LOCAL ENVIRONMENT
//--------------------------------------------------------------------
//DVGHQHTF EXEC PGM=DVGHQHTF,PARM="&LANG",TIME=144/zerodot,REGION=/zerodotK
//STEPLIB DD DSN=DVG.NFTP23/zerodot.TABLE.SDVGLMD1,DISP=SHR
//SYSUDUMP DD SYSOUT=&UDUMP
//DVGSNAP DD SYSOUT=&SNAP,DCB=(LRECL=125,BLKSIZE=1632)
//DVGTRO DD DSN=DVG.NFTP230.REQUEST.QUEUE,DISP=SHR
//DVGLOG DD DSN=DVG.NFTP230.QLOG,DISP=OLD
//DVGNVT DD DSN=DVG.NFTP230.SRVGRP.TABLE,DISP=OLD
//SYSLIB DD DSN=DVG.NFTP230.PARMLIB(DVGHQHPAS),DISP=SHR
```

Figure 52. Sample Startup Procedure for the Queue Handler (DVGSTQH)

You can change the highlighted parameters shown in Figure 52. You can also change the values of the following, to suit the needs of your installation.
DVGQHPRO
Defines the LANG, SNAP, UDUMP, and LOG parameters.

DVGQHFTP
The load module name of the queue handler after you have installed the product using the jobs provided.

You can use the optional PARM parameter to specify a message definition module other than the default English module.

To specify a language other than English, you must have the necessary language message module available. The name of the module must be DVGCCMDx, where x is the first character of the specified language. The following language names are supported:

- English (U.S.)
- French
- German
- Italian
- Japanese
- Portuguese
- Spanish
- User (any language requiring single-byte encoding)
- Double (any language requiring double-byte encoding).

**Note:** Message modules for these languages are not necessarily available from IBM. The initial distribution tape contains only English messages. However, you can translate the NetView FTP V2.2.1 MVS messages to provide your own message module. For details refer to Chapter 11, “Changing the Text of NetView FTP V2.2.1 MVS Messages and the Language They Are Issued In” on page 129.

Specifying TIME=1440 prevents the queue handler from being canceled by the system because it exceeds a given time limit.

The REGION parameter should be coded as shown to reduce the possibility of storage shortage during the queue handler’s processing. If you specify another value, and a shortage of storage occurs it is possible that the queue handler no longer handles requests with all specifications. For the storage requirements of the queue handler refer to Appendix D, “Storage Requirements” on page 199.

STEPLIB or JOBLIB
Defines the library that is to be searched first for the load module DVGQHFTP. If module DVGQHFTP cannot be found in the specified library, NetView FTP V2.2.1 MVS searches for it in the link library and the LPA library.

If you want to use a language message module other than the default, this module must be available from the current job library, step library, link library, or LPA library.

SYSUDUMP
Defines the data set that is used by the system to produce a dump if the queue handler terminates abnormally and the internal SNAP dump facility is not available.

Although SYSUDUMP is rarely used, you must provide it, in order to obtain enough information if a problem arises.
**DVGSNAP**

Defines the SNAP dump data set, which the queue handler uses for its internal trace facility and for when it encounters an abnormal termination situation.

You can either spool the dump by specifying SYSOUT=x, or select a tape or disk data set.

If you select SYSOUT, you must specify valid DCB parameters for SNAP output.

If you select a data set it must be allocated with the following DCB parameters:

- For a standard dump of 120 characters per line:
  - RECFM=VBA
  - LRECL=125
  - BLKSIZE=882 or 1632.

- For a high density dump on a 3800 printing subsystem of 204 characters per line:
  - RECFM=VBA
  - LRECL=209
  - BLKSIZE=1470 or 2724.

The specification of this DD statement is mandatory. For the amount of data that the queue handler produces during tracing, refer to “Tracing the Queue Handler” on page 139.

**DVGTRQ**

Defines the data set that contains the request queue. For more information about the request queue see Chapter 4, “Setting Up and Maintaining the Request Queue” on page 67. The data-set name shown is an example only. You must specify this DD statement.

**DVGLOG**

Defines the data set into which the queue handler writes all messages except those only displayed on the operator’s console. DVGLOG contains a log of all queue handler commands that were processed during the queue handler’s run time.

The sample shows DVGLOG allocated to a data set. You can also use SYSOUT or tape for DVGLOG. In this case the following DCB parameters must be used:

RECFM=F or FB
LRECL=120.

For RECFM=FB you can specify a block size as a multiple of the LRECL parameter.

This DD statement is mandatory. Because the queue handler logs each queue handler command, this data set can become very large if the queue handler runs continuously for a long time.

Examples of log files are shown in Appendix G, “NetView FTP V2.2.1 MVS Log Files” on page 219.
DVGNDT

Defines the data set that contains the server group table. For more information about the server group table refer to Chapter 7, “Setting Up a Server Group Table” on page 85. If you want to make changes to the server group table while the queue handler is running, specify FREE=CLOSE in the queue handler startup procedure. However, for the changes to become active, the queue handler must be stopped and restarted. This is shown in the following example:

```
//DVGNDT DD DSN=DVG.NFTP23.V2R2M0.SRVGRP,DISP=SHR,FREE=CLOSE
```

Figure 53. Sample Server-Group Table Definition

The data-set name shown is an example only. If you do not want to use a server group table you can omit this DD statement.

SYSIN

Defines the data set from which the queue handler reads its initialization parameters. Some of these initialization parameters are mandatory, so you must have this data set. For more information on the initialization parameters of the queue handler, read “Specifying Values for the Queue Handler Initialization Parameters” on page 99. The data-set name shown in the figure is an example only. The data set must be allocated with the following DCB parameters:

- RECFM=F or FB
- LRECL=80.

For RECFM=FB you can specify a block size that is a multiple of the logical record length.
Specifying Values for the Queue Handler Initialization Parameters

The initialization parameters are read from SYSIN at queue handler startup time. The parameters are coded as initialization control statements in the format:

```
   KEYWORD=value
```

For information on coding rules, see the chapter on writing NetView FTP V2 MVS batch jobs in the *NetView FTP V2 MVS User's Guide*. The keywords for the queue-handler initialization parameters are shown in Figure 54.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Keyword</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Name</td>
<td>NFTPNAME</td>
<td>99</td>
</tr>
<tr>
<td>Maximum Number of Servers</td>
<td>MAXSRV</td>
<td>100</td>
</tr>
<tr>
<td>Server Prefix</td>
<td>SRVPREF</td>
<td>100</td>
</tr>
<tr>
<td>Delay Time</td>
<td>DELAY</td>
<td>100</td>
</tr>
<tr>
<td>Automatically Started Servers</td>
<td>STARTSV1</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>STARTSV2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTSV3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTSV4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTSV5</td>
<td></td>
</tr>
<tr>
<td>Master Password</td>
<td>ADMIN</td>
<td>101</td>
</tr>
<tr>
<td>Number of FSB Slots</td>
<td>FSBSLOTS</td>
<td>101</td>
</tr>
<tr>
<td>CSA Storage Limit</td>
<td>CSAELIM</td>
<td>102</td>
</tr>
<tr>
<td>Events</td>
<td>NVEVENT</td>
<td>102</td>
</tr>
<tr>
<td>SMF Recording</td>
<td>SMFREC</td>
<td>103</td>
</tr>
<tr>
<td>Internal Trace</td>
<td>TRACE</td>
<td>103</td>
</tr>
</tbody>
</table>

Installation Name

The name that you assign to the NFTPNAME keyword defines your NetView FTP V2.2.1 MVS installation in the network. Use the same name as the JES network node of your system if you use the JES network node name, otherwise use a name that is unique in your network and ensure that the report recipient parameters are coded with the correct location ID.
This name is used:

- To identify your NetView FTP V2.2.1 MVS in the checkpoint records that the servers write during file transfers when receiving. Using this name enables any server running within your network node to process a transfer restart.
- As a default for the node name of the local recipient of the file-transfer report, if no node name was specified in the request.

You can use this name when you define your local NetView FTP V2.2.1 MVS in your own server group table. The other installations in your network can also use this name when they define your NetView FTP in their server group tables. The relationship between NFTPNAME and the server group table are shown in Chapter 7, “Setting Up a Server Group Table” on page 85.

You must specify NFTPNAME, the name can be up to eight alphanumeric characters long, starting with a letter.

**Maximum Number of Servers**

MAXSRV defines how many servers can be started on a system. The queue handler rejects any attempt to start a server if this limit has already been reached.

MAXSRV can have any value between 1 and 99. The default value is 20.

![MAXSRV Diagram]

**Server Prefix**

SRVPREF is a common prefix for the names of the cataloged procedures for your servers. This enables the operator to specify only the numeric suffix when starting and stopping a server from the console. SRVPREF can be up to six alphanumeric characters, following the rules for such names in MVS.

The default prefix is FTPSRV.

![SRVPREF Diagram]

**Delay Time**

DELAY defines the minimum amount of time that requests are to be left in the request queue after they have been given the status finished.

If the queue handler rebuilds the request queue and encounters a finished request for which the delay time has elapsed, it deletes it from the request queue.

Keeping a request in the request queue lets the user query the file-transfer request even after the corresponding file transfer has finished. Specifying a long delay time can lead to the request queue filling up. The delay time is specified in hours, and can be from 0 to 999. The default value is 48.
Automatically Started Servers

Use these parameters to specify the suffixes of the cataloged procedure names of the servers that are to be started automatically when the queue handler finishes its initialization.

Each suffix is a 2-digit number from 00 to 99. The total number of suffixes specified cannot exceed the value specified for the MAXSERV parameter.

If you omit these parameters, no servers are started automatically when the queue handler finishes its initialization.

For example:

```
STARTSRV1=/zerodot1
STARTSRV2=(/zerodot5,4/zerodot,11)
```

Master Password

The value assigned to the ADMIN keyword defines the master password. This must be used if the issued command (such as delete, modify, or query) is restricted to a NetView FTP V2.2.1 MVS administrator. For a description of the function of a NetView FTP V2.2.1 MVS administrator, refer to the NetView FTP V2 MVS User's Guide.

The specification of this parameter is optional. If this parameter is missing, NetView FTP V2.2.1 MVS does not allow any NetView FTP V2.2.1 MVS administrator functions.

If you do not want to keep the master password with the other initialization parameters in a single data set, you can specify it in a separate data set and concatenate this data set to SYSIN. The master password can be up to eight characters.

Number of FSB Slots

FSBSLOTS defines the number of slots in the FSB. This affects the FSB size, which uses CSA storage, and the total number of concurrent subtasks.

Appendix D, “Storage Requirements” on page 199 provides details of the FSB storage requirements.
**total**  Defines the total number of FSB slots. It can be any value from 1 to 32.

**tso**  Defines the number of FSB slots that are reserved for interactive interface users. It can be any value from 0 to 31.

The default specification is: FSBSLOTS=(6,3).

### CSA Storage Limit

CSAELIM limits the additional CSA storage that is needed for file-transfer requests generating a NetView FTP Shared Block Extension (FSBX).

Specifying the CSA storage limit is important if you have virtual storage constraints.

The parameter value specifies the total additional CSA space that can be dynamically allocated when file transfers that require an FSBX are being processed. The value specified must be large enough to avoid performance bottlenecks, and small enough to guarantee that this additional space is available.

Appendix D, “Storage Requirements” on page 199 provides details of the FSBX storage requirements.

```
┌ ┐ ─NOLIM─
   /
 ──CSAELIM= ──┼ ┼─
n       ───────────────────────────────────────/
└ ┘ ─

CSAELIM

An optional parameter.

**NOLIM**  Specifies that no limit is set for CSA storage allocation for the FSBX. This is the default value.

**n**  Specifies the total number of bytes of CSA storage to be used for the FSBX. You can specify any value from 0 to 999 999.

**nK**  Specifies the total number of kilobytes of CSA storage to be used for the FSBX. You can specify any value from 0KB to 999KB (KB equals 1024 bytes).

### Events

NVEVENT indicates if NetView FTP V2.2.1 MVS is to write operator terminal messages for the following events:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I/O error on the data set</td>
</tr>
<tr>
<td>2</td>
<td>VSAM error on the request queue, checkpoint data set, sending data set, or receiving data set</td>
</tr>
<tr>
<td>3</td>
<td>Abend during I/O</td>
</tr>
<tr>
<td>21</td>
<td>Queue handler ready</td>
</tr>
<tr>
<td>22</td>
<td>Queue handler stopped</td>
</tr>
</tbody>
</table>
Abnormal termination of the queue handler
Request queue status
Server status
Request queue rebuild
File-transfer request added.

These events are reported in such a way that they can be recognized and processed by NetView CLISTs. However, if there is no NetView CLIST active, the messages are sent to the operator console.

NVEVENT is an optional parameter.

SMF Recording
SMFREC indicates for which SMF record SMF recording is to be performed.

\(n\) Specifies the SMF record number under which the SMF recording takes place. A value from 000 to 255 can be specified. If nothing is specified, no SMF recording is done. Values from 000 to 127 are reserved for MVS components or IBM subsystems. Values from 128 to 255 are available for use by other programs.

Internal Trace
Specifying a value for the TRACE keyword tells NetView FTP V2.2.1 MVS to take an internal trace while the queue handler runs.

GROUP\(n\) Specifies one of the queue-handler module groups
CL\(m\) Specifies a trace class.

Module groups and trace classes for queue-handler traces are described in detail in “Tracing the Queue Handler” on page 139.

Use the internal trace for maintenance purposes only. Do not use it during normal production time, as it severely degrades the performance of the queue handler. For more information on the trace facility, refer to Chapter 12, “Taking Traces” on page 135.
Sample Initialization Parameters for the Queue Handler

The following example shows a set of initialization parameters for the queue handler:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFTPNAM=CENTRAL4</td>
<td>Local JES network node name</td>
</tr>
<tr>
<td>DELAY=0</td>
<td>No guaranteed inquiry time</td>
</tr>
<tr>
<td>ADMIN=SECRET</td>
<td>This is the master password</td>
</tr>
<tr>
<td>STARTSV1=(1,4,34,57)</td>
<td>Start these servers automatically</td>
</tr>
</tbody>
</table>

Figure 55. Sample Initialization Parameters for the Queue Handler

- The name of this NetView FTP V2.2.1 MVS installation is CENTRAL4, which is the same name as that of the local JES network node.
- After a file-transfer request has been given the status **finished**, it can be deleted from the request queue when the queue handler performs a request queue rebuild.
- Since the SRVPREF has not been specified, it defaults to FTPSRV.
- The master password is SECRET.
- The STARTSV1 parameter automatically starts the following servers when the queue handler starts:
  - FTPSRV1
  - FTPSRV4
  - FTPSRV34
  - FTPSRV57.
Procedure to Start a Server

The following describes how to start an SNA server and an OSI server.

An SNA or OSI server can be started by:

- Submitting a job. Samples are provided in DVGSTSV in library
  **DVG.NFTP230.INSTLIB**. You can modify the jobs to suit your own system.
- By writing cataloged procedures that start a server automatically or can start a
  server by entering operator commands.

The JCL statements of a sample procedure to start an SNA server automatically or
by the operator are shown in Figure 56.

The JCL statements of a sample procedure to start an OSI server automatically or
by the operator are shown in Figure 57 on page 109.

**Note:** DVGSVAPP must be activated, see “Setting Up ACF/VTAM Resources” on
page 22.

```
//DVGSV03 PROC LANGL=ENGLISH, default language for log file messages
// LANGR=ENGLISH, default language for report file messages
// SNAP=A, default class for TRACE/SNAPDUMP
// UDUMP=A, default class for SYSTEM DUMP
// PRINT=A default class for SYSPRINT
///c5197--------------------------------------------------------------------
///c5197 Start a NetView FTP V2.2.1 MVS Server
///c5197--------------------------------------------------------------------
//DVGSTSV EXEC PGM=DVGSVFTP,PARM='&LANGL,&LANGR'
// STEPLIB DD DSN=DVG.NFTP230.SDVGLMD,DISP=SHR
// DVGCR DD DSN=DVG.NFTP230.CHECKPNT.FILE,DISP=SHR
// DVGEXEC DD DSN=DVG.NFTP230.DVGEXEC,DISP=SHR
// DVGSNAP DD SYSSOUT=&SNAP,DCB=(LRECL=125,BLKSIZE=1632)
// DVGLOG DD DSN=DVG.NFTP230.SVLOG03,DISP=OLD
// SYSPRINT DD SYSSOUT=SPRTINT
// SYSUDUMP DD SYSSOUT=UDUMP
// SYSLIB DD DSN=DVG.NFTP230.PARMLIB(DVGSVP03),DISP=SHR
// DVGSTSV03...```

**Figure 56. Sample Startup Procedure for a Server (DVGSV03)**

You can change the highlighted parameters shown in Figure 56. Specify all other
parameters as shown.

**DVGSV03**

Defines the procedure name. Symbolic variables can be given a default
value. However, they can be overwritten when the procedure is called. It
defines the LANG, SNAP, UDUMP, PRINT, and LOG parameters. The procedure
must have a name in the form `pref suf`, where `pref` is the prefix specified
for the SRVPREF keyword and `suf` is a 2-digit number (00 to 99). For
example, if the value of the SRVPREF keyword in the queue handler initialization
parameters is FTPSRV, a valid name is FTPSRV03 or FTPSRV64.
**DVGSTSV**

Defines the procedure step name. Symbolic variables can be specified here to be passed to the server. It is the load module name of the server after you have installed the product using the jobs provided.

You can use the optional PARM parameter to specify a message definition module other than the default English module.

To specify a language other than English, you must have the necessary language message module available. The name of the module must be DVGCCMDx, where x is the first character of the specified language. The following languages are supported:

- English (U.S.)
- French
- German
- Italian
- Japanese
- Portuguese
- Spanish
- User (any language requiring single-byte encoding)
- Double (any language requiring double-byte encoding).

Note: Message modules for these languages are not necessarily available from IBM. The initial distribution tape contains only English messages. However, you can translate the NetView FTP V2.2.1 MVS messages to provide your own message module. For details refer to Chapter 11, "Changing the Text of NetView FTP V2.2.1 MVS Messages and the Language They Are Issued In" on page 129.

Depending on their destination, the server messages can have different languages. NetView FTP V2.2.1 MVS distinguishes between two destination classes for the server messages:

**Class 1**

- Messages written to the server log file
- Messages written to the operator's console.

**Class 2**

- Messages written to the file-transfer report file
- Messages issued to a TSO user's terminal (transfer-completion messages).

You can specify in the PARM parameter the language for class 1 independently from that for class 2. This is shown in the following example:

```
//.... EXEC PGM=DVGSVFTP,PARM='(JAPANESE,ENGLISH)',...
```

The first value of the PARM parameter defines the language for class 1.
The second value of the PARM parameter defines the language for class 2.
If no value is specified, the default ENGLISH is assumed for classes 1 and 2.
If only one value is specified, it applies to classes 1 and 2.
If for class 1, a language is specified that needs double-byte character encoding (Double or Japanese), the messages to the operator’s console are issued in English.

If for class 2, a language is specified that needs double-byte character encoding (Double or Japanese), the messages to the TSO user’s terminal are issued in English.

With the TIME parameter you can prevent the server from being canceled by the system because it exceeds a given time limit. For this purpose specify TIME=1440. This prevents the system from canceling the server.

The REGION parameter should be coded as shown to reduce the possibility of storage shortage during the server’s processing.

you specify another value, and a shortage of storage occurs it may be possible that the server will no longer establish additional sessions. For the storage requirements of the server refer to Appendix D, “Storage Requirements” on page 199.

**STEPLIB or JOBLIB**
Defines the library that is to be searched first for the load module DVGSVFTP. If DVGSVFTP cannot be found in the specified library, NetView FTP V2.2.1 MVS searches for it in the link library and the LPA library.

If you want to use a language message module other than the default, this module must be available from the current job library, step library, link library, or LPA library.

The same is true for all user-exit routines and post-transfer programs that a server is to invoke.

**DVGCR**
Defines the checkpoint data set. If you want to share the checkpoint data set between multiple servers you must code DISP=SHR. For more information about the checkpoint data set refer to Chapter 5, “Setting Up and Maintaining the Checkpoint Data Set” on page 73.

The data-set name shown in the figure is an example only.

The specification of this DD statement is mandatory.

**DVGEXEC**
Defines the data set containing the REXX library.
**DVGREP**

Defines the file-transfer report file data set in which the server writes all the messages it issues during one file transfer. The server sends the file-transfer report file to a TSO, CMS, or ICCF user if this is specified in the file-transfer request.

The file-transfer report file is overwritten with new messages for the next file transfer.

You must allocate the file-transfer report file on DASD using the following DCB parameters:

- **RECFM=F or FB**
- **LRECL=80.**

For **RECFM=FB** you can specify a block size as a multiple of the **LRECL** parameter.

The file-transfer report file must be defined large enough to contain 1000 records.

The data-set name shown in the figure is an example only.

The specification of the DD statement is optional. If you omit it, the server cannot send a report to an interactive user even if this is specified in the file-transfer request.

**DVGSNAP**

Defines the SNAP dump data set, which the server uses for its internal trace facility and for when it encounters an abnormal termination situation.

You can either spool the dump by specifying **SYSOUT=x** or select a tape or disk data set.

If you select **SYSOUT**, you must specify valid DCB parameters for SNAP output.

If you select a data set it must be allocated with the following DCB parameters:

- For a standard dump of 120 characters per line:
  - **RECFM=VBA**
  - **LRECL=125**
  - **BLKSIZE=882 or 1632.**

- For a high density dump on a 3800 printing subsystem of 204 characters per line:
  - **RECFM=VBA**
  - **LRECL=209**
  - **BLKSIZE=1470 or 2724.**

This DD statement is mandatory.
**DVGLOG**

Defines the data set into which the server writes all messages except those displayed only on the operator’s console. DVGLOG contains a log of all file transfers that were processed during the server’s runtime.

The sample shows DVGLOG allocated to a data set. You can also use a SYSOUT or tape for DVGLOG. In this case the following DCB parameters must be used:

- **RECFM=F or FB**
- **LRECL=120.**

If you specify RECFM=FB, you can specify a block size as a multiple of the LRECL parameter.

This DD statement is mandatory.

**SYSUDUMP**

Defines the data set that is used by the system to produce a dump if the server terminates abnormally and the internal SNAP dump facility is not available.

Although SYSUDUMP is unlikely to be used, you must provide it in order to obtain enough information if a problem arises.

**SYSIN**

Defines the data set from which the server reads its initialization parameters. Some of these initialization parameters are mandatory, so you must have this data set. For more information on the initialization parameters of the server, read “Specifying Values for the Server Initialization Parameters” on page 113.

The data-set name shown in the figure is an example only.

The data set must be allocated with the following DCB parameters:

- **RECFM=F or FB**
- **LRECL=80.**

For RECFM=FB you can specify a block size as a multiple of the LRECL parameter.

```plaintext
//DVGSVOSI PROC LANGL=ENGLISH, default language for log file messages
// LANGR=ENGLISH, default language for report file messages
// SNAP=A, default class for TRACE/SNAPDUMP
// UDUMP=A, default class for SYSTEM DUMP
// PRINT=A default class for SYSPRINT
/**------------------------------------------------------------------**/
/** Start a NetView FTP V2.2.1 MVS OSI Server                        **
/**------------------------------------------------------------------**/
//DVGSTOSI FTP EXEC PGM=DVGMSOCM, PARM='&LANGL,&LANGR', TIME=1440, REGION=6K
//STEPLIB DD DSN=DVG.NFTP23.SDVGLMD3,DISP=SHR
//DVGRF DD DSN=DVG.NFTP23.RECOVERY.FILE,DISP=SHR
//DVGREP DD DSN=DVG.NFTP23.REPORT.FILEOSI,DISP=SHR
//DVGSNAP DD SYSOUT=&SNAP, DCB=(LRECL=125,BLKSIZE=1632)
//DVLOG DD DSN=DVG.NFTP23.SVLOGOSI
//SYSPRINT DD SYSOUT=&PRINT
//SYSUDUMP DD SYSOUT=&UDUMP
//SYSIN DD DSN=DVG.NFTP23.PARMLIB(DVGCXOSI),DISP=SHR
```

Figure 57. Sample Startup Procedure for an OSI Server (DVGSVOSI)
You can change the highlighted parameters shown in Figure 57. Specify all other parameters as shown.

The parameters are the same as those for the startup procedure for an SNA server except for one:

**DVGRF**

Defines the recovery file. For more information about the recovery file refer to “Recovering OSI File Transfers” on page 94.

The data-set name shown in the figure is an example only.

The specification of this DD statement is mandatory.

---

### Batch Jobs to Start a Server

The JCL statements for a sample batch job to start an SNA server are shown in Figure 58.

```
//jobname JOB (accn,n), 'id',
//MSGCLASS=x,MSGLEVEL=(1,1),
//CLASS=A,NOTIFY=uid,USER=uid
// *--------------------------------------------------------------------
//  START A NETVIEW FTP V2 MVS SERVER IN CONTINUOUS MODE
// *--------------------------------------------------------------------
//DVGSVFTP EXEC DVGSV/zerodot3
//SYSIN DD /c5197
// *--------------------------------------------------------------------
/* Initialization Parameters for the Server */
LUNAME=FTP/zerodot2 logical unit name
DLOGMOD=FTPBIND logmode table entry name
CLASS=5 served request class
/*

Figure 58. Sample Batch Startup Job for a Continuous Mode SNA Server Using the Server Procedure
```
The JCL statements for a sample batch job to start an OSI server are shown in Figure 59.

```
//jobname JOB (accn,n),'id',
// MSGCLASS=x,MSGLEVEL=(1,1),
// CLASS=A,NATTRY=uid,USER=uid
// *---------------------------------------------------------------------
/* START A NETVIEW FTP V2 MVS OSI SERVER */
// *---------------------------------------------------------------------
//DVGSVOSI EXEC DVGSVOSI
//SYSIN DD /c5197
/* Initialization Parameters for the OSI Server */
OISVNAME=DVGSVOSI OSI server name
CLASS=O
MAXSESS=10
AVLSESS=9
AVLLRH=9
AVLRRH=0
SRVMODE=C
OBTINTVL=120
OIRECDEL=120 recovery delay time
OIFTAMID=ftamid VTAM LU name of OSI/FS
OIAAPPLID=applid VTAM LU name of OSI server
OIUSERID=userid local user name
OIUSERPWD=password local user password for OSI server
/*
```

Figure 59. Sample Batch Startup Job for an OSI Server Using the Server Procedure

For an explanation of the JCL statements shown in the batch jobs refer to “Procedure to Start a Server” on page 105. The server initialization parameters shown replace the contents of the data set specified by the SYSIN DD statement in the server procedure.
The JCL statements for a batch job to start a recursive single-mode server are shown in Figure 60.

```jcl
//jobname JOB (accn,n),'id',
// MSGCLASS=x,MSGLEVEL=(1,1),
// CLASS=A,NOTIFY=uid,USER=uid
//------------------------------------------------------------------------------
/* START NETVIEW FTP V2 MVS SERVER IN SINGLE TRANSFER MODE */
//------------------------------------------------------------------------------
//SINGLE EXEC DVGSV03
//SYSIN DD *
* Initialization Parameters for the Server
* *************************************************************
LUNAME=FTP02 logical unit name
DLOGMOD=FTPBIND logmode-table entry name
CLASS=G served request class
MAXSESS=1 maximum number of simultaneous sessions
SRVMODE=SINGLE server runs in single transfer mode

/*
/* Check if server terminated successfully after a transmission
/*
COND=((0000,EQ,SINGLE.DVGSVFTP), +
(0400,EQ,SINGLE.DVGSVFTP), +
(0800,EQ,SINGLE.DVGSVFTP), +
(1200,EQ,SINGLE.DVGSVFTP), +
(1600,EQ,SINGLE.DVGSVFTP), +
(0004,EQ,SINGLE.DVGSVFTP), +
(0404,EQ,SINGLE.DVGSVFTP))
/*******************************************************************************/
/* Start a batch job for subsequent processing if all checks are negative. A severe error occurred. */
/*
SYSPRINT DD SYSOUT=*
SYST1 DD DSN=OTHER.JOBS.CNTL(NEXTPROC),DISP=SHR
SYST2 DD SYSOUT=(A,INTRDR)
SYSSIN DD DUMMY

/* NetView FTP V2.2.1 MVS Server in single mode */

/* RESTART EXEC PGM=IEBGENER, */
/* Check if server terminated abnormally or was stopped by the operator. */
COND=((0000,EQ,SUBMIT))

/* Restart the NetView FTP V2.2.1 MVS Server in single transfer mode */
SYSPRINT DD SYSOUT=*
SYST1 DD DSN=GDG.JOBS.CNTL(STSINGLE),DISP=SHR
SYST2 DD SYSOUT=(A,INTRDR)
SYSSIN DD DUMMY

/* Figure 60. Sample Batch Job for a Single Mode Server */
```

NetView FTP V2 MVS Installation, Operation, and Administration
You must change the values shown in lowercase type in Figure 60 to meet the needs of your installation. You can also change the highlighted parameters. All other parameters must be specified as shown. For an explanation of the JCL statements shown refer to “Procedure to Start a Server” on page 105. When the server is started, it waits until it either obtains a request from its queue handler or is contacted by a remote server. When it establishes a session, it starts the file transfer. When the file transfer ends, whether successfully or not, the server terminates. Subsequent steps of the batch job can check the condition code from the transfer.

If the condition code indicates that the transfer was successful, or that only a transmission error occurred, the server can be restarted. If the condition code indicates that a severe error occurred, the server is not restarted; instead, for example, the log file can be printed.

**Note:** The condition codes from single-mode servers and continuous-mode servers are derived differently. See *NetView FTP Messages and Codes* for details.

---

**Specifying Values for the Server Initialization Parameters**

The following describes the initialization parameters required for an SNA server and an OSI server.

The initialization parameters are read from SYSIN at server startup time. The parameters are coded as initialization control statements in the format:

```
>KEYWORD=value
```

For information on coding rules, see the chapter on writing NetView FTP V2 MVS batch jobs in the *NetView FTP V2 MVS User's Guide*. 
Initialization Parameters for an SNA Server
The keywords for the initialization parameters for an SNA server are shown in Figure 61.

**Figure 61. Server Initialization Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Keyword</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAM LU Name</td>
<td>LUNAME</td>
<td>115</td>
</tr>
<tr>
<td>Server Classes</td>
<td>CLASS</td>
<td>115</td>
</tr>
<tr>
<td>Server Running Mode</td>
<td>SRVMODE</td>
<td>116</td>
</tr>
<tr>
<td>Termination Time</td>
<td>TTIME</td>
<td>116</td>
</tr>
<tr>
<td>Mode Name</td>
<td>DLOGMOD</td>
<td>117</td>
</tr>
<tr>
<td>Byte Checkpoint Interval</td>
<td>BYTECR</td>
<td>117</td>
</tr>
<tr>
<td>Time Checkpoint Interval</td>
<td>TIMECR</td>
<td>117</td>
</tr>
<tr>
<td>Default First Qualifier for a new receiving file</td>
<td>GID</td>
<td>118</td>
</tr>
<tr>
<td>Obtain Interval</td>
<td>OBTINTVL</td>
<td>118</td>
</tr>
<tr>
<td>Retry Delay Time</td>
<td>RETRYDELAY</td>
<td>118</td>
</tr>
<tr>
<td>Report Spool Class</td>
<td>REPCLASS</td>
<td>119</td>
</tr>
<tr>
<td>Names of Pre-Transfer and Post-Transfer User-Exit Routines</td>
<td>PPEXIT</td>
<td>119</td>
</tr>
<tr>
<td>Name of Pre-Transfer User-Exit Routine</td>
<td>PRETRAN</td>
<td>119</td>
</tr>
<tr>
<td>Name of Post-Transfer User-Exit Routine</td>
<td>POSTTRAN</td>
<td>119</td>
</tr>
<tr>
<td>Name of the Post-Conversation User-Exit Routine</td>
<td>POSTCONV</td>
<td>119</td>
</tr>
<tr>
<td>Limit to the Maximum Number of Simultaneous File Transfers</td>
<td>MAXSESS</td>
<td>120</td>
</tr>
<tr>
<td>Maximum Number of Simultaneous File Transfers</td>
<td>AVLSESS</td>
<td>120</td>
</tr>
<tr>
<td>Maximum Number of Simultaneous Locally-Initiated File Transfers</td>
<td>AVLLRH</td>
<td>120</td>
</tr>
<tr>
<td>Maximum Number of Simultaneous Remotely-Initiated File Transfers</td>
<td>AVLRRH</td>
<td>120</td>
</tr>
<tr>
<td>Security Parameters</td>
<td>SECPAR</td>
<td>121</td>
</tr>
<tr>
<td>Events</td>
<td>NVEVENT</td>
<td>122</td>
</tr>
<tr>
<td>Suppression of STOW Message</td>
<td>SUPSTOW</td>
<td>123</td>
</tr>
<tr>
<td>Internal Trace</td>
<td>TRACE</td>
<td>123</td>
</tr>
<tr>
<td>SMF Recording</td>
<td>SMFREC</td>
<td>89</td>
</tr>
</tbody>
</table>
VTAM LU Name
Use the LUNAME keyword to specify the ACF/VTAM LU name of the server and, if required, its password.

\textit{luname} Can be up to 8 characters.
\textit{password} Can be up to 8 characters.

You must specify a value for this parameter.

\begin{verbatim}
 fashion

\end{verbatim}

Server Classes
Use the CLASS keyword to assign one or more classes to a server. A server is allowed to process a request only if the server class specified in the request is one of the classes assigned to the server.

The value of this parameter can be a single class, a list of up to 15 classes (0 through 9 or A through Z) enclosed in parentheses and separated by commas, or the word ALL (for all classes). The default value is ALL.

\begin{verbatim}
 fashion

\end{verbatim}

You can use the different classes to route requests to the server best suited to carry out the corresponding file transfer. For example, at your location a certain class might be reserved for each of the following:

- File transfers needing a server running in single mode
- File transfers to be carried out at night
- Long duration, high-volume file transfers
- File transfers between certain locations
- Fast file transfers, requiring a server with high-dispatching priority, a large RU size, and a long checkpointing interval
- File transfers on error prone ACF/VTAM lines, requiring a server with a small RU size, and a short checkpointing interval
- Tape file transfers, so that outstanding tape mounts do not prevent other file transfers from being carried out
- File transfers to be carried out by a server with an internal trace started, to test user-exit modules or to investigate problems without affecting other servers.
Server Running Mode
Use the SRVMODE keyword to specify in which mode a server is to run:

SINGLE
Single running mode.
A server can only carry out one file transfer. When the file transfer is finished the server stops.

CONT
Continuous running mode, update of GDG table.
A server carries out any number of consecutive file transfers. It runs until it is stopped. This is the default value.

After a generation data set is transferred, the GDG table of this data set is updated. If you are using MVS/XA, program temporary fix (PTF) OY10531 must be installed.

(CONT,GDG)
Continuous running mode, update of GDG table.
This parameter corresponds to parameter CONT. It is still used to support compatibility.

Termination Time
Use the TTIME keyword to specify the termination time of a server. When the specified time is reached, the server stops automatically. If a file transfer is in progress when a server’s termination time is reached, the server waits until the file transfer is finished, before terminating.

hh:mm The time in 24-hour format. For example, 3:23 pm is specified as 15:23.

yy/mm/dd The date in year-month-day format. For example, 11 March 1994 is specified as 94/03/11. The default is the current date.

If no value for this parameter is specified, the server runs indefinitely.
Mode Name
Use the DLOGMOD keyword to specify the name of an entry in the user-defined ACF/VTAM logon mode table.

An entry for this mode name must be defined in the logon mode table for the remote server LU. The default value for the DLOGMOD keyword is FTPBIND. For further information see “Creating a Logon Mode Table” on page 82.

Byte Checkpoint Interval

You can specify the byte checkpoint interval as follows:

\[ nK \] Number of kilobytes between checkpoints; an integer from 1 to 9999.

\[ nM \] Number of megabytes between checkpoints; an integer from 1 to 999.

\[ 0 \] This indicates that the server does not automatically take checkpoints. This does not preclude exceptional checkpointing, that is, taking checkpoints under special situations, such as a line interruption.

The more often checkpoints are taken, the smaller the amount of data that must be retransmitted should the file transfer be interrupted. However, taking checkpoints takes time and uses system resources. You must determine an appropriate value for this parameter empirically.

The default value of 32K is used if BYTECR or TIMECR is not specified.

If you specify a value for this parameter, do not specify a time checkpoint interval.

Time Checkpoint Interval

Use the TIMECR keyword to specify the time interval at which a server is to write checkpoints to the checkpoint file when it is the receiving server during a file transfer.

\[ n \] Specifies the time in seconds between checkpoints; an integer from 1 to 9999.

\[ 0 \] This indicates that the server should not automatically take checkpoints. The server will still perform exceptional checkpointing, that is, it will take checkpoints in special situations, such as a line interruption.

The more often checkpoints are taken, the smaller the amount of data that must be retransmitted should the file transfer be interrupted. However, taking checkpoints takes time and uses system resources. You must determine an appropriate value for this parameter empirically.
If you specify a value for this parameter, do not specify a byte checkpoint interval. If TIMECR is not specified, the value for BYTECR is used.

**Default First Qualifier for a New Receiving File**

If a server receives a file transferred from a remote system, and if no file name was specified for the receiving file, the server creates a name for it. Use the GID keyword to specify the character string that the server must use as the first qualifier when creating a name for a receiving file. The value of this parameter can be a string of up to 8 characters.

For more information about how NetView FTP creates names refer to the *NetView FTP V2 MVS User’s Guide*.

**Obtain Interval**

Use the OBTINTVL keyword to specify the amount of time, in seconds, a server must wait after unsuccessfully trying to obtain a request before trying again. The value of this parameter can be any integer from 1 to 9999. The default value is 120.

**Retry Delay Time**

If a file transfer is interrupted and if the corresponding request specified that automatic transfer restart was to be used, the local server writes the request back to the request queue and changes its status back to waiting. This way, the request is rescheduled later. The server also sets the request’s not-before date and time to the current time plus the number of seconds specified for the RETRYDELAY keyword. This forces the server (and any other servers that are eligible to process that request) to wait at least that number of seconds before processing that request again. As long as the new not-before date and time has not passed, the request has the status waiting and is not dispatchable.

The value can be a number in the range 1 through 9999. The default value is 300.

**Note:** The NetView FTP V2.1 MVS keyword REQDELAY is still accepted. However, you should use the keyword RETRYDELAY for servers being newly set up.
Report Spool Class
Use the REPCLASS keyword to specify the SYSOUT class to which servers are to spool the file-transfer reports they create.

The value of REPCLASS is a letter from A to Z or an integer from 0 to 1. The default value is B.

Names of the Pre-Transfer, Post-Transfer, and Post-Conversation User-Exit Routines
The PRETRAN keyword is used to specify the name of the pre-transfer user-exit routine. Use the POSTTRAN keyword to specify the post-transfer, and the POSTCONV keyword to specify the post-conversation user-exit routine. For compatibility with previous releases of NetView FTP V2.2.1 MVS, the PPEXIT keyword can also be used to specify the name or names of the pre-transfer or post-transfer user-exit routines, but not the post-conversation user-exit routine. Each name can be a string of one to eight characters.

PPEXIT=name
The pre-transfer and the post-transfer user-exit routines are called as name.

PPEXIT=name1,name2
The pre-transfer user-exit routine is called as name1. The post-transfer user-exit routine is called as name2.

PRETRAN=name
The pre-transfer user-exit routine is called as name.

POSTTRAN=name
The post-transfer user-exit routine is called as name.

POSTCONV=name
The post-conversation user-exit routine is called as name.

Note: PPEXIT continues to be supported by NetView FTP V2 MVS for migration reasons. However, if possible, use the new keywords.

If you do not specify a value for this parameter, the server does not use the pre-transfer, post-transfer, or post-conversation user exits.
For further information on using user exits and when they are called, refer to "NetView FTP Customization."

**Limit to the Maximum Number of Simultaneous File Transfers**

MAXSESS specifies the limit to the maximum number of file transfers that a server can hold by limiting the number of subtasks.

For servers running in single mode, the MAXSESS value must be 1. The value of this parameter is an integer from 1 to 32. The default value is 16.

This value must not exceed the value of the DSESLIM parameter specified in the ACF/VTAM APPL statement. See “Defining an SNA Server as an ACF/VTAM Application Program Node” on page 79.

**Maximum Number of Simultaneous File Transfers**

AVLSESS specifies the number of simultaneous file transfers that a server can hold. The operator can change this parameter while the server is running, but the value must never exceed the value specified in the MAXSESS server initialization parameter.

The value of this parameter can be an integer from zero to the value of MAXSESS.

A value of zero indicates that neither locally nor remotely-initiated requests are served. The default is the value of MAXSESS.

**Maximum Number of Locally-Initiated File Transfers**

AVLLRH specifies the maximum number of locally-initiated file transfers that can be active in a server. It is used as a limit value at startup time. This value can be modified by the operator. The value of this parameter can be an integer from zero to the value of AVLSESS.

A value of zero indicates that no locally initiated requests are served. The default value is 1, if the value of AVLSESS is less than 3, otherwise, the default is \((AVLSESS - 2)\).

**Maximum Number of Simultaneous Remotely-Initiated File Transfers**

AVLRRH specifies the maximum number of remotely-initiated file transfers that can be simultaneously active in a server. It is used as a limit value at startup time. This value can be modified by the operator.
The value of this parameter can be an integer from zero to the value of AVLSESS.

A value of zero indicates that no remotely-initiated requests are served. The default value is 1, if the value of AVLSESS is less than 3, otherwise, the default is (AVLSESS-2).

---

Security Parameters

SECPAR specifies the way in which a NetView FTP V2 MVS server handles security parameters specified in a request (xSECURP) for a file transfer.

---

YES

The requesting system is a system that processes local requests (one created at the local server’s system). The server does one of the following:

- If the originator of the request specifies security parameters, the server uses these.
- If the originator of the request specifies an asterisk (*) for the security parameters for the file at the requesting system, the server retrieves the security parameters and automatically inserts them into the request. If an asterisk is also specified for the responding system, the values retrieved for the requesting system are inserted for the responding system.
- If the originator of the request does not specify security parameters for the file at the requesting system, the server retrieves the parameters and automatically inserts them into the request.

The responding system is a system that processes remote requests (ones created at another system). The local server carries out the transfer only if the security parameters are specified for the file at its system.

This is the default value.

NO

When processing a local request (one created at the local server’s system), the server does one of the following:

- If the originator of the request specifies security parameters, the server uses these.
- If the originator of the request specifies an asterisk (*) for the security parameters, the server retrieves the security parameters and automatically inserts them into the request. If an asterisk is also specified for the responding system, the values retrieved for the requesting system are inserted for the responding system.
- If the originator of the request does not specify security parameters, the server tries to gain access to the file using its own authorization.
When processing a remote request (one created at another system), the local server does one of the following:

- If the originator of the request specifies security parameters, the server uses these.
- If the originator of the request does not specify security parameters, the server tries to gain access to the file using its own authorization.

It is normally advisable to use the default value, in order to ensure data security. NO can be used, for instance, if you are not using a System Authorization Facility (SAF) product, or if you have established your own data security provisions. For more information on data security, refer to “Data Security” on page 180.

**Note:** The BOTH option that was available in earlier releases of NetView FTP MVS has been removed to improve data security. Wherever appropriate replace BOTH with the YES or NO options.

**Events**

NVEVENT indicates if NetView FTP V2.2.1 MVS is to write operator terminal messages for the following events:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I/O error on the data set</td>
</tr>
<tr>
<td>2</td>
<td>VSAM error on the request queue, checkpoint data set, sending data set, or receiving data set</td>
</tr>
<tr>
<td>3</td>
<td>Abend during I/O</td>
</tr>
<tr>
<td>61</td>
<td>Server ready</td>
</tr>
<tr>
<td>62</td>
<td>Server stopped</td>
</tr>
<tr>
<td>63</td>
<td>Abnormal termination of a server</td>
</tr>
<tr>
<td>64</td>
<td>File transfer started</td>
</tr>
<tr>
<td>65</td>
<td>File transfer stopped</td>
</tr>
<tr>
<td>66</td>
<td>Security program error during user validation</td>
</tr>
<tr>
<td>67</td>
<td>Dynamic allocation error during file access.</td>
</tr>
</tbody>
</table>

These events are reported in such a way that they can be recognized and processed by NetView CLISTs. However, if there is no NetView CLIST active, the messages are sent to the operator console.

NVEVENT is an optional parameter.
Suppression of STOW Message

If you transfer a PDS data set that has a large number of members and you do not suppress the STOW messages, the log file and reports become very large. Use the SUPSTOW keyword to suppress the message DVG371I from the server transfer-log file.

YES  The message DVG371I is not printed to the server transfer-log file.

NO   The message DVG371I is printed to the server transfer-log file. This is the default.

| SUPSTOW= | YES | NO |

SMF Recording

SMFREC indicates for which SMF record SMF recording must be performed.

| SMFREC= | n |

n  Specifies the SMF record number under which the SMF recording takes place. You can specify a value from 000 to 255. If you do not specify a value, no SMF recording takes place. Values from 000 to 127 are reserved for MVS components or IBM subsystems. Values from 128 to 255 are reserved for other programs.

Internal Trace

Specifying a value for TRACE tells NetView FTP V2.2.1 MVS to take an internal trace while the server is running.

Use the internal trace for maintenance purposes only. Do not use it during normal production time, as it severely degrades the performance of NetView FTP V2.2.1 MVS.

If nothing is specified, no trace is taken.

GROUP

Specifies one of the server module groups

CL

Specifies a trace class.

Module groups and trace classes for server traces are described in detail in “Tracing an SNA Server” on page 140.
Example of Initialization Parameters for a Server

The following example shows sample initialization parameters for a server with LU name FTP01.

```
LUNAME=FTP01
DLOGMOD=FTBIND
BYTECR=50K
GID=DVG,CLASS=(1,3,9)
```

*Figure 62. Example of Initialization Parameters for a Server*

If this server is the receiving part of a transfer, it periodically takes a checkpoint after every 50KB have been received. If a name for a new receiving physical sequential data set or PDS must be created, this server uses the string DVG as the first qualifier. This server can process file-transfer requests with classes 9, 3, and 1. The entry (MODEENT) in the module table (MODTAB) uses the name FTPBIND.

Initialization Parameters for an OSI Server

The keywords for the initialization parameters for an OSI server are shown in Figure 63.

*Figure 63. OSI Server Initialization Parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Keyword</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSI Server Name</td>
<td>OISVNAME</td>
<td>125</td>
</tr>
<tr>
<td>VTAM LU name of OSI/File Services</td>
<td>OIFTAMID</td>
<td>125</td>
</tr>
<tr>
<td>VTAM LU Name for the Server</td>
<td>OIAAPPLID</td>
<td>125</td>
</tr>
<tr>
<td>Local User Name</td>
<td>OIUSERID</td>
<td>125</td>
</tr>
<tr>
<td>Local User Password</td>
<td>OIUSRPWD</td>
<td>126</td>
</tr>
<tr>
<td>New Local User Password</td>
<td>OINUSRPW</td>
<td>126</td>
</tr>
<tr>
<td>Server Running Mode</td>
<td>SRVMODE</td>
<td>125</td>
</tr>
<tr>
<td>Obtain Interval</td>
<td>OBTINTVL</td>
<td>126</td>
</tr>
<tr>
<td>Limit the Maximum Number of Simultaneous Requests</td>
<td>MAXSESS</td>
<td>126</td>
</tr>
<tr>
<td>Maximum Number of Active Transfer Requests</td>
<td>AVLSESS</td>
<td>126</td>
</tr>
<tr>
<td>Maximum Number of Locally Initiated Requests</td>
<td>AVLRRH</td>
<td>127</td>
</tr>
<tr>
<td>Maximum Number of Remotely Initiated Requests</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retry Delay Time</td>
<td>RETRYDELAY</td>
<td>127</td>
</tr>
<tr>
<td>Recovery Delay Time</td>
<td>OIRECDEL</td>
<td>127</td>
</tr>
<tr>
<td>Termination Time</td>
<td>TTIME</td>
<td>128</td>
</tr>
<tr>
<td>Server Classes</td>
<td>CLASS</td>
<td>128</td>
</tr>
<tr>
<td>Internal Trace</td>
<td>TRACE</td>
<td>128</td>
</tr>
</tbody>
</table>
Server Name
Use the OISVNAME keyword to specify the name of the OSI/File Services server as it is recorded in the queue handler. This name distinguishes between several OSI servers having the same class. You must specify a value for this parameter.

\textit{osiname} Can be up to 8 characters.

\begin{verbatim}
  OISVNAME=name
\end{verbatim}

VTAM LU Name of OSI/File Services (OSI/File Services LU Name)
Use the OIFTAMID keyword to define the APPC LU name of the OSI/File Services subsystem to VTAM.

\textit{identifier} Can be up to 8 characters.

\begin{verbatim}
  OIFTAMID=identifier
\end{verbatim}

VTAM LU Name of the Server (Local LU Name)
Use the OIAPPLID keyword to define the APPC LU name of the OSI server to VTAM.

\textit{luname} Can be up to 8 characters.

\begin{verbatim}
  OIAPPLID=luname
\end{verbatim}

SRVMODE
The OSI server always runs in CONTINUOUS mode, any other value is rejected.

\begin{verbatim}
  SRVMODE=C
\end{verbatim}

LRD User Name of OSI Server
Use the OIUSERID keyword to specify the local user name that is defined in the local resource directory (LRD). This defines the OSI server as a local user to OSI/File Services. This parameter corresponds to the OSI/File Services parameter LOCLUSID.

\textit{username} Can be up to 8 characters.

\begin{verbatim}
  OIUSERID=username
\end{verbatim}
Local User Password
Use the OIUSRPWD keyword to specify the local user password that is defined in the LRD for the OSI server. This parameter corresponds to the OSI/File Services parameter LOCLUOPW.

`password` Can be up to 8 characters.

```
>>>OIUSRPWD=password
```

New Local User Password
Use the OINUSRPW password to specify a new local-user password for the server. When you specify this parameter, the OSI server invokes the OSI/File Services function CHANGE USERPASSWORD. This parameter corresponds to the OSI/File Services parameter LOCLUNPW.

`npassword` Can be up to 8 characters.

```
>>>OINUSRPW=npassword
```

Obtain Interval
Use the OBTINTVL keyword to specify the time, in seconds, the OSI server must wait after unsuccessfully trying to obtain a request or unsuccessfully trying to get a result from OSI/File Services, before trying again. The value of this parameter is an integer from 1 to 9999.

```
>>>OBTINTVL=120
```

Limit to the Maximum Number of Simultaneous Requests
Use the MAXSESS keyword to specify the maximum number of requests that can be handled simultaneously. It is used to limit the amount of storage used to keep information about the requests until they are completed. The value of this parameter is an integer from 1 to 32.

```
>>>MAXSESS=16
```

Maximum Number of Active Transfer Requests
Use the AVLSESS keyword to limit the number of active requests that can be handled. This is dependent on the amount of requests that OSI/File Services can handle. To ensure that responder activities can be handled simultaneously and are not blocked by initiator activities, the number of available server sessions (FTAM requests) should be less than the maximum number of parallel FTAM sessions.

The value specified must be equal to or less than the value specified for MAXSESS. This can be modified by the operator.
Maximum Number of Locally Initiated Requests
As an OSI server can only handle locally initiated requests, the value specified for the AVLLRH keyword must be the same as AVLSESS.

Maximum Number of Remotely Initiated Requests
As an OSI server can only handle locally initiated requests, the value for the AVLRRH keyword must be set to zero.

Retry Delay Time
If a file transfer is interrupted, the OSI server writes the request back to the request queue and changes its status back to waiting. This way, the request is rescheduled later. The server also sets the request’s not-before date and time to the current time plus the number of seconds specified for the RETRYDELAY keyword. This forces the OSI server to wait at least that number of seconds before processing that request again. As long as the new not-before date and time has not passed, the request has the status waiting and is not dispatchable.

It can be any value from 1 to 9999. The default value is 300.

Note: The NetView FTP V2.1 MVS keyword REQDELAY is still accepted. However, you should use the keyword RETRYDELAY for newly setup servers.

Recovery Delay Time
Use the OIRECDEL keyword to specify the amount of time, in seconds, the OSI server tries to recover all the requests specified in the recovery file after an abnormal server termination.

During this time the server continues to try to get the requests that remain in the recovery file. The server returns any new requests, not in the recovery file, to the queue handler with status WAITING. Each request’s not-before date and time are set to the current time plus the number of seconds specified with the OIRECDEL keyword. After recovery is complete, the server cannot get the requests before the not-before date and time. The value of this parameter is an integer from 1 to 9999. The default is 300.
Termination Time
Use the TTIME keyword to specify the termination time of the server. When the specified time is reached, the server stops automatically. If a file transfer is in progress when a server's termination time is reached, the server waits until the file transfer is finished before terminating.

**hh:mm** The time in 24-hour format. For example, 3:23 pm is specified as 15:23.

**yy/mm/dd** The date in year-month-day format. For example, 11 March 1994 is specified as 94/03/11. The default is the current date.

If no value for this parameter is specified, the server runs indefinitely.

**Server Class**
Use the CLASS keyword to assign specific classes that are used exclusively for OSI transfers.

**Internal Trace**
Specifying a value for TRACE tells NetView FTP V2.2.1 MVS to take an internal trace while the server is running.

Use the internal trace for maintenance purposes only. Do not use it during normal production time, as it severely degrades the performance of NetView FTP V2.2.1 MVS.

If nothing is specified, no trace is taken.

**GROUP** Specifies one of the server module groups

**CL** Specifies a trace class.

Module groups and trace classes for server traces are described in detail in “Tracing an SNA Server” on page 140.
Chapter 11. Changing the Text of NetView FTP V2.2.1 MVS Messages and the Language They Are Issued In

The messages issued by NetView FTP V2.2.1 MVS are located in a macro that is written in Assembler. You can change the text of the NetView FTP V2.2.1 MVS messages to meet your requirements.

To change the messages or to create your own messages, do the following:

1. Copy the macro DVG$GMC, which is stored in the macro library DVG.NFTP230.SDVGMAC0. Either copy DVG$GMC into another library or copy it into the same library and give it a different name.

2. If you are only changing the English messages and not translating them into another language, skip to step 4.

3. In your copy of DVG$GMC, find the label MSGDEF. This label marks the beginning of the messages. In the AIF branch table that follows the message definitions, find the label that corresponds to the language you are using. Copy the original English messages below that label.

4. Change the messages in your copy of DVG$GMC. Follow the rules given in “Rules to Follow when Changing the Message Texts” on page 130. If you are translating the messages into a language that uses double-byte characters, refer to “Special Considerations for Languages That Require Double-Byte Characters” on page 132.

5. Change the value for the &LANG parameter in the prototype statement for your copy of DVG$GMC to the first letter of the language of your messages.

6. Put your copy of DVG$GMC back into the correct library.

7. Create a load module with the name DVGCCMDx, where x is the first letter of one of the following languages:
   - English (U.S.)
   - French
   - German
   - Italian
   - Japanese
   - Portuguese
   - Spanish
   - User (any language requiring single-byte encoding)
   - Double (any language requiring double-byte encoding).
   
This load module is your message module.
The library **DVG.NFTP230.SDVGSAM0** contains a module that you can use as a sample. The sample has the name DVGCXAMD and is shown in Figure 64.

```plaintext
TITLE 'DVGCXAMD - SAMPLE MESSAGE MODULE'
/*****************************************************************************/
/** MODULE-NAME = DVGCXAMD */
/** DESCRIPTIVE-NAME = MESSAGE MODULE (EXAMPLE) */
/** 5685-108 (C) COPYRIGHT IBM CORP. 1991, 1994 */
/** LICENSED MATERIALS - PROPERTY OF IBM */
/** SEE COPYRIGHT INSTRUCTIONS, G120-2083 */
/** STATUS = NETVIEW FTP V2 MVS REL.2.1 */
.TODO DMGMC SECTYPE=CSECT,SECTNAM=DVGCCMDU
SPACE 3
END
```

**Figure 64. Module DVGCXAMD—Sample Message Module**

In your message module:

- The macro name must be the name you gave to your copy of DVG$GMC.
- The value of SECTYPE must be CSECT.
- The value of SECTNAM must be the same as the name of the load module.

8. Assemble your message module.

9. Link-edit your message module using the same load module name that you specified for the SECTNAM operand.

**Rules to Follow when Changing the Message Texts**

- Do not use the character @ when changing messages. This character is used by NetView FTP V2.2.1 MVS to indicate where in the message the variable information is to be inserted.
- You can change the fixed portion of the text, even the length of the fixed portion.
- You can change the relative positions of the variables. The number of variables, however, must remain the same.
- All messages in the DVG$GMC macro must be written in uppercase letters.
- A variable or blank must appear every 38 bytes.
- A variable is denoted by the string &VP.nn, where nn is a 2-digit number. Generally the numbers start with 01 and are incremented by 1.
- A variable must be preceded by a blank unless the variable is the first item in the message.
- A variable must appear in the first 255 bytes.
- The variables referred to in a changed message must be the same as those referred to in the original message.
- The value of a variable is set by the module that issues the message. You cannot set the value of a variable.
• Only the following characters can appear after a variable:
  
  Blank
  , Comma
  ; Semicolon
  . Period
  ) Right parenthesis
  ( Left parenthesis
  : Colon
  ' Single quote
  = Equal sign
  - Hyphen
  Shift-out (double-byte character encoding only).

• Do not change or translate the following messages:
  
  DVG200I
  DVG201I
  DVG202I
  DVG220I
  DVG221I
  DVG299I
  DVG400I
  DVG401I
  DVG408A
  DVG410I
  DVG670I.

• If a message is longer than 1024 bytes, NetView FTP V2.2.1 MVS truncates it.

• Do not change the message number.

• A message type must be one character. NetView FTP V2.2.1 MVS uses the following message types:
  
  W Warning messages
  I Informational messages
  A Messages requiring operator action.

• Change only the message part of your copy of DVG$GMC and not any internal macrodefinitions.
Special Considerations for Languages That Require Double-Byte Characters

Take the following into consideration when using languages that require double-byte characters:

- You do not have to change the macro prototype statement unless you want to change the shift-out or shift-in characters. These are only needed when your messages contain double-byte character strings.
  - &STARTSK defines the shift-out character. The default is X'0E'.
  - &ENDSK defines the shift-in character. The default is X'0F'.

- If your messages contain double-byte character strings and if you want to issue a blank character, do not use the double-byte blank, but instead use a combination of shift-in and shift-out. This is because NetView FTP V2.2.1 MVS blank suppression reduces the double-byte blank to a single blank and produces unreadable output.

- Do not have more than 34 double-byte characters between one shift-out/shift-in pair, otherwise, NetView FTP V2.2.1 MVS’s message segmentation cannot work.

- If your double-byte character messages contain X'7D' or X'50', double these characters to ensure an error-free run of Assembler when you create your message module.

- Variables must not appear in double-byte character strings marked by shift-out and shift-in.

- To create a message module for a language that uses double-byte characters, Assembler H Version 2 Release 1 must be installed on your system.
Part 5. Problem Determination

This part describes how to take traces, what to do in the case of abnormal termination, and how to find and report program problems.
Chapter 12. Taking Traces

This chapter contains Diagnosis, Modification, or Tuning Information.

Taking traces allows you to:

- Document program errors for IBM Support Centers, so that the IBM Support Center personnel know:
  - The flow of modules within the NetView FTP V2.2.1 MVS component where you started the trace
  - How the main storage areas, which these modules accessed, are changed during the execution of the modules
  - The state of a specific storage area at a given time while NetView FTP V2.2.1 MVS is processing a file-transfer request.

An IBM software engineer will tell you how to invoke a trace and what you should trace.

- Test the interface between NetView FTP V2.2.1 MVS and user-written exit routines. The trace facility documents contain the following:
  - Which control information NetView FTP V2.2.1 MVS passes to the user-written exit routine
  - How the exit routine changes the control information
  - The state, at a given point in time, of a specific storage area that is defined by the exit routine.

You can trace the queue handler and an SNA server or OSI server.

When you run a trace, NetView FTP V2.2.1 MVS writes dumps of storage areas to the data set with the ddname DVGSNAP. NetView FTP V2.2.1 MVS uses the SNAP macroinstruction to write these dumps. Certain server traces are written to the log file.

For further information on the SNAP macroinstruction, refer to MVS/Extended Architecture Supervisor Services and Macro Instructions.

NetView FTP V2.2.1 MVS begins writing a dump at defined trace points in the modules that support the trace facility. Each storage dump is accompanied by the symptom record that describes the corresponding dump. The layout of the symptom record is shown in Figure 82 on page 148. This only applies in those NetView FTP V2.2.1 MVS modules that support the trace facility.

When to Take Traces

Take traces only to help you to solve severe problems that you cannot solve in other ways, for example, to help you identify a NetView FTP V2.2.1 MVS module that is not working properly or to help you test your user-exit routines. Do not use it during normal production time, as it can severely degrade performance.
Types of Traces

There are three different types of traces:

- Entry and exit traces, which have class 1.
- Specific module traces, which have classes 2 to 8.
- Patch-area traces, which require temporary code changes and are used only by IBM Support Centers.

Entry and Exit Trace

An entry and exit trace is a class-1 NetView FTP V2.2.1 MVS internal trace. With the entry and exit trace you can follow the module flow in the queue handler, an SNA server, or an OSI server. Each time a module is entered or exited, NetView FTP V2.2.1 MVS dumps the following storage areas:

- The NetView FTP V2.2.1 MVS symptom record. The symptom record indicates that NetView FTP V2.2.1 MVS has taken an entry or exit trace. The message in the symptom record shows the module that is entered or exited. The message also displays the return code of the module when it is exited.
- The register contents at the time when the module is entered or exited.

Figure 65 on page 138, Figure 66 on page 138, and Figure 67 on page 139 show those modules of the queue handler, of an SNA server, and of an OSI server, respectively, for which entry and exit traces are available.

Specific Module Traces

In some modules you can also take a specific module trace. Classes 2 to 8 of NetView FTP V2.2.1 MVS internal traces are specific module traces. With a specific module trace, you can follow the logic flow inside those modules of a server or the queue handler that are equipped with special trace points and you can check the contents of specific storage areas.

At some trace points, NetView FTP V2 MVS produces a dump of storage areas that have specific significance. The output of a specific module trace for each trace point consists of the following items:

- The NetView FTP V2.2.1 MVS symptom record. The symptom record indicates that NetView FTP V2.2.1 MVS has taken a trace of a specific module. The message in the symptom record indicates that point of the module where NetView FTP V2.2.1 MVS takes the trace and identifies the storage area that is dumped.
- The module’s current registers.
- The dump of a storage area that is of particular interest at the time when NetView FTP V2.2.1 MVS takes the trace of the specific module. Some trace points do not produce this storage area dump. The specific trace point determines the particular storage area that NetView FTP V2.2.1 MVS dumps.

Specific module traces are available for queue handler, SNA server, and OSI server. These traces are described in:

- “Tracing the Queue Handler” on page 139
- “Tracing an SNA Server” on page 140
- “Tracing an OSI Server” on page 143.
Activating the Trace Facility

You activate the NetView FTP V2.2.1 MVS trace facility by specifying a module group and one or more trace classes for the Trace initialization parameter. You can code the Trace parameter more than once. You suppress tracing by omitting the Trace initialization parameter. This parameter is available for each NetView FTP server and the queue handler.

For the queue handler, you can also start, modify, or stop a trace with the MODIFY operator command. For further information, refer to “Starting and Stopping an Internal Trace” on page 60.

The queue handler modules are allocated to the six groups as shown in Figure 65.

The following example shows how you would set the trace to perform an entry and exit trace on the queue-handler modules that are related to operator commands:

```
TRACE=(GROUP1,CL1)
```

The modules of an SNA server are allocated to the ten groups as shown in Figure 66.

The following example shows how you would set up the trace to perform a trace of the interface block in the module that connects the user-written file handler.

```
TRACE=(GROUP1,CL2,CL3,CL4,CL5)
```

The modules of an OSI server are allocated to the groups shown in Figure 67 on page 139.

The output produced by the trace function can be very large. Trace output files, therefore, can overflow if they are not large enough. To reduce the output and to get a readable trace it can be necessary to select specific modules to be traced.

Starting and Stopping a Trace

Specify the trace setup with the Trace initialization parameter, which is described in:

- “Specifying Values for the Queue Handler Initialization Parameters” on page 99 for the queue handler
- “Specifying Values for the Server Initialization Parameters” on page 113 for the SNA server
- “Initialization Parameters for an OSI Server” on page 124 for the OSI server.

The queue handler or the server reads the trace setup at its startup time. The trace in a server remains in effect until you stop the server. You can stop a trace in a queue handler by using the MODIFY operator command or by stopping the queue handler.

You should not invoke the trace facility on those servers that are part of your main NetView FTP V2.2.1 MVS production system. Instead, reserve one server for tracing purposes. Start the server that you want to trace with the appropriate Trace parameter.
Module Groups for Traces

The figures show the module groups for a queue handler, an SNA server, and an OSI server.

Queue-Handler Module Groups

The queue-handler modules are allocated to the six groups in the order shown in Figure 65.

<table>
<thead>
<tr>
<th>Group</th>
<th>Group Classification</th>
<th>Assigned Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modules related to operator commands</td>
<td>DVGMOQPI, DVGMQCOM, DVGCQQB, DVGMQDIS</td>
</tr>
<tr>
<td>2</td>
<td>Function handler (subtask main module)</td>
<td>DVGMQFUN</td>
</tr>
<tr>
<td>3</td>
<td>QH commands issued by interface modules</td>
<td>DVGCQOND, DVGCQORY, DVGCQADD, DVGCQDEL, DVGCQMFY, DVGCQFOR, DVGCQRES</td>
</tr>
<tr>
<td>4</td>
<td>QH commands issued by server modules</td>
<td>DVGCQOPN, DVGCQCLS, DVGCQSTA, DVGCQNOT, DVGCQOBT, DVGCQGSS, DVGMQCHK</td>
</tr>
<tr>
<td>5</td>
<td>Request-queue file handler</td>
<td>DVGCQTRQ</td>
</tr>
<tr>
<td>6</td>
<td>SMF recording</td>
<td>DVGMQSMF</td>
</tr>
</tbody>
</table>

1 Entry and exit tracing does not exist for module DVGMQFUN, but a specific trace is available.

SNA-Server Module Groups

The SNA server modules are allocated to the ten groups in the order shown in Figure 66.

<table>
<thead>
<tr>
<th>Group</th>
<th>Group Classification</th>
<th>Assigned Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User-written file handler</td>
<td>DVGCSFR, DVGCSTCM</td>
</tr>
<tr>
<td>2</td>
<td>File transfer manager</td>
<td>DVGSCSF, DVGCSTCM</td>
</tr>
<tr>
<td>3</td>
<td>File handler modules</td>
<td>DVGCSCDC, DVGMSVE, DVGCSVK, DVGMSSM, DVGSMPO, DVGMSBLK, DVGMSTTR, DVGMSDH, DVGMSBP, DVGMSDLDS, DVGSCVR</td>
</tr>
<tr>
<td>4</td>
<td>Communication handling (open / close)</td>
<td>DVGMSSMD, DVGMSMTI, DVGMSMPF, DVGMSTPS, DVGSCFCN</td>
</tr>
<tr>
<td>5</td>
<td>VTAM handling modules</td>
<td>DVGCSAP</td>
</tr>
<tr>
<td>6</td>
<td>Generalized data stream parser</td>
<td>DVGCSP1, DVGCSPB9, DVGCSPD9, DVGCSDML, DVGCSPM1, DVGCSPB2, DVGCSPD7</td>
</tr>
<tr>
<td>7</td>
<td>Server main task modules</td>
<td>DVGMSFT, DVGMSIN, DVGMSNO, DVGCSWSM</td>
</tr>
<tr>
<td>8</td>
<td>User-exit routine interfaces / job submission / post-transfer program call</td>
<td>DVGCSPP, DVGMSPC, DVGMSXDM, DVGMSJOB, DVGMSPMG</td>
</tr>
<tr>
<td>9</td>
<td>File allocation</td>
<td>DVGMSDN, DVGMSFC, DVGSCDC, DVGMSDNA, DVGCSDCN, DVGMSDN, DVGMSDNT, DVGCSFCX</td>
</tr>
<tr>
<td>10</td>
<td>Access control and data security</td>
<td>DVGMSAC, DVGMSBC, DVGSCCK</td>
</tr>
</tbody>
</table>

1 Entry and exit tracing does not exist, but specific module traces are available.
OSI-Server Module Groups
The OSI server modules are allocated to the ten groups in the order shown in Figure 67.

<table>
<thead>
<tr>
<th>Group</th>
<th>Group Classification</th>
<th>Assigned Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control Module</td>
<td>DVGMSOCM</td>
</tr>
<tr>
<td>2</td>
<td>Initialization Module</td>
<td>DVGMSOIN</td>
</tr>
<tr>
<td>3</td>
<td>Termination Module</td>
<td>DVGMSOCU</td>
</tr>
<tr>
<td>4</td>
<td>Queue Handler Interface Module</td>
<td>DVGMSOQH</td>
</tr>
<tr>
<td>5</td>
<td>Parameter Transformation Module</td>
<td>DVGMSOCP</td>
</tr>
<tr>
<td>6</td>
<td>LRD Handling Module</td>
<td>DVGMSOLR</td>
</tr>
<tr>
<td>7</td>
<td>OSI/File Services Interface Handling Module</td>
<td>DVGMSOFS</td>
</tr>
<tr>
<td>8</td>
<td>Interrupt Handling Module</td>
<td>DVGMSOIR</td>
</tr>
<tr>
<td>9</td>
<td>General Utility Module</td>
<td>DVGMSGU</td>
</tr>
<tr>
<td>10</td>
<td>Error Handling Module</td>
<td>DVGMSOER</td>
</tr>
</tbody>
</table>

Tracing the Queue Handler
Figures 68, 69, and 70 show the following:
- The modules of the queue handler for which specific module traces are available
- The data that NetView FTP V2.2.1 MVS dumps for each module
- The trace classes for the specific module groups.

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGMOFUN</td>
<td>The internal copy of FSB/FSBX before and after processing of QH commands.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGCSQADD</td>
<td>The request control block and, if available, the request control block extensions before a request is added to the request queue.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGCQNOT</td>
<td>The request control block.</td>
</tr>
<tr>
<td>4</td>
<td>DVGCQOBT</td>
<td>The request control block.</td>
</tr>
</tbody>
</table>
Tracing an SNA Server

You can run an entry and exit trace in most modules of a server.

The following figures show:
- The modules of an SNA server where specific module traces are available
- The data that NetView FTP V2.2.1 MVS dumps for each module
- The trace classes for the specific module groups.

**Figure 71. User-Written File Handler (Group 1)**

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGCSUR</td>
<td>The user-written file handler parameter list (UPL) before or after calling the exit routine for function GET, PUT, or POINT.</td>
</tr>
<tr>
<td>3</td>
<td>DVGCSUR</td>
<td>The storage area defined by the exit routine after calling the exit routine for function GET, PUT, or POINT.</td>
</tr>
<tr>
<td>4</td>
<td>DVGCSUR</td>
<td>The user-written file handler parameter list (UPL) before or after calling the exit routine for function OPEN or CLOSE.</td>
</tr>
<tr>
<td>5</td>
<td>DVGCSUR</td>
<td>The storage area specified by a user-written exit routine after it has been called for function OPEN or CLOSE.</td>
</tr>
</tbody>
</table>

**Note:** The module DVGCSUR establishes the interface between NetView FTP V2.2.1 MVS and a user-written file handler. You can use the trace facility when testing this user-exit routine. Refer to the NetView FTP Customization guide.

**Figure 72. File-Transfer Manager (Group 2)**

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGSTCM</td>
<td>Protocol data elements.</td>
</tr>
<tr>
<td>4</td>
<td>DVGSTCM</td>
<td>Parameters passed to this module from the error handling routines.</td>
</tr>
<tr>
<td>5</td>
<td>DVGSCRH</td>
<td>The checkpoint record.</td>
</tr>
</tbody>
</table>

**Figure 73. File-Transfer Manager (Group 3)**

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGCCDC</td>
<td>The data-conversion parameter list.</td>
</tr>
</tbody>
</table>

**Figure 74. VTAM Handling Modules (Group 5)**

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGCSAP</td>
<td>Internal control block APC at entry to and exit from this module.</td>
</tr>
<tr>
<td>3</td>
<td>DVGCSAP</td>
<td>The work areas and the RPL and RPL6 for the VTAM requests Display, CNOS, ALLOC, Send_CONF, and RCVFMH5.</td>
</tr>
<tr>
<td>Trace Class</td>
<td>Module</td>
<td>Data Traced</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>1 (PCC)</td>
<td>DVGCSPI</td>
<td>Before ... Dump of PCC before GDS parser invocation.</td>
</tr>
<tr>
<td>1 (PCC)</td>
<td>DVGCSPI</td>
<td>After ... Dump of PCC after GDS parser invocation.</td>
</tr>
<tr>
<td>1 (001)</td>
<td>DVGCSPE9</td>
<td>Init - on Encoding server; what is dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s Request Parameter List (RPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - GDS parser’s Encode Output Area (DWKA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - GDS parser’s parse table of the implemented NFTP protocol.</td>
</tr>
<tr>
<td>1 (001)</td>
<td>DVGCSPD9</td>
<td>Init - on Decoding server; what is dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s Request Parameter List (RPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - GDS parser’s Decode I/O look-aside buffer (DWKA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - GDS parser’s parse table of the implemented NFTP protocol.</td>
</tr>
<tr>
<td>1 (004)</td>
<td>DVGCSPE9</td>
<td>Return - on Encoding server; what is dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Request Parameter List (RPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - GDS parser’s Encode Output Area (DWKA).</td>
</tr>
<tr>
<td>1 (004)</td>
<td>DVGCSPD9</td>
<td>Return - on Decoding server; what is dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Request Parameter List (RPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - GDS parser’s Item Decode Output Area (DITMA).</td>
</tr>
<tr>
<td>1 (007)</td>
<td>DVGCSPE9</td>
<td>Termination ... snap of GDS parser’s internal Main Control Block (DABCB).</td>
</tr>
<tr>
<td>1 (007)</td>
<td>DVGCSPD9</td>
<td>Termination ... snap of GDS parser’s internal Main Control Block (DABCB).</td>
</tr>
<tr>
<td>2 (002)</td>
<td>DVGCSPE9</td>
<td>Encoding server; what is dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Item Encode Work Area (DSTGA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - GDS parser’s Encode Output Area (DWKA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - GDS parser’s Stack</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - GDS parser’s output buffer.</td>
</tr>
<tr>
<td>2 (002)</td>
<td>DVGCSPD9</td>
<td>Decoding server; what is dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s Request Parameter List (RPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - GDS parser’s Dynamic Save Area (DSTGA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - GDS parser’s Decode I/O look-aside buffer (DWKA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - GDS parser’s Stack</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 - GDS parser’s input buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7 - GDS parser’s parse table of the implemented NFTP protocol.</td>
</tr>
<tr>
<td>3 (003)</td>
<td>DVGCSPE9</td>
<td>Dump of I/O buffers.</td>
</tr>
<tr>
<td>3 (003)</td>
<td>DVGCSPD7</td>
<td>Dump of I/O buffers.</td>
</tr>
<tr>
<td>3 (003)</td>
<td>DVGCSPD9</td>
<td>Dump of I/O buffers.</td>
</tr>
<tr>
<td>4</td>
<td>DVGCSPSV</td>
<td>Formatted message units are written to the server’s log file and report file. See the example shown in Figure 76 for data traced.</td>
</tr>
</tbody>
</table>

Note: The values in parentheses after the trace class are the internal IDs for General Data Stream (GDS) parser trace. See the example shown in Figure 82 on page 148.
Figure 75 (Part 2 of 2). LU 6.2 Conversations (Group 6)

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Data Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 (005)</td>
<td>DVGCSPE9</td>
<td>Push ... Dump of GDS parser stack for each new group of the parse table:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Information dumped on encode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Stack.</td>
</tr>
<tr>
<td>5 (005)</td>
<td>DVGSPD9</td>
<td>- Information dumped on decode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Stack.</td>
</tr>
<tr>
<td>5 (006)</td>
<td>DVGCSPE9</td>
<td>Pop ... Dump of GDS parser stack for each return to a lower level of the parse table:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Information dumped on encode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Stack.</td>
</tr>
<tr>
<td>5 (006)</td>
<td>DVGCPD9</td>
<td>- Information dumped on decode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Stack.</td>
</tr>
<tr>
<td>5 (008)</td>
<td>DVGCSPE9</td>
<td>Repeating Group ... a new repeating group has been started in the parse table; information dumped:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - GDS parser’s internal Main Control Block (DABCB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - GDS parser’s Stack.</td>
</tr>
<tr>
<td>5 (008)</td>
<td>DVGCPD9</td>
<td>Parse table descriptor ... each new element in the parse table; information dumped: parse table descriptor.</td>
</tr>
<tr>
<td>5 (DML)</td>
<td>DVGCSMML</td>
<td>Data modification layer control block is dumped before / after compression / decompression encryption / decryption.</td>
</tr>
</tbody>
</table>

Figure 76. Example of a Trace for Group 6, Class 4, Written to the Server’s Log and Report File

DVG842I NAME = TIMU_prefix ; ID = 1B1B1B1B; LENGTH = 0 ; VALUE =  
DVG842I NAME = U usage_disp_1 ; ID = 1B1B1B1B001; LENGTH = 1 ; VALUE = D6  
DVG842I NAME = U usage_disp_3 ; ID = 1B1B1B1B003; LENGTH = 1 ; VALUE = C5  
DVG842I NAME = T Program_ID ; ID = 1B1B1B1B1; LENGTH = 5 ; VALUE = AB01  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 3 ; VALUE = ABC  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 3 ; VALUE = ZAW  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 5 ; VALUE = DUMMY  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 2 ; VALUE = PO  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 2 ; VALUE = FB  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 4 ; VALUE = LI12  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 5 ; VALUE = 84680  
DVG842I NAME = Simple_name ; ID = 1B1B1B1B1718217DA01; LENGTH = 5 ; VALUE = MEM10  
DVG842I NAME = T file_type ; ID = 1B1B1B1B1718217DA01; LENGTH = 1 ; VALUE = D4  
DVG842I NAME = TIMU_suffix ; ID = 1B1B1B36; LENGTH = 0 ; VALUE =  

Note: VALUE is truncated if it exceeds 100 bytes.

Figure 77. Server Main Task (Group 7)

<table>
<thead>
<tr>
<th>Trace Class</th>
<th>Module</th>
<th>Trace Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DVGMSFT</td>
<td>The request control block and, if available, the request control block extensions as obtained from the queue handler.</td>
</tr>
<tr>
<td>3</td>
<td>DVGMSIN</td>
<td>The VTAM resource vector list.</td>
</tr>
</tbody>
</table>
Tracing an OSI Server

Classes 1 and 2 of NetView FTP V2.2.1 MVS internal traces are specific module traces. At some trace points, NetView FTP V2.2.1 MVS produces a dump of storage areas that have specific significance. For a list of modules and the information that is traced, see Figure 81.
<table>
<thead>
<tr>
<th>Trace Group</th>
<th>Assigned Module(s)</th>
<th>Entry or Exit</th>
<th>Specific Module Trace Classes 1 and 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Initialization Module DVGMSOIN, Recovery Module DVGMSORC</td>
<td>G_Ind_Act_Req L_Req_Table (L_REQTAB) G_Ser_Str (GSS) SMS (SMA) Storage_Tab (STORAGE)</td>
<td>G_Ind_Act_Req L_Req_Table (L_REQTAB) G_Ser_Str (GSS) L_Request (L_REQ) RCB RXB CCS (CGB) STAE (STAENTR) COS (CCI) Storage_Tab (STORAGE)</td>
</tr>
<tr>
<td>3</td>
<td>Termination Module DVGMSOCU</td>
<td>G_Ser_Str (GSS) L_Req_Table (L_REQTAB)</td>
<td>G_Ser_Str (GSS) L_Request (L_REQ) CCS (CGB)</td>
</tr>
<tr>
<td>4</td>
<td>Queue Handler Interface Module DVGMSOQH</td>
<td>G_Ser_Str (GSS) L_Request (L_REQ) L_Request_Table (L_REQTAB) RSE (RCE) CCS (CGB) SMS (SMA) STAE (STAAREA) Storage_Tab (STORAGE)</td>
<td>COS (CCI) CCS (CGB)</td>
</tr>
<tr>
<td>5</td>
<td>Parameter Transformation Module DVGMSOCP</td>
<td>G_Ser_Str (GSS) G_Ind_Act_Req L_Req_Table (L_REQTAB) L_Request (L_REQ) RSE (RCE) STAE (STAENTR) Storage_Tab (STORAGE) RXB</td>
<td>G_Ind_Act_Req L_Req_Table (L_REQTAB) L_Request (L_REQ) RXB Storage_Tab (STORAGE) O_LRD</td>
</tr>
<tr>
<td>6</td>
<td>LRD Handling Module DVGMSOLR</td>
<td>G_Ser_Str (GSS) L_Request (L_REQ) G_Ind_Act_Req L_Req_Table (L_REQTAB) RSE (RCE) O_LRD RXB</td>
<td>O_LRD RXB</td>
</tr>
<tr>
<td>7</td>
<td>OSI/File Services Interface Handling Module DVGMSOFS</td>
<td>G_Ind_Act_Req L_Req_Table (L_REQTAB) G_Ser_Str (GSS) L_Request (L_REQ) RSE (RCE) STAE (STAENTR)</td>
<td>G_Ser_Str (GSS) L_Request (L_REQ) O_MAC</td>
</tr>
<tr>
<td>8</td>
<td>Interrupt Handling Module DVGMSOIR</td>
<td>G_Ind_Act_Req L_Req_Table (L_REQTAB) G_Ser_Str (GSS) SMS (SMA) CCS (CGB)</td>
<td>G_Ser_Str (GSS) L_Request (L_REQ) CCS (CGB) SMS</td>
</tr>
<tr>
<td>9</td>
<td>General Utility Module DVGMSOGU</td>
<td>G_Ind_Act_Req L_Req_Table (L_REQTAB) G_Ser_Str (GSS) Storage_Tab (STORAGE)</td>
<td>L_Req_Table (L_REQTAB) Storage_Tab (STORAGE) G_Ser_Str (GSS)</td>
</tr>
<tr>
<td>10</td>
<td>Error Handling Module DVGMOER</td>
<td>G_Ser_Str (GSS) L_Req_Table (L_REQTAB) G_Ind_Act_Req</td>
<td>G_Ser_Str (GSS) L_Request (L_REQ) CCS (CGB)</td>
</tr>
</tbody>
</table>
SNA Server and Queue Handler Trace Output

When NetView FTP V2.2.1 MVS takes a trace, it produces the following output for each trace point in the NetView FTP V2.2.1 MVS modules:

- A NetView FTP V2.2.1 MVS symptom record
- The contents of the general registers at the time the trace is taken
- A storage area dump.

The area that NetView FTP V2.2.1 MVS dumps depends on the module where it takes the trace and the class of the trace.

The NetView FTP V2.2.1 MVS Symptom Record as Trace Output

Figure 82 on page 148 shows an example of a NetView FTP V2.2.1 MVS symptom record as NetView FTP V2.2.1 MVS writes it for a GDS parser trace.

COMP ID
The NetView FTP V2.2.1 MVS component ID.

FMID
The NetView FTP V2.2.1 MVS function modification identifier.

VER REL MOD
Version, release, and modification level.

ERROR/TRACE TYPE
One of the following for a normal trace:

- ENTR Entry trace
- EXIT Exit trace
- TRCE Specific module traces.

One of the following for an abend:

- PCHK Exit trace
- N/A All other kinds of error traces.

See “Types of Traces” on page 136 for an explanation of these trace types.

SYS CODE
Not applicable.

USR CODE
The number of the related trace message (see Message Area on page 146).

LOAD MODULE NAME
If NetView FTP V2.2.1 MVS cannot retrieve the load module name, it inserts the character string “UNKNOWN.”

CSECT NAME
Name of the NetView FTP V2.2.1 MVS CSECT (program module) where NetView FTP V2.2.1 MVS took the trace.

CCI ADDRESS
Content of general register 12, which generally points to the NetView FTP V2.2.1 MVS Common Control Block Insertion (CCI).
TASK ID
Identifier for the main task or subtask of the NetView FTP V2.2.1 MVS component where NetView FTP V2.2.1 MVS took the trace:

00 Main task of the queue handler or a server.
01-nn Subtask of the queue handler or a server.

The subtasks of the queue handler are the processors of the queue handler commands. The subtasks of the server perform the file transfers.

SAVE AREA ADDR
Address of a 72-byte area that contains the general registers at the time NetView FTP V2.2.1 MVS issued the trace. This register save area appears in a separate dump that comes after the dump of the NetView FTP V2.2.1 MVS symptom record. See “Register Dump as Tracing Output” on page 147.

SDWA ADDRESS
Not applicable.

Message Area
The NetView FTP message in the symptom record indicates whether the event that caused the dump was:

- An ABEND
- A program error
- An internal trace.

The message area contains the message that relates to this trace:

- DVG200I patch area trace
- DVG201I entry trace
- DVG202I exit trace
- DVG220I user-exit trace
- DVG221I user-exit trace
- DVG020I specific module trace.

This message contains a string constant that describes the trace event.
Register Dump as Tracing Output

Figure 82 shows an example of a trace dump. NetView FTP V2.2.1 MVS issued this dump during a GDS parser trace in the server module DVGCSPE9.

When NetView FTP V2.2.1 MVS takes a trace in a module, it dumps the general register save area of that module. The save area has the following fullwords:

<table>
<thead>
<tr>
<th>Fullword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>Address of calling module’s save area</td>
</tr>
<tr>
<td>3</td>
<td>Address of called module’s save area</td>
</tr>
<tr>
<td>4</td>
<td>General Register 14</td>
</tr>
<tr>
<td>5</td>
<td>General Register 15</td>
</tr>
<tr>
<td>6</td>
<td>General Register 0</td>
</tr>
<tr>
<td>7</td>
<td>General Register 1</td>
</tr>
<tr>
<td>8</td>
<td>General Register 2</td>
</tr>
<tr>
<td>9</td>
<td>General Register 3</td>
</tr>
<tr>
<td>10</td>
<td>General Register 4</td>
</tr>
<tr>
<td>11</td>
<td>General Register 5</td>
</tr>
<tr>
<td>12</td>
<td>General Register 6</td>
</tr>
<tr>
<td>13</td>
<td>General Register 7</td>
</tr>
<tr>
<td>14</td>
<td>General Register 8</td>
</tr>
<tr>
<td>15</td>
<td>General Register 9</td>
</tr>
<tr>
<td>16</td>
<td>General Register 10</td>
</tr>
<tr>
<td>17</td>
<td>General Register 11</td>
</tr>
<tr>
<td>18</td>
<td>General Register 12.</td>
</tr>
</tbody>
</table>
Figure 82. Example of a NetView FTP Module Trace (Class 1 - Termination)

1 Symptom record
2 General register save area dump
3 Trace data.
Amount of Output Produced during Tracing

Each trace requires about six to seven pages of paper per trace point. The total amount of output that NetView FTP V2.2.1 MVS produces during tracing depends on:

- The number of commands that the queue handler processes while the trace is active
- The number of records in the request queue
- The duration of the trace
- The number of file transfers that a server performs while the trace is active
- The number of records that a server transfers during a file transfer
- The number of RUs or VTAM I/O buffers that a server sends or receives during a file transfer.

Frequency of Entering and Exiting the Queue Handler Modules

The following figures show how often a module is entered and exited during normal processing of the queue handler.

**Figure 83. Frequency of Entering and Exiting Modules Related to Operator Commands (Group 1)**

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMQOPI</td>
<td>Once per operator command</td>
</tr>
<tr>
<td>DVGMQCOM</td>
<td>Once per queue handler command</td>
</tr>
<tr>
<td>DVGCQQB</td>
<td>Once per request queue rebuild</td>
</tr>
<tr>
<td>DVGMQDIS</td>
<td>Once per DISPLAY command</td>
</tr>
</tbody>
</table>

**Figure 84. Frequency of Entering andExiting Function Handler (Subtask Main Module) (Group 2)**

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMQFUN</td>
<td>Once for each FSBSLOT in the queue handler's run time</td>
</tr>
</tbody>
</table>

**Figure 85. Frequency of Entering and Exiting Modules Related to Commands (Group 3)**

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCQQND</td>
<td>Once per GETOPSYS queue handler command</td>
</tr>
<tr>
<td>DVGCQQRY</td>
<td>Once per QUERY queue handler command</td>
</tr>
<tr>
<td>DVGCOADD</td>
<td>Once per ADD queue handler command</td>
</tr>
<tr>
<td>DVGCQDEL</td>
<td>Once per DELETE queue handler command</td>
</tr>
<tr>
<td>DVGCMQFY</td>
<td>Once per MODIFY queue handler command</td>
</tr>
</tbody>
</table>

**Figure 86. Frequency of Entering and Exiting Server Modules Related to Commands (Group 4)**

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCQQOPN</td>
<td>Once per OPEN queue handler command</td>
</tr>
<tr>
<td>DVGCQCLS</td>
<td>Once per CLOSE queue handler command</td>
</tr>
<tr>
<td>DVGCQSTA</td>
<td>Once per STATUS queue handler command</td>
</tr>
<tr>
<td>DVGCQN0T</td>
<td>Once per NOTIFY queue handler command</td>
</tr>
<tr>
<td>DVGCQOBT</td>
<td>Once per OBTAIN queue handler command</td>
</tr>
<tr>
<td>DVGCQGSS</td>
<td>Once per GETSSESSPRM queue handler command</td>
</tr>
<tr>
<td>DVGCQCHG</td>
<td>Once per MODIFY queue handler command</td>
</tr>
<tr>
<td>DVGMQCHK</td>
<td>Once per OBTAIN queue handler command</td>
</tr>
</tbody>
</table>
Frequency of Entering and Exiting the SNA Server Modules
The following figures show how often a module is entered and exited during the normal processing of an SNA server for communication between two NetView FTP V2.2.1 MVS systems. The figures show only those server modules that are called frequently and therefore contribute the major part of the amount of tracing output.

Figure 87. Frequency of Entering and Exiting TRQ File Handler (Group 5)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCQTRQ</td>
<td>Once per queue handler command that gains access to the request queue (ADD, DELETE, MODIFY, OBTAIN, NOTIFY) Once for each queried request, on a QUERY ADM queue handler with the file option active The number of records in the request queue when the queue handler is started The number of finished requests for which the delay time has elapsed when the request queue is rebuilt</td>
</tr>
</tbody>
</table>

Figure 88. Frequency of Entering and Exiting Modules and Server Modules Related to Commands and Queue Rebuild (Group 6)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMQSMF</td>
<td>Sum of mask numbers 3 and 4 and once for each operator command and queue rebuild</td>
</tr>
</tbody>
</table>

Figure 89. Frequency of Entering and Exiting User-Written File Handler Modules (Group 1)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCSUR</td>
<td>Once per sent or received record (user file handler)</td>
</tr>
</tbody>
</table>

Figure 90. Frequency of Entering and Exiting File Transfer Manager Modules (Group 2)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCSFP</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGCASTCM</td>
<td>Once per file transfer</td>
</tr>
</tbody>
</table>

Figure 91. Frequency of Entering and Exiting File Handler Modules (Group 3)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMSVE</td>
<td>Once per sent or received record (VSAM ESDS)</td>
</tr>
<tr>
<td>DVGCSVRR</td>
<td>Once per sent or received record (VSAM RRDS)</td>
</tr>
<tr>
<td>DVGCSVK</td>
<td>Once per sent or received record and once per written checkpoint record (VSAM KSDS)</td>
</tr>
<tr>
<td>DVGMSMM</td>
<td>Once per sent or received record (sequential files)</td>
</tr>
<tr>
<td>DVGMSLDS</td>
<td>Once per block of a PDS</td>
</tr>
<tr>
<td>DVGMSPO</td>
<td>Once per CI for a VSAM LDS</td>
</tr>
<tr>
<td>DVGMSBLK</td>
<td>Once per block of a PDS if blocking of sending and receiving PDS differ</td>
</tr>
<tr>
<td>DVGMSSTR</td>
<td>If PDS contains TTR information, once per block of a PDS, plus once per member, plus once per note list</td>
</tr>
<tr>
<td>DVGMSDH</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSBP</td>
<td>Once per block of a PDS</td>
</tr>
</tbody>
</table>

Note: Only one of the file-handling modules DVGMSMM, DVGCSUR, DVGMSVE, DVGCSVRR, DVGMSLDS, DVGMSPO, or DVGCSVK is used during a file transfer. DVGCSVK is also used by the receiving transfer program to write checkpoint records.
### Figure 92. Frequency of Entering and Exiting File Communication Handling Modules (Open/Close) (Group 4)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMSMSD</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSMPF</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSMPS</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGCSCFCN</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSMTI</td>
<td>Once per file transfer</td>
</tr>
</tbody>
</table>

### Figure 93. Frequency of Entering and Exiting File ACF/VTAM Handling Modules (Open/Close) (Group 5)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCSAP</td>
<td>Several times per file transfer</td>
</tr>
</tbody>
</table>

### Figure 94. Frequency of Entering and Exiting Generalized Data Stream Parser Modules (Group 6)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGCSPI</td>
<td>Once per sent or received file</td>
</tr>
<tr>
<td>DVGCSDML</td>
<td>Once per sent or received file</td>
</tr>
<tr>
<td>DVGCSEP9</td>
<td>Once per sent or received file</td>
</tr>
<tr>
<td>DVGCSPD9</td>
<td>Once per sent or received file</td>
</tr>
<tr>
<td>DVGCSPEM</td>
<td>Once per sent or received file</td>
</tr>
<tr>
<td>DVGCSEP8</td>
<td>Once per sent or received file</td>
</tr>
<tr>
<td>DVGCSPD7</td>
<td>Once per sent or received file</td>
</tr>
</tbody>
</table>

### Figure 95. Frequency of Entering and Exiting Server Main Task Modules (Group 7)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMSFT</td>
<td>Once during a server’s run time</td>
</tr>
<tr>
<td>DVGMSNO</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGCSWSM</td>
<td>Once per file transfer</td>
</tr>
</tbody>
</table>

### Figure 96. Frequency of Entering and Exiting User-Exit Routine Interfaces (Group 8)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMSPP</td>
<td>Twice per file transfer</td>
</tr>
<tr>
<td>DVGMSPCX</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSXDM</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSJOB</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSPGM</td>
<td>Once per file transfer</td>
</tr>
</tbody>
</table>

### Figure 97. Frequency of Entering and Exiting File Allocation Modules (Group 9)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMSDN</td>
<td>Two to four times per file transfer</td>
</tr>
<tr>
<td>DVGMSFC</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGCSCFCX</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGCSCDC</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGCSCDNC</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSDNA</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSDND</td>
<td>Once per file transfer</td>
</tr>
<tr>
<td>DVGMSDNT</td>
<td>Once per file transfer</td>
</tr>
</tbody>
</table>
Figure 98. Frequency of Entering andExiting Access Control and Data Security Modules (Group 10)

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Entered and Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGMSAC</td>
<td>Several times per file transfer. Depends on the security parameters specification.</td>
</tr>
<tr>
<td>DVGMSCY</td>
<td>Once per record/block, depending on file type (only if encryption is activated)</td>
</tr>
<tr>
<td>DVGMSCK</td>
<td>Once per receiving, twice per sending server (only if encryption is activated)</td>
</tr>
</tbody>
</table>

OSI Server Trace Output

When NetView FTP V2.2.1 MVS takes a trace, it produces the following output for each trace point in the NetView FTP V2.2.1 MVS OSI modules:

- A header line containing the function name, return code, and name of the storage area dumped
- A storage area dump.

The area that NetView FTP V2.2.1 MVS dumps depends on the module where it takes the trace and the class of the trace.
Chapter 13. Handling NetView FTP V2.2.1 MVS Abnormal Terminations

This chapter contains Diagnosis, Modification, or Tuning Information.

This chapter describes what NetView FTP V2.2.1 MVS does if an abnormal termination occurs in one of its components.

Refer to Chapter 14, “Reporting Program Problems” on page 163 to find out what to do if such an error occurs.

Recovery Routines in NetView FTP V2.2.1 MVS

ESTAE recovery routines are established in the queue handler, the SVC routine, and in the servers. They serve the following purposes:

• Error recording by producing:
  – A SNAP dump of the storage in error.
  – A SNAP dump of the NetView FTP V2.2.1 MVS symptom record. The NetView FTP V2.2.1 MVS symptom record is described in “The Symptom Record” on page 165.
  – A special message to indicate the event of an abnormal termination.

• Error recovery by setting up retry routines that perform the necessary cleanup processing before termination of the task in error.

Notes:

1. Under certain conditions, the recovery operations can fail. For example, if no virtual storage can be obtained from the system, or if the operator stopped a NetView FTP V2.2.1 MVS component by the CANCEL system command.

2. If the recovery operations fail, it is possible that NetView FTP V2.2.1 MVS will not close any data sets. You can close VSAM data sets by using the AMS VERIFY command.

   If you issue the AMS VERIFY command, ignore the VSAM-open warning message IDC3351I, with a return code of 116.

3. If a server ended abnormally, and the data set that the server was to send is a VSAM data set read by an alternate index, you may have to:
   – Verify the alternate index
   – Delete and redefine the path before you can restart the file transfer.

No ESTAE recovery routines are established for the user interface routines. However, because the SVC routine is part of the user interface routines, its ESTAE routine may get control.

Because of the restrictions placed on the SVC, its ESTAE does not attempt to recover the error. It performs cleanup functions and then passes control to the Recovery Termination Manager (RTM).
Error Situations in the Interactive Interface

There are situations, where the interactive interface cannot display a panel or a message. In other situations, an error panel is displayed, but the user must circumvent the error situation. In these cases a return code is displayed on the first line of the panel, if you called the interactive interface with ISPF 7.1 (dialog test).

To call the interactive interface under dialog test, select option 1 (Function) from the Dialog Test Primary Option Menu. Then invoke the Dialog Function/Selection Menu using:

- PGM: DVGIFII
- PARM: x (where x is your National Language Character).

Figure 99 explains the error situations and gives some hints on how to solve the problem.

<table>
<thead>
<tr>
<th>Error Panel</th>
<th>Explanation</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGME9AE</td>
<td>This panel is displayed only for partitioned data set member selection. The error situation can occur, if you send a partitioned data set that does not already exist on another location without specifying one or all members. The interactive interface creates a member list when you select SUBMIT from the Current Task Selection panel (DVGME0Kx) or from the List Saved Requests panel (DVGME0Rx). You select or exclude which members to transfer from this list. Since the data set does not already exist, a member list cannot be created and that error panel is displayed.</td>
<td>1. Check that you selected the correct data set for transfer. 2. If it is correct (for example, an error may have been indicated if the transfer is to start at a later time after you have created the data set), you cannot use the member selection function. You have to transfer a single member or the complete data set. In both cases the member selection list is not displayed. 3. Correct the request and submit it.</td>
</tr>
<tr>
<td>DVGME9CE</td>
<td>This error occurs when a panel cannot be displayed. One of the following may be true: 1. The panel or the message library is not allocated. 2. The specified panel does not exist in the panel library. 3. A message should be displayed with the panel. The message is not contained in the message library. 4. The specified panel contains an error.</td>
<td>1. Check the allocated panel and message libraries. 2. Check if the displayed panel exists in the panel library. 3. Test the displayed error panel using ISPF 7.2 to find any errors in that panel.</td>
</tr>
<tr>
<td>DVGME9DE</td>
<td>ISPF returned with an error code.</td>
<td>1. Panel DVGME9DE displays the information you need to analyze the error: the last ISPF command, the return code, and the short and long error message. 2. Analyze the return code and the ISPF error message by using the ISPF books.</td>
</tr>
<tr>
<td>DVGME9FE</td>
<td>ISPF could not open the saved request table. Possible explanations are: 1. The ISPTABL or ISRTABL is not allocated. 2. The ISPTABL or ISRTABL is in error.</td>
<td>Check your allocation and correct it.</td>
</tr>
<tr>
<td>DVGME9GE</td>
<td>This error occurs for a serious internal error. The error panel shows the previous panel and the next.</td>
<td>Report these panel names to your IBM representative.</td>
</tr>
</tbody>
</table>
Abnormal Termination in the User Interface Routines

The user interface routines normally run for only a very short time. They do not have any recovery routines that intercept an abnormal termination.

In the case of an abnormal termination in the batch job interface routine, the system takes a dump unless you prevent this by JCL statements.

In the case of an abnormal termination in the application program interface routine, the system schedules the recovery routine of your application program, if you have such a routine. Otherwise, the system produces a dump unless you prevent this by JCL statements.

In the case of an abnormal termination in the interactive interface routine, the abend is normally intercepted by ISPF. The following procedure provides additional ABEND information:

1. Press the End PF key, to return to the ISPF/PDF primary option menu (ISR@PRIM).
2. Select option 7.1 and press the Enter key, to go to the Dialog Function/Selection menu (ISRYFP).
3. Invoke one of the following programs, depending on the task you were performing when the ABEND occurred:

<table>
<thead>
<tr>
<th>Program</th>
<th>ABEND occurred in</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVGIFIQA</td>
<td>A query administrator task</td>
</tr>
<tr>
<td>DVGIFIQ</td>
<td>A normal query task</td>
</tr>
<tr>
<td>DVGIFII</td>
<td>Any other task</td>
</tr>
</tbody>
</table>

   You will probably get an ISPF subtask error and go into TSO READY state.
4. Enter TEST and press the ENTER key, to go into TSO TEST mode. The ABEND code is then displayed.
5. Enter W(HERE) and press the ENTER key, to get further details of the location and reason for the ABEND.

Abnormal Termination of the Queue Handler

The processing in case of abnormal termination depends on whether the error occurred during the processing in the queue handler’s main task or in one of its service subtasks.

The queue handler main task controls all processing of the queue handler. It also processes the operator commands.

The queue handler subtasks process the queue handler commands.
The ESTAE Recovery Routine of the Queue Handler Main Task

This recovery routine is implemented in module DVGMQFTP. It performs the following steps:

1. The FSB is modified so that all queue handler commands are prohibited except for:
   - CLOSE
   - NOTIFY
   - STATUS.

2. The NetView FTP V2.2.1 MVS symptom record is set up and issued via a SNAP macroinstruction. For further information on the SNAP macroinstruction, refer to MVS/Extended Architecture Supervisor Services and Macro Instructions.

3. A storage dump is produced via SNAP using the following parameters:

   **PDATA**  All
   **SDATA**  LSQA, SWA, CB, TRT, DM, ERR, IO, Q.

4. If no SNAP dump can be issued, a system dump is requested with the DUMP=YES option of the SETRP macroinstruction. For information on the SETRP macroinstruction, refer to any of the following:
   - MVS/Extended Architecture System Programming Library: System Macros and Facilities, Volume 2
   - MVS/Extended Architecture Supervisor Services and Macro Instructions.

   Otherwise, no system dump is requested.

5. A message indicating the abnormal termination event is issued.

6. If the Recovery Termination Manager (RTM) can provide a System Diagnostic Work Area (SDWA), and the queue handler main line processing has already set up a retry routine address, control is given back to the RTM requesting that the retry routine is to get control.

   The retry routine tries to finish the processing of the queue handler in a controlled manner. This implies the following:
   - All started servers are posted for immediate termination. However, it is possible that the queue handler will not accept any commands from the servers or the user interface routines.
   - All outstanding queue handler commands are terminated.
   - All opened data sets are closed.
   - The space obtained from CSA for the FSB and FSBX is freed.
   - All queue handler subtasks are detached.
   - All held locks are released.

7. If the System Abend Code is hex 122, hex 222, or hex 522, or if the RTM could not provide an SDWA, all outstanding queue handler commands are terminated, the space obtained from CSA for the FSB and FSBX is freed, and the request queue is closed. No retry is requested when returning to the RTM. No SNAP dump is produced if no SDWA is available.
The ESTAE Recovery Routine of a Queue Handler Service Subtask

This recovery routine is implemented in module DVGMQFUN. It performs the following steps:

1. The FSB is flagged so that this specific subtask remains nondispatchable for the rest of the queue handler's processing. Another flag, which indicates that the queue handler has entered the state of immediate termination, is set.

2. The NetView FTP V2.2.1 MVS symptom record is set up and dumped via a SNAP macroinstruction.

3. The following storage areas are dumped via a SNAP macroinstruction:
   - FSB and FSBX
   - Queue handler global control block (QHCB)
   - Request vector table
   - Request queue directory
   - Server table
   - Server group table.

4. A storage dump is produced via a SNAP macroinstruction using the following parameters:
   - **PDATA** All
   - **SDATA** LSQA, SWA, CB, TRT, DM, ERR, IO, Q.

5. A message indicating the abnormal termination event is issued.

6. The address space that waits for the completion of the queue handler command is told that the queue handler command has terminated.

   The return and reason code for the queue handler command reflects the abnormal termination situation.

7. All held locks are released.

8. The buffers for checkpoint data-set processing are released.

9. If no SNAP dump can be issued, a system dump is requested with the **DUMP=YES** option of the SETRP macroinstruction. Otherwise, no system dump is requested.

**Note:** The main task of the queue handler treats a termination of one of its subtasks as an error situation. It processes this situation as if the operator had stopped the queue handler using the operator interface routine.

Abnormal Termination of a Server

The processing in case of abnormal termination of an SNA server depends on whether the error occurred during the processing in the server’s main task or in one of its transferring subtasks. However, because the SVC routine is part of the server, its ESTAE routine may get control.

Because of the restrictions placed on the SVC, its ESTAE does not attempt to recover the error. It performs basic cleanup functions and then passes control to the RTM. The server main task controls all processing of the server. It also processes the scheduling of file-transfer requests. The server subtasks perform the file transfer. What occurs in the case of an abnormal termination of an OSI server is described in “Recovery of an OSI Server” on page 160.
The ESTAE Recovery Routine of the Server Main Task

This recovery routine is implemented in module DVGMSAM. It performs the following steps:

1. All outstanding timers are canceled.
2. All subtasks are informed to terminate immediately. This implies that any current file transfer is terminated after a checkpoint has been taken.
3. The NetView FTP V2.2.1 MVS symptom record is set up and dumped via a SNAP macroinstruction.
4. A storage dump is produced via SNAP using the following parameters:
   
   **PDATA** All
   **SDATA** LSQA, SWA, CB, TRT, DM, ERR, IO, Q.

5. A message indicating the abnormal termination event is issued.
6. If the resource termination manager can provide an SDWA, and the server main line processing has already set up a retry routine address, control is given back to RTM, requesting that the retry routine should get control. The retry routine tries to finish the processing of the server in a controlled manner. This implies the following:
   
   • The cleanup processing for the terminated file transfers is done, including user notification and writing of the NetView FTP log file and the file-transfer report file.
   • All server subtasks are detached after they have finished.
   • The data sets previously opened by the server main task are closed.
   • The server’s ACF/VTAM ACB is closed.
   • The CLOSE queue handler command is issued to disconnect the server from the queue handler.

**Note:** If the operator terminates the server with a CANCEL command, the retry routine cannot be scheduled.

7. If no SNAP dump can be issued, a system dump is requested with the DUMP=YES option of the SETRP macroinstruction. Otherwise, no system dump is requested.
The ESTAE Recovery Routine of the Server Subtasks

This recovery routine is implemented in module DVGMSAX. It performs the following steps:

1. If the abnormal termination results from the following macroinstructions:
   - OPEN
   - PUT (QSAM)
   - CLOSE (QSAM)
   - STOW (QSAM)
   - WRITE (QSAM).

   and an SDWA could be provided, control is given back immediately to the RTM, requesting it to give control to a retry routine. The retry routine issues an error message and then performs all necessary termination processing for the file-transfer request. No dump is taken. If, in this case, no SDWA was obtained by the resource termination manager, no retry is requested by NetView FTP V2.2.1 MVS and the processing in the recovery routine continues.

2. The appropriate return and reason codes for the interrupted transfer are set up and the file-transfer request is updated in the request queue. It is given the status **finished**.

3. The NetView FTP V2.2.1 MVS symptom record is set up and dumped via a SNAP macroinstruction.

4. The following storage areas are dumped via SNAP:
   - Dynamic work areas used by the server subtask that abended
   - Dynamic work areas used by the server main task
   - Area used by module DVGMCQI (queue handler interface module)
   - Area used by the pre-transfer user-exit routine
   - Area used by the post-transfer user-exit routine
   - Area used by the post-conversation user-exit routine
   - Area used by the message definition module for the NetView FTP log file
   - Area used by the message definition module for the file-transfer report file.

5. A storage dump is produced via SNAP, using the following parameters:
   - **PDATA** All
   - **SDATA** LSQA, SWA, CB, TRT, DM, ERR, IO, Q.

6. A message indicating the abnormal termination event is issued.

7. If necessary, the input or output data set is closed.

   **Note:** No checkpoint is taken in case of an abnormal end. If you want to restart the transfer, NetView FTP V2.2.1 MVS uses a checkpoint that was taken earlier, if one exists.

8. If an abnormal termination occurred in the pre-transfer user exit, the server subtask terminates immediately without any recovery action.
9. The post-conversation user exit is called on abnormal termination of the server except when:

   a. The abnormal termination occurred in either the pre-transfer or the post-
      transfer user-exit routine of the server, or within the post-conversation user
      exit itself.

   b. The server has been canceled by the operator.

10. If the abnormal termination did not obviously occur in the pre-transfer and post-
     transfer user-exit routines, the post-transfer user-exit routine is called.

11. If necessary, the input or output data set is deallocated.

12. If necessary, the temporary SAF environment is reset.

13. The buffers needed for checkpoint data-set handling are freed.

14. If necessary, the session with the remote server is terminated.

15. If no SNAP dump can be issued, then either a system dump with the
     DUMP=YES option of the SETRP macroinstruction is requested, or no system
     dump is requested.

The abnormal termination of a subtask in the server leads to a termination of the
whole server if the reason for the abend is one of the following:

- NetView FTP V2.2.1 MVS issued an SVC 13. These user abends are coded to
  indicate program logic error events.

- A program check occurred.

- An SVC was issued by a locked or SRB routine.

Recovery of an OSI Server

In the case of an abnormal termination of an OSI server, any file transfers that
have been passed to OSI/File Services by that server are not affected if the
abnormal termination did not affect OSI/File Services. Only a request that is cur-
rently active in the OSI server is affected by the abnormal termination.

The data structures necessary to complete the transfer request and to inform the
user of the result of the transfer are destroyed. Therefore, the minimum amount of
information to continue processing after abnormal termination of the OSI server (the
contents of the Local Request Table) is written to a recovery file after each signif-
icant change.

If more than one OSI server is used, each server sends its own server name to the
queue handler together with NOTIFY ACTIVE so that each server can be identified.
Every request that has not been given a specific server name by the requester is
given the name of the server that has been selected by the queue handler for the
transfer. This means, each request is associated with one particular server and is
requeued on this server at recovery time. For more information about restarting
OSI file transfer requests, see “Automatic Restart of an OSI File Transfer” on
page 191.
OSI Log File Entries
The log file is used to document the steps performed by the OSI server. This lets
you trace the jobs and the file-transfer requests already performed. The following
information is stored in the log file:

- Starting and stopping a server
- Starting and stopping a file transfer with OSI/File Services
- Changing a server’s session values
- Incorrect or missing server initialization parameters or request parameters
- Error returns from a function
- Entry and exit of major functions if the trace mask parameter is active
- Successful reading of the recovery file.

Errors are reported in the log file together with the appropriate condition codes.

Abnormal Termination of OSI/File Services
In the case of an abnormal termination of OSI/File Services, the OSI server
receives a return code indicating the error situation when it tries to communicate
with OSI/File Services.

There is no mechanism to inform the queue handler that OSI/File Services is no
longer available or to notify the OSI server that OSI/File Services is available again.
Chapter 14. Reporting Program Problems

This chapter explains how to find and report problems within NetView FTP V2.2.1 MVS.

Handling Problems in NetView FTP V2.2.1 MVS

If a problem occurs when you are using NetView FTP V2.2.1 MVS, use the following steps to find out what is causing the problem, whether a correction or a circumvention exists, and how to report a problem to IBM if no correction or circumvention exists.

1. Initial Evaluation
   Determine which system component is causing the problem. If NetView FTP V2.2.1 MVS is causing the problem, identify the NetView FTP V2.2.1 MVS components that are involved.

2. Searching to See If the Problem Is Known
   If the initial evaluation shows that NetView FTP V2.2.1 MVS is causing the problem, develop a set of symptom keywords to describe the problem. Use these keywords to search the software support database for similar problems and their correction or circumvention. If you do not have access to the software support database, proceed with the next step.

3. Reporting the Problem
   If you cannot find a description of the problem, or if no solution is provided with the problem description, report the problem to an IBM Support Center. The Support Center representative helps you search for a problem description. If the results of this search still do not provide a solution, the problem is transferred to an IBM Support Center specialist.

4. Verifying the Problem Description
   The IBM Support Center specialist will contact you. Together, you verify that the symptom keywords used for searching for the problem have been adequate. If the problem is of a high impact or severity, the IBM specialist tries to develop a circumvention to help you continue productive work. If a problem description without a solution has been found, you will be told when a solution from the IBM change team becomes available.

5. Submitting an APAR
   If no description of a similar problem can be found, the IBM specialist prepares an authorized program analysis report (APAR) and sends it to the IBM change team. If necessary, he will ask you to gather additional information and to send it to the IBM change team.

   The IBM specialist processes the APAR through IBM's database of known problems and tells you when a solution has been reported by the IBM change team.
Initial Evaluation

The following describes the error recording facilities in NetView FTP V2.2.1 MVS. These facilities can help you in an initial evaluation of a problem.

Messages

All NetView FTP V2.2.1 MVS components issue messages if they detect an error. The messages are routed to destinations that depend on the NetView FTP V2.2.1 MVS component that has issued them. All NetView FTP V2.2.1 MVS messages are described in *NetView FTP Messages and Codes*.

The messages are generated by:

- The queue handler
- A server
- The batch job interface routine
- The application program interface routine.

The messages have one of the following formats:

- DVGnnnI
- DVGnnnA
- DVGnnnW

where nnn is a unique 3-digit number and the last letter indicates whether the message is informational (I), requires an action (A) from the recipient, or is a warning (W).

The messages that are generated by the interactive interface routine have the format DVGInnn, where nnn is a unique 3-digit number.

In addition to the NetView FTP V2.2.1 MVS messages, the operating system can issue system messages when an error occurs. These system messages are normally contained in the job log of the failed NetView FTP V2.2.1 MVS component.

Codes

For each queue handler command and each file transfer, NetView FTP V2.2.1 MVS issues a return code and a reason code. These codes indicate the result of the command or transfer. They are described in *NetView FTP Messages and Codes*.

Additionally, all NetView FTP V2.2.1 MVS components, except for the interactive interface routine, place a return code in register 15 before they return control to the operating system. These return codes are also described in *NetView FTP Messages and Codes*.

NetView FTP V2.2.1 MVS is able to detect program logic errors. For instance, it can detect if a field in a control block does not have a permissible value. If NetView FTP V2.2.1 MVS encounters a program logic error, it issues an ABEND macroinstruction with a user ABEND code. These codes are explained in *NetView FTP Messages and Codes*. 
Dumps

There are two types of dumps available with NetView FTP V2.2.1 MVS, SNAP dumps and ABEND dumps.

**SNAP Dumps**
When the queue handler or a server terminates abnormally, the ESTAE recovery routines provide SNAP dumps of the storage areas in error. Refer to “Abnormal Termination of the Queue Handler” on page 155, and “Abnormal Termination of a Server” on page 157 for descriptions of which areas are dumped in which abnormal termination situations, and which SNAP dump options are used by NetView FTP V2.2.1 MVS. To enable the queue handler or a server to issue a SNAP dump, you must allocate a data set with the DD name DVGSNAP in the startup job or procedure for these components.

**ABEND Dumps**
When no SNAP dump can be issued by the queue handler or a server, or when an abnormal termination occurs in the batch job interface routine, the interactive interface routine, or the application program interface routine, the operating system produces an ABEND dump. To enable the operating system to issue an ABEND dump, you must allocate a data set with the DD name SYSUDUMP or SYSMDUMP.

**Special Considerations for the Interactive Interface**
Normally, an ABEND in the interactive interface routine does not force an ABEND of the Interactive System Productivity Facility (ISPF); instead, ISPF displays a panel, which includes the ABEND code and which allows you to continue with the main selection panel. In this case no ABEND dump is issued.

If an ABEND occurs while the interactive interface routine is processing, you should enter the dialog test mode of Interactive System Productivity Facility and Program Development Facility (ISPF/PDF), and then reproduce the error. In this case ISPF terminates abnormally and a dump can be taken by pressing the ENTER key after the ABEND message appears, provided that a SYSUDUMP, SYSMDUMP, or SYSABEND data set has been allocated.

For information on using the dialog test option, refer to *ISPF and ISPF/PDF Version 2 for MVS Diagnosis.*

---

**The Symptom Record**

When the NetView FTP V2.2.1 MVS queue handler or an SNA server terminates abnormally, these NetView FTP V2.2.1 MVS components write symptom records as a diagnosis aid. The symptom record is a source of useful information when you are creating a keyword string to identify a problem.

For more information on the symptom record, refer to “The NetView FTP V2.2.1 MVS Symptom Record as Trace Output” on page 145.
The Internal Trace Facility

NetView FTP V2.2.1 MVS provides an internal trace facility, which allows you to prepare error documentation material for the IBM Support Center.

For information on the trace facility, refer to the Chapter 12, “Taking Traces” on page 135.

Searching for Similar Problems

If you are unable to resolve a problem using the information provided in NetView FTP Messages and Codes, report the problem to the IBM Support Center. The following tells you how to specify a set of keywords, called symptom keywords, that describe the problem in a standard way.

Make a copy of the keyword sheet in “Keyword String Development Sheet” on page 175 and insert the keywords as described here. The symptom keywords are used to search for a similar problem. If a matching problem is found, it contains an error description and usually a fix or a circumvention.

A problem that occurs in NetView FTP V2.2.1 MVS can be caused by a NetView FTP V2.2.1 MVS module, or can be the result of errors in components called by NetView FTP V2.2.1 MVS (for example, data management routines) or in the system environment. If the problem has symptoms that are not unique to NetView FTP V2.2.1 MVS and could be caused by a component being used by NetView FTP V2.2.1 MVS, the problem reports of that component should also be searched for a matching problem description.

The symptom keywords can be used more conveniently to search the online Information Systems database (Info-Sys). This database can be installed in MVS systems and also contains descriptions of the known problems in IBM licensed programs.

Report the problem to the IBM Support Center if any of the following are true:

- You have no access to the Information Systems database
- No matching problem is found
- No fix is provided for the problem
- The fix does not resolve the problem.

The IBM Support Center representative will use the symptom keywords to search a more up-to-date database for a matching problem description and notify you of the result.

The Symptom Keywords

The following describes how to fill in and use the keyword sheet in “Keyword String Development Sheet” on page 175. The sheet is divided into the following major parts:

- Component Identification Keywords
- Environment Identification Keywords
- Type-of-Failure Keywords
- Area-of-Failure Keywords
- Transfer-Failure Keywords.
Identifying the Component Identification Keywords

**Operating System**
Identify your operating system. For example, **MVS SP 2.1**, and enter it on the sheet.

**Component Identifier**
The component identifier for NetView FTP V2.2.1 MVS is 568510801. Attach the identifier to the prefix **PIDS/**. This information is in the COMP ID field of the symptom record.

**FMID**
The System Modification Program (SMP/E) identifiers (FMID) for NetView FTP V2.2.1 MVS are as follows:

- **HNTV230** NetView FTP V2.2.1 MVS
- **JNTV231** NetView FTP V2.2.1 MVS Japanese Language feature

Attach the FMID for your environment to the prefix **PIDS/00**. This information is in the FMID field of the symptom record.

**Product Version/Release/Modification Level**
Attach the version, the release, and the modification level of NetView FTP V2.2.1 MVS to the prefix **LVLS/**. This information is in the VERS REL MOD field of the symptom record. For example, NetView FTP MVS Version 2 Release 2 Modification Level 1 is represented as **LVLS/221**.

**Put Level**
In addition to the release level information, it is important to know which program temporary fixes (PTFs) have been applied to NetView FTP V2.2.1 MVS. This information can be retrieved by executing the System Modification Program (SMP/E) PTF list for MVS. Attach the release level information to the prefix **PTFS/**. You can also use this keyword to specify an APAR number or the preventive service level.

Identifying the Environment Identification Keywords

Use an additional keyword to specify the environment in which NetView FTP V2.2.1 MVS is used. Add the related information to the following keywords:

- **PCSS/** Attach the FMID of your version of ACF/VTAM to this keyword. This is useful when a telecommunication error occurs during a transfer.
- **PCSS/** Attach the FMID of your version of RACF to this keyword. This is useful when a problem occurs with the RACF support in NetView FTP V2.2.1 MVS.
- **PCSS/** Attach the FMID of your version of MVS/DFP* to this keyword.
Identifying the Type-of-Failure Keywords

The type of failure can be described by one (or several) of the keywords from the list shown in Figure 100. Select the appropriate keywords and enter the required data on the sheet.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Type of Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB/cxxx</td>
<td>ABEND with ABEND indicator c and code xxx</td>
</tr>
<tr>
<td>MS/msgnum</td>
<td>A message with number msgnum has been issued</td>
</tr>
<tr>
<td>LOOP</td>
<td>The program seems to be looping</td>
</tr>
<tr>
<td>WAIT</td>
<td>The program seems to be waiting for an event without doing anything</td>
</tr>
<tr>
<td>INCORROUT</td>
<td>Incorrect output from NetView FTP V2.2.1 MVS</td>
</tr>
<tr>
<td>PERFM</td>
<td>The performance of the program is degraded</td>
</tr>
<tr>
<td>PUBS/docnum</td>
<td>The book docnum of the NetView FTP V2.2.1 MVS document is in error</td>
</tr>
</tbody>
</table>

**AB/cxxx**

Use this keyword when NetView FTP V2.2.1 MVS terminates abnormally. When this happens a storage dump is produced. For the queue handler and servers, this storage dump contains the symptom record. The ABEND code is in either the SYS ABEND CODE field or the USER ABEND CODE field of the symptom record. The format of the keyword is AB/, followed by a single character, denoting a user ABEND code (U) or a system ABEND code (S), and finally the 3-character ABEND code.

**MS/msgnum**

Use this keyword when an error message is issued. This also applies to messages issued by components other than NetView FTP V2.2.1 MVS, for example, system messages.

The complete message identification is to be attached to the prefix MS/. For example, message DVG071I should be represented as MS/DVG071I.

The message IDs for errors related to the interactive interface are described in *NetView FTP Messages and Codes*.

**LOOP**

Use this keyword when a NetView FTP V2.2.1 MVS component appears to be looping. A NetView FTP V2.2.1 MVS component might be looping if a queue handler command or a file transfer does not finish, but seems to be processing.

The operator should cancel the NetView FTP V2.2.1 MVS component, using the DUMP option.

**WAIT**

Use this keyword if a NetView FTP V2.2.1 MVS component appears to be in a wait state. NetView FTP V2.2.1 MVS is probably in a wait state if a queue handler command or file transfer does not finish and seems not to be processing.

You can see at the operator console whether a NetView FTP V2.2.1 MVS component is active or suspended. If it is active, it is in a LOOP. If it is suspended, it is in a WAIT state. The waiting NetView FTP V2.2.1 MVS component should be canceled by the operator with the DUMP option.
INCORROUT
Use this keyword if the output of a NetView FTP V2.2.1 MVS request is missing or incorrect. Describe the type of output and the deviation.

PERFM
The performance of NetView FTP V2.2.1 MVS is influenced by various factors. Use this keyword if the performance is below explicitly stated expectations.

PUBS/docnum
Use this keyword if a problem is caused by wrong or misleading NetView FTP V2.2.1 MVS documentation. The order number of the book in error should be attached to the prefix PUBS/. For example, this book has the order number SH12-5657. An error in this book would be represented as PUBS/SH125657.
The suffix of the order number can also be added as a 2-digit number. For example, PUBS/SH12565704. The pages and the sections that are in error can be added to the description of the problem to make the search more specific.

Describing the Area-of-Failure Keywords
The NetView FTP V2.2.1 MVS keyword string contains the area-of-failure keywords described in Figure 101. Select the most appropriate keywords and enter the required data. Some area-of-failure keywords are part of the NetView FTP V2.2.1 MVS symptom record.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Area of Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCSS/name</td>
<td>The NetView FTP V2.2.1 MVS function that failed</td>
</tr>
<tr>
<td>RIDS/csect</td>
<td>The failed NetView FTP V2.2.1 MVS CSECT (module)</td>
</tr>
<tr>
<td>RIDS/lmdname</td>
<td>The failed NetView FTP V2.2.1 MVS load module</td>
</tr>
<tr>
<td>PRCS/0000nnnn</td>
<td>The type of the error</td>
</tr>
</tbody>
</table>
The **name** parameter indicates if a queue handler command or a file transfer has failed. It can have the following values:

- **01** OPEN by a server failed
- **02** CLOSE by a server failed
- **05** OBTAIN of a file-transfer request by a server failed
- **06** NOTIFY by a server failed
- **07** DELETE-finished command failed
- **08** MODIFY server session values failed
- **0A** Query-actual record count
- **0F** Call DVGCIIRQX
- **10** ADD command failed
- **20** DELETE-specific-request command failed
- **30** DELETE-all-requests command failed
- **40** MODIFY class, priority, or status of a request failed
- **50** QUERY-specific-request command failed
- **60** QUERY-all-requests command failed
- **61** QUERY ADMINISTRATOR function failed
- **70** GETLUNAME command failed
- **80** GETOPSYS command failed
- **90** STATUS command failed
- **B0** RESTART command failed
- **D0** FORCEDELETE command failed
- **XM** File transfer failed.

**RIDS/csect**

If an ABEND occurs in the queue handler or in a server, NetView FTP V2.2.1 MVS attempts to obtain the name of its failing CSECT (module). Attach the name found in the CSECT NAME field of the symptom record to the prefix **RIDS/**.

If an ABEND occurs in one of the user interface routines, the name of the load module and possibly the name of the CSECT will be in the dump. Attach one of the following to the prefix **RIDS/**:

- The name of the CSECT, if it is in the dump
- The name of the load module, if the name of the CSECT is not in the dump.

**RIDS/lmdname**

If an ABEND occurs in the queue handler or in a server, NetView FTP V2.2.1 MVS attempts to obtain the name of its failed load module. Attach the name found in the LOAD MODULE NAME field of the symptom record to the prefix **RIDS/**.

If an ABEND occurs in one of the user interface routines, the name of the load module will be in the dump. Attach this name to the prefix **RIDS/**.

**PRCS/0000nnnn**

If an ABEND occurs in the queue handler or in a server, NetView FTP V2.2.1 MVS attempts to set up the type of the error that caused the ABEND. Enter the value found in the ERROR TYPE field of the symptom record in place of **nnnn**.
Describing the Transfer-Failure Keywords

The NetView FTP V2.2.1 MVS keyword string contains the transfer-failure keywords as described in Figure 102. These keywords are helpful if you have a problem transferring a file in a specific environment. Because a file transfer always involves two programs, the sending program and the receiving program, there are keywords for the sending node and the receiving node of a file transfer.

The values for the keywords must be retrieved from the file transfer that ended in error. Select the most appropriate keywords and complete them with the required data.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Transfer Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIDS/0Sfid</td>
<td>The FMID of the sending program.</td>
</tr>
<tr>
<td>PIDS/0Rfid</td>
<td>The FMID of the receiving program.</td>
</tr>
<tr>
<td>PCSS/Sc</td>
<td>Data set allocation indicator for the sending program.</td>
</tr>
<tr>
<td></td>
<td>If you specify a value for c, it must be one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>DYNAMIC</strong> If dynamic allocation was performed.</td>
</tr>
<tr>
<td></td>
<td><strong>JOB</strong> If job allocation was performed.</td>
</tr>
<tr>
<td>PCSS/Rc</td>
<td>Data set allocation indicator for the receiving program.</td>
</tr>
<tr>
<td></td>
<td>If you specify a value for c, it must be one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>DYNAMIC</strong> If dynamic allocation was performed.</td>
</tr>
<tr>
<td></td>
<td><strong>JOB</strong> If job allocation was performed.</td>
</tr>
<tr>
<td>PCSS/Sc</td>
<td>Type of the sending data set.</td>
</tr>
<tr>
<td></td>
<td>If you specify a value for c, it must be one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>PS</strong> For a physical sequential data set on DASD (including CMS files).</td>
</tr>
<tr>
<td></td>
<td><strong>PO</strong> For a partitioned data set.</td>
</tr>
<tr>
<td></td>
<td><strong>SL</strong> For a physical sequential data set on labeled tape.</td>
</tr>
<tr>
<td></td>
<td><strong>NL</strong> For a physical sequential data set on unlabeled tape.</td>
</tr>
<tr>
<td></td>
<td><strong>VSAM</strong> For a VSAM data set.</td>
</tr>
<tr>
<td></td>
<td><strong>USER</strong> For a data set handled by a user-written file handler.</td>
</tr>
<tr>
<td>PCSS/Rc</td>
<td>Type of the receiving data set.</td>
</tr>
<tr>
<td></td>
<td>If you specify a value for c, it must be one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>PS</strong> For a physical sequential data set on DASD (including CMS files).</td>
</tr>
<tr>
<td></td>
<td><strong>PO</strong> For a partitioned data set.</td>
</tr>
<tr>
<td></td>
<td><strong>SL</strong> For a physical sequential data set on labeled tape.</td>
</tr>
<tr>
<td></td>
<td><strong>NL</strong> For a physical sequential data set on unlabeled tape.</td>
</tr>
<tr>
<td></td>
<td><strong>VSAM</strong> For a VSAM data set.</td>
</tr>
<tr>
<td></td>
<td><strong>USER</strong> For a data set handled by a user-written file handler.</td>
</tr>
<tr>
<td>PCSS/Sc</td>
<td>Security parameter indicator for the sending node.</td>
</tr>
<tr>
<td></td>
<td>If you specify a value for c, it must be the following:</td>
</tr>
<tr>
<td></td>
<td><strong>SECURITY</strong> Security parameters are specified.</td>
</tr>
<tr>
<td>PCSS/Rc</td>
<td>Security parameter indicator for the receiving node.</td>
</tr>
<tr>
<td></td>
<td>If you specify a value for c, it must be the following:</td>
</tr>
<tr>
<td></td>
<td><strong>SECURITY</strong> Security parameters are specified.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Transfer Failure</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PCSS/Sc</td>
<td>Notification parameter indicator for the sending node. If you specify a value for c, it must be the following:</td>
</tr>
<tr>
<td></td>
<td>NOTIFY A recipient for a file-transfer report is specified.</td>
</tr>
<tr>
<td>PCSS/Rc</td>
<td>Notification parameter indicator for the receiving node. If you specify a value for c, it must be the following:</td>
</tr>
<tr>
<td></td>
<td>NOTIFY A recipient for a file-transfer report is specified.</td>
</tr>
</tbody>
</table>

### Search Strategy

The symptom keywords describe different characteristics of a program failure. The more information provided, the more selective is the search result, resulting in fewer matches. Start the search for the problem description when enough keywords are available to yield a reasonable search result.

1. Inspect each of the retrieved problem descriptions. Eliminate the APAR corrections that are already installed in your system.
2. Compare the error description of the remaining problem records with your problem.
3. If you find a matching problem description, apply the suggested correction.
4. If no solution is provided with the problem description, ask your IBM representative to contact you when a correction is available.

If you cannot find a problem description, extend the search by omitting symptom keywords. The search becomes more general, and the matches found may not apply to your problem. You might start by dropping the environment specification, to include similar failures of NetView FTP V2.2.1 MVS in other environments. If the problem is not environment-specific, the matches might solve your problem. Next, you can omit the maintenance and release-level keywords, to include similar failures for other releases of NetView FTP V2.2.1 MVS.

By dropping other keywords, you retrieve more and more problem descriptions. It will, however, become more and more difficult to decide if the problem described matches your problem.
Submitting an APAR

If you cannot find a solution for the problem in the IBM database of known problems, ask the IBM Support Center to submit an authorized program analysis report (APAR) to IBM. All diagnostic information available should be added to describe the problem as precisely as possible. Here is a list of information that you should prepare:

- The storage dump, if the program terminates abnormally.
- The NetView FTP V2.2.1 MVS log file of the NetView FTP V2.2.1 MVS component which is in error. If possible add the queue handler log file.
- The job log of the NetView FTP V2.2.1 MVS component that is in error.
- A list of the parameters that were specified for a queue handler command, or a file-transfer request.
- If the error is related to the request queue, a printout of the request queue data set with hexadecimal and translated values.
- If the error is related to the checkpoint data set, a printout of the checkpoint data set with hexadecimal and translated values.
- If the error involves a NetView FTP V2.2.1 MVS component that is in a wait state, an address space dump of the waiting component. To create such a dump, use the DUMP system command, which is described in the System Commands book for your operating system.

Additionally, the IBM Support Center may request you to run an internal trace or an ACF/VTAM trace.
Description of the Symptom Record

The queue handler and servers issue the symptom record if either of the following occurs:

- An internal trace is performed
- An abnormal termination occurs and a NetView FTP V2.2.1 MVS recovery routine gets scheduled by the recovery termination manager.

NetView FTP V2.2.1 MVS uses the IDUMP macroinstruction to issue the symptom record.

The startup job or procedure for the queue handler or a server must have a DD statement with the name DVGSNAP.

For more information about specifying the DD statement for the queue handler and a server refer to “DVGSNAP” on page 97.

The symptom record for an internal trace is described in “The NetView FTP V2.2.1 MVS Symptom Record as Trace Output” on page 145.

If there is an abnormal termination, NetView FTP V2.2.1 MVS can produce the symptom record only if the following conditions are met:

- NetView FTP V2.2.1 MVS has reached a point in its processing where it has established its ESTAE recovery routines. This happens at the end of its initialization processing. Therefore, if an abnormal termination should occur during the initialization phase of the queue handler or a server, no symptom record is issued.
- No error occurs that prevents the ESTAE recovery routines from working as designed. The following error situations can prevent a symptom record from being issued:
  - Another abnormal termination occurs during the processing of the ESTAE recovery routines
  - An I/O error on the SNAP dump data set occurs.

Note: Some of the symptom record fields are not filled if NetView FTP V2.2.1 MVS cannot retrieve the related information during its error handling. This is true, for example, if the operating system's recovery termination manager cannot set up a system diagnostic work area (SDWA).

When you read a dump with the symptom record you only have to pay attention to the translated section of the dump. You can normally find this on the right-hand side of the hexadecimal dump. NetView FTP V2.2.1 MVS has already translated all addresses and binary variables, so that you can easily read them in the translated section.

Refer to “The NetView FTP V2.2.1 MVS Symptom Record as Trace Output” on page 145 for an example of a symptom record.
Keyword String Development Sheet

Enter the keywords that apply to your problem.

**Component Identification Keywords:**
- Operating System: __________
- Component Identifier: PIDS/______
- FMID: PIDS/00______
- Product Version/Release/Modification Level: LVLS/______
- PUT Level: PTFS/______

**Environment Identification Keywords:**
- VTAM FMID: PCSS/______
- RACF FMID: PCSS/______
- DFP FMID: PCSS/______

**Type-of-Failure Keywords:**
- ABEND Code: AB/_________
- Message: MS/_________
- Loop condition: LOOP
- Wait condition: WAIT
- Incorrect output: INCORROUT
- Performance: PERFM
- Documentation: PUBS/_________

**Area-of-Failure Keywords:**
- Failing command or function: PCSS/_________
- CSECT (module): RIDS/_________
- Load module: RIDS/_________
- Error type: PRCS/0000_________

**Transfer-Failure Keywords:**
- FMID of sending program: PIDS/0S_________
- FMID of receiving program: PIDS/0R_________
- Allocation type of sending data set: PCSS/S_______
- Allocation type of receiving data set: PCSS/R_______
- Type of sending data set: PCSS/S_______
- Type of receiving data set: PCSS/R_______
- Security parameters for sending node: PCSS/S_______
- Security parameters for receiving node: PCSS/R_______
- Recipient on sending node: PCSS/S_________
- Recipient on receiving node: PCSS/R_________
Appendix A. Data Integrity and Data Security

NetView FTP V2.2.1 MVS allows effective use of the security functions of VSAM and System Authorization Facility (SAF) for MVS-controlled processors. In addition, it provides you with pre-transfer, post-transfer, and post-conversation user exits that permit you to implement an installation-defined security scheme. For further information on the pre-transfer, post-transfer, and post-conversation user exits, refer to the NetView FTP Customization guide.

Note: The pre-transfer, post-transfer, and post-conversation user exits are not available for OSI transfers.

Data Integrity

Between the creation of a file-transfer request and its execution there is a time gap.

As long as the file transfer has not been started, NetView FTP V2.2.1 MVS does not lock the sending or receiving data sets to ensure that the data is not modified. This can leave data exposed, unless you apply precautions against modifying the data sets before the transfer begins.

You can use the not-before and not-after parameters in the file transfer request to mark a time interval within which NetView FTP V2.2.1 MVS can transfer a data set. You must ensure that the data sets are not changed within this interval.

Checking for the Correct Data Set with Job Allocation

If you use job allocation to identify the sending or receiving data set, you must code a DD statement in the startup job or procedure of the server accessing the data set.

To ensure that the correct data set has been found, you can specify the name of the sending or receiving data set in the file-transfer request. NetView FTP V2.2.1 MVS checks the names of the data set in the DD statement against the data-set name in the file-transfer request. If it finds a mismatch, it stops the file transfer.

Checking the Consistency of the Parameters in a Request

Before NetView FTP V2.2.1 MVS transfers any data, it checks that parameter values in the file-transfer request are consistent and valid. Because an inconsistency suggests a possible integrity exposure, NetView FTP V2.2.1 MVS rejects the request before any data is transferred.

Checking the Completeness of a File Transfer

Unless you use a user-written file handler to access data, NetView FTP V2.2.1 MVS maintains a count of sent and received records during a file transfer. At the end of the file transfer the two counts are compared. If a discrepancy is found a message is issued and a return and a reason code is set up. These codes tell the request originator that the file transfer was not successful.

Even if a file transfer fails, the receiving data set exists and jobs that process the received file will process incomplete data.
Data Security

NetView FTP V2.2.1 MVS allows effective use of:

- System Authorization Facility (SAF) products
- VSAM password protection
- Pre-transfer user-exit facility.

These security functions are described in the following.

System Authorization Facility

Note: NetView FTP V2.2.1 MVS does not use the direct interface with SAF products such as RACF, but uses the SAF MVS router interface via the RACROUTE macroinstruction.

An SAF product, such as RACF, helps an MVS installation to control a user’s ability to use resources. It provides users with data set read and write authority on a user basis. The SAF product identifies a user by the USER parameter supplied in a job control JOB statement. The user may also be required to supply a password in the PASSWORD parameter in the JOB statement.

Authorization for a Server: The SAF product regards the startup job for a server as it does for any batch job running under MVS. You can specify a user ID, password, and group ID under which the server accesses the data sets. Similarly, if you use a procedure to start a server, you can define the procedure and its scope of access rights to the SAF product. However, this is not recommended, as it allows any remote user to contact your server and to use the access authority that you give to the server to read and write to your system. As a solution to this problem, NetView FTP V2.2.1 MVS offers dynamic data security.

Dynamic Data Security: Using the values of security parameters specified in the file transfer request, a server that is going to access a data set creates a temporary SAF environment. This SAF environment is established before the data set is allocated and opened. During the existence of the temporary SAF environment, NetView FTP V2.2.1 MVS processes the file under the authorization of the user ID specified in the security parameters of the file-transfer request. The server deletes this environment after it closes and then deallocates the data set. The effect is similar to specifying SAF parameters in a job control statement. However, with NetView FTP V2.2.1 MVS, you can create the data security environment dynamically and request the creation of the security environment from either system of the file transfer.

Because a server can process up to 32 file transfers at the same time, it can also have up to 32 temporary SAF environments at the same time.

If you have security parameters for the server startup job or procedure, the temporary SAF environment supersedes, for as long as it exists, the original environment that was created by the security parameters for the server startup job or procedure. When the temporary SAF environment is deleted, the original SAF environment is reestablished.
This approach of creating a temporary SAF environment has the following advantages:

- You can define new data sets to the SAF product using the authority of the user who created the receiving data set.
- The servers themselves do not have to have access rights for the data sets that are involved in a transfer. This makes NetView FTP V2.2.1 MVS easier to install and protects it from misuse.

NetView FTP V2.2.1 MVS uses the RACROUTE macroinstruction to create a temporary SAF environment. For the APPL parameter the RACROUTE macro supplies the string DVGFTP. This string is made available to installation exit routines. For more information on the RACROUTE macro refer to the following:

- MVS/Extended Architecture System Programming Library: System Macros and Facilities, Volume 2
- MVS/Extended Architecture Supervisor Services and Macro Instructions.

If you use NetView FTP V2.2.1 MVS’s dynamic data security function, you can specify, in the file-transfer request, a user ID, password, and a group ID security parameter for both the sending and the receiving data sets. Under certain circumstances NetView FTP V2.2.1 MVS automatically retrieves the request originator’s password and default connect group from the installed SAF product. NetView FTP V2.2.1 MVS uses either the specified or retrieved security parameter values to gain access to the data sets. The retrieved parameters can be used on the remote system, for instance if the request originator has identical user IDs and passwords on both systems. The NetView FTP Parameter Reference describes how to specify these parameters, and when NetView FTP V2.2.1 MVS automatically retrieves them.

NetView FTP V2.2.1 MVS encrypts the passwords as soon as they are read in. This ensures that the passwords are encrypted before they are transferred across the network and are not visible in a dump.

You can use the SECPAR server initialization parameter to specify whether or not the server automatically rejects all requests for which SAF security parameter values do not exist. Automatic rejection is the default value for SECPAR.

The SECPAR server initialization parameter can be used to prevent requests being processed that do not contain security parameters. Additionally, this parameter controls whether or not the transfer program uses the request originator’s security parameters or the security parameters specified in the request. For further information on the SECPAR parameter, refer to Security Parameters on page 121.

Note: In a request, you can specify xSECURP='*' for the sending and the receiving system. This is useful if both the local and the remote system have the same SAF environment.

The originator of the file-transfer request can give access authorization to the server. In this case, the server runs under the authorization of the user ID specified in the file-transfer request.
Synchronized SAF Environments: If SAF environments at both systems are synchronized and you specify xSECURP='*', NetView FTP V2.2.1 MVS retrieves the request originator's default security parameters. NetView FTP V2.2.1 MVS inserts these parameters into the request and uses them to gain access to the data sets at the sending and receiving system. Passwords are encrypted with the method used by SAF production at the requesting system, and encrypted by the method used at the responding system. The two methods must be the same.

If SAF environments at both systems are not synchronized, the request originator must specify the remote security parameters or the remote server needs to be authorized by the SAF product to access the data set. However, hard-coding passwords and SAF authorization can be avoided by customizing NetView FTP V2.2.1 MVS using the pre-transfer user-exit routine. This user-exit routine is described in the NetView FTP Customization guide. The steps taken by the pre-transfer user-exit routine at the responding system are summarized as follows:

1. The pre-transfer user-exit routine gets the security parameters sent by the requesting system. These values can be specified by the requester or are retrieved by NetView FTP V2.2.1 MVS. The corresponding fields in the APL are APLxACUI, APLxACU2, APLxACPW, APLxACGI, and APLxENCR.

2. Establish the security parameter values according to the rules of the responding system. The user-exit routine can use the parameter values from the previous step, together with its own control tables, and it can extract the parameter values through the MVS SAF router services (RACROUTE REQUEST=EXTRACT).

3. The pre-transfer user-exit routine returns the new security parameter values to NetView FTP V2.2.1 MVS. NetView FTP V2.2.1 MVS then establishes the SAF environment at the responding system.

Notes:

a. NetView FTP V2.2.1 MVS does not retrieve the default security parameters at the responding system using the xSECURP parameter. This is not allowed as it exposes the security mechanism by allowing the user ID for the responding system to be entered at the requesting system that does not represent an authorized user.

b. NetView FTP V2.2.1 MVS does not expose the requesting system's security parameters to the responding system.

Restricting Access to Commands: The standard operator commands START, STOP, and MODIFY are available to the operator and cannot be restricted. The administrator functions can be protected with a master password.

Preventing a Remote System from Initiating a File Transfer: You can use the pre-transfer user-exit routine to prevent a remote system initiating a file-transfer request. You can do this, for example, if you want to transfer files to and from a remote system, but do not want the remote system to start the transfer. To do this, check the APL field APLPSMOD, which can be either L or R for locally and remotely-initiated file transfers, and modify the request to your needs. For more information, see the NetView FTP Customization guide. Another approach is to start servers with the AVLRRH initialization parameter set to 0.
VSAM Password Protection
When you define a VSAM data set, you can specify a password that protects it against any one of the following:

- Altering, updating, or reading
- Updating or reading
- Reading.

If the VSAM data set is VSAM password protected then you can only use the VSAM data set if the correct password is specified when the data set is opened. You specify the password in a file-transfer request.

User-Exit Facilities
The pre-transfer user exit allows an installation, for example, to:

- Check that the user has authority to transfer the specified data set, before the transfer of data starts.
- Indicate that the security environment was verified by the user-exit routine. Then NetView FTP can build the security environment without verifying the password.
- Temporarily or permanently reject a file-transfer request.
- Change or supply parameters in a file-transfer request.
- Insert security parameters that NetView FTP V2.2.1 MVS uses to create its temporary SAF environment. You can use this feature if you do not want to let remote users know the SAF passwords at your installation.

See the NetView FTP Customization guide for more information about the user-exit facilities.

Data Security for OSI File Transfers
For a user of OSI/File Services, who has the capability to access other local filestores, only the filestore passwords control access to files in those filestores. These files might normally be RACF protected. Because OSI/File Services has RACF authorization to access files in its filestores, users who have access to OSI/File Services do not need to provide RACF security information to transfer a file. This means, users who do not have RACF access to a file could still transfer that file using OSI/File Services.

If NetView FTP V2.2.1 MVS is used for OSI file transfers, NetView FTP V2.2.1 MVS:

- Checks that the user ID is registered as a user ID of OSI/File Services.
- Inspects the request using a pre-queuing user exit, and rejects it if it fails some installation-dependent security check.
- Checks, using RACF, that the request originator has access to the data set to be transferred.

Using RACF third-party checking is one of the major advantages of using NetView FTP V2.2.1 MVS in combination with OSI/File Services.
OSI/File Services filestore owners can define the accessibility to their filestores, limiting it:

- Only to themselves, in which case it is a private filestore.
- To other local users, in which case it is a public filestore.
- To users on a remote system, in which case it is an open filestore.

Users can also password-protect their filestores.

For each local OSI/File Services user, the privileged user defines the user capability in the local system, indicating whether users can access:

- Their own filestores
- Other local filestores
- Their own filestores and other local filestores.

The filestore accessibility and user capability is set when a user is defined to OSI/File Services.

The OSI server can only check that OSI/File Services-user ABC has allowed OSI/File Services-user DEF to transfer ABC’s files, if the user ID in OSI/File Services is the same as the TSO user ID. This is only possible for the interactive interface, because for the batch and API interfaces the job name is used instead of the user ID.

For more information about OSI/File Services, refer to the OSI/File Services System/370 User’s Guide.
Appendix B. Transfer Error Recovery

Error Recovery for SNA Transfers

The transfer error recovery in NetView FTP V2.2.1 MVS for SNA servers is based on its checkpoint facility.

Checkpoint Facility

Note: The checkpoint facility is not available for OSI/File Services.

NetView FTP V2.2.1 MVS offers a checkpoint facility to recover from interrupts during a file transfer. You can recover from the following types of errors without transferring the complete data set again:

- Telecommunication-line interruptions
- Abnormal termination of the remote or local system (system malfunction)
- Operator termination of the remote or local FTP.

The checkpoint data set is used to store this recovery information, which is called a checkpoint record. Each checkpoint record is related to one file transfer. The checkpoint record is always written by the receiving server.

Using the recovery information that NetView FTP V2.2.1 MVS has stored when the error occurred, it can resume the interrupted file transfer from the last recorded checkpoint.

Sharing the Checkpoint Data Set

A checkpoint data set is opened and accessed by a server. All servers at one NetView FTP V2.2.1 MVS installation can share the same checkpoint data set. This allows NetView FTP V2.2.1 MVS to restart a file transfer on one server that was interrupted while it was being processed on another server.

You can also define a checkpoint data set for each server in your installation. If you do this, however, an interrupted file transfer can be restarted from checkpoint on the same server only. The request originator of a file-transfer request must ensure this by specifying that the request can be processed only by one specific server.

You must define the checkpoint data set before you can start any server.

The Checkpoint Record

Depending on the NetView FTP level of the requesting transfer program, a checkpoint record consists of the following information:

NetView FTP V2.2 MVS, NetView FTP for Workstations

- Network ID of the requesting NetView FTP
- LU name of the requesting server
- ID of the request queue
- Request number
- Name of the sending data set
- Name of the receiving data set
- Name of the sending transfer program
- Name of the receiving transfer program
- Sending volume serial number (for physical sequential data sets or PDSs)
- Receiving volume serial number (for physical sequential data sets or PDSs)
- Sending PDS member names (for PDSs)
- Receiving PDS member names (for PDSs)
- Sending VSAM catalog name (for VSAM data sets)
- Receiving VSAM catalog name (for VSAM data sets)
- Status information about the sending data set such as record count or byte count
- Status information about the receiving data set such as record count or byte count.

**NetView FTP V2.1 MVS, NetView FTP VM, NetView FTP VSE, NetView FTP/400**

- Name of the sending data set
- Name of the receiving data set
- Name of the sending transfer program
- Name of the receiving transfer program
- Sending volume serial number
- Receiving volume serial number (for physical sequential data sets or PDSs)
- Sending PDS member names (for PDSs)
- Receiving PDS member names (for PDSs)
- Sending VSAM catalog name (for VSAM data sets on VSE or VM systems)
- Receiving VSAM catalog name (for VSAM data sets on VSE or VM systems).

Each checkpoint record has a key that uniquely relates it to the file transfer for which it is created. The key must be unique to avoid the checkpoint record being used to restart a file transfer to which it does not belong.

If the requesting system is NetView FTP V2.2.1 MVS or NetView FTP for Workstations, the requesting server builds a unique key when the file transfer starts. If the requesting system is NetView FTP V2.1 MVS, NetView FTP VM, NetView FTP VSE, or NetView FTP/400, you must ensure that the keys are unique by using unique names for data sets on your system. Also, you should always specify a data-set name, even for a data set on an unlabeled tape.

**When a Server Takes Checkpoints**

During a file transfer, a receiving server writes a checkpoint record into the checkpoint data set. The checkpoint record contains all the information that NetView FTP V2.2.1 MVS needs to restart the corresponding transfer from the most recent checkpoint.

During a file transfer, the receiving server keeps the checkpoint record in its own address space. This in-storage representation of the checkpoint record always reflects the current status of the file transfer. If necessary, the receiving server sends it to the sending server. The receiving server writes the checkpoint record to the checkpoint data set:

- When certain types of errors occur; this is called exceptional checkpointing.
- At specified regular intervals; this is called regular checkpointing.
**Exceptional Checkpointing:** The receiving server takes a checkpoint when any of the following occurs:

- ACF/VTAM signals a RECEIVE error to NetView FTP V2.2.1 MVS.
  This can be caused by any of the following:
  - The session is lost due to a problem in the network.
  - The sending NetView FTP V2.2.1 MVS encounters an error, including abnormal termination.
  - The sending server is stopped.
- The receiving server or its queue handler is stopped by the operator.
- The file-transfer request for which the file transfer is being performed was deleted by a user.
- The queue handler encounters an error and forces all its servers to stop.
- An error occurs in the servers main task that forces all transferring subtasks to terminate the current file transfers.

Recovery information written by exceptional checkpointing provides a record of the amount of data transferred to the receiving system. Only that portion of data that is lost on the line must be retransferred.

**Regular Checkpointing:** The receiving server takes a checkpoint, when one of the following has occurred since the last checkpoint was taken:

- The server has received the number of ACF/VTAM kilobytes that were defined by the BYTECR initialization parameter.
- After each member of a PDS that has been transferred.
- The time specified by the TIMECR initialization parameter has elapsed.

The BYTECR or TIMECR initialization parameter defines the interval between regular checkpointing. This kind of checkpointing is useful to provide recovery information for those types of errors where NetView FTP V2.2.1 MVS does not perform any exceptional checkpointing, for example:

- Abnormal termination of the server during the file transfer
- Abnormal termination of the system (system malfunction)
- I/O error on the receiving data set.

When a file transfer that was interrupted by an error of that type is restarted, only the portion of data that was sent after the most recent checkpoint was taken must be retransferred.

A checkpoint record remains in the checkpoint data set until it is deleted. It is deleted:

- If the associated file transfer has finished successfully
- If the associated file transfer has finished unsuccessfully and the End-of-Processing option for an unsuccessful file transfer was DELETE.

If unsuccessfully finished file-transfer requests are not restarted, the corresponding checkpoint records can fill up the checkpoint data set. You should delete those checkpoint records manually.
Error Recovery for OSI Transfers

OSI Error Handling

If any module of the OSI server returns an error condition to the control module, error processing is invoked.

When an error occurs, the currently active request (that is, the request currently being passed to OSI/File Services) is canceled or the OSI server is stopped, depending on the processing state and the severity of the error. If an error occurs in the final state when the OSI/File Services activity is terminated and can no longer be canceled, a message is sent to the local user or the operator.

Because the OSI server only passes the OSI file-transfer request to OSI/File Services and is not involved in the transfer itself, the OSI server cannot recover a transfer. Recovery mechanism for transfer failures must be performed by OSI/File Services. The OSI/File Services product cannot be configured for a fixed number of retries. The OSI server does not attempt to retry a request once it has been rejected by OSI/File Services.

In case of errors detected in one of the modules, the type and reason for the error is returned to the control module. Appropriate actions are taken and an error message is written to the log file.

There are two ranges of error codes, one for specific server errors and one for errors occurring in OSI/File Services.

The return codes OSI/File Services are modified by adding a fixed value to them to indicate that an error occurred with an OSI request. The condition codes generated by OSI/File Services are passed unchanged as reason codes to the queue handler. In case of an error the return code indicates how the reason code is to be interpreted.

Recovery Mechanisms

The OSI server connects NetView FTP V2.2.1 MVS with OSI/File Services. It does not handle or transfer the MVS file, but passes the information about the transfer request to OSI/File Services. The OSI server must rely on the lower OSI layers for recovery from transmission errors.

Errors concerning the FTAM layer must be handled by OSI/File Services as defined in the FTAM standard. However, Release 1 of OSI/File Services has not implemented recovery mechanisms such as F-RECOVER or F-RESTART. If such an error occurs, the transfer is stopped immediately and the OSI server is informed about the failure. The OSI server has no possibility to recover these errors by itself.

Errors concerning the MVS file system, logical errors, or syntactical errors when using the interface can only be detected but not corrected by the OSI server or OSI/File Services, respectively. Therefore, the OSI server does not attempt to retry an FTAM request once it has been rejected by OSI/File Services until it is re-entered by the local user.
In addition to the errors described here, there may be error situations where one of the system components terminates abnormally, for example the queue handler, the OSI server itself, or OSI/File Services. In these cases some mechanisms to recover information about the request are invoked automatically (see “Automatic Restart of an OSI File Transfer” on page 191).

Automatic Restart of an SNA File Transfer

NetView FTP V2.2.1 MVS can use automatic transfer restart to recover from errors.

How Automatic Restart of a File Transfer Works

The following items show how NetView FTP V2.2.1 MVS’s automatic transfer restart facility works:

- This facility enables NetView FTP V2.2.1 MVS to recover from the following errors:
  - There is a session interruption. This can happen when there is a severe ACF/VTAM problem, or if the secondary system of the file transfer terminates abnormally.
  - The operator terminates one or both of the two servers involved in the transfer.
  - The operator terminates the queue handler and thereby implicitly terminates all servers running on that system.
  - NetView FTP V2.2.1 MVS temporarily cannot allocate the sending or receiving data set. NetView FTP V2.2.1 MVS analyzes the feedback information from its dynamic allocation process. Depending on the feedback information it classifies the error as being permanent or temporary.

If any of the following dsname allocation error codes are returned, the error is classified as being temporary:

- 020C Request for exclusive use of a shared data set cannot be done.
- 0210 Requested data set unavailable.
- 0214 Unit not available.
- 0220 Requested volume not available.
- 0224 Eligible device types do not contain enough units.
- 0228 Specified volume or unit in use by system.
- 0260 Unit does not meet specified status requirements.
- 0264 Invalid request due to current unit status.
- 0268 Tape device is broken.

Otherwise the error is classified as being permanent.

See MVS/XA System Macros and Facilities for a detailed description of the temporary allocation error codes.

- The pre-transfer user-exit routines on the sending or the receiving system have temporarily rejected the file-transfer request.
- The main task of one of the two servers involved in the transfer abends.
The RETRY file-transfer request parameter value determines whether NetView FTP V2.2.1 MVS performs an automatic transfer restart or not. See the NetView FTP Parameter Reference for more information.

If the request specifies automatic transfer restart, the server that obtained the file-transfer request now marks it with the status waiting and writes it back to the request queue. The request number remains the same. Both servers notify the users specified in the file-transfer request about the interruption.

The request remains nondispatchable for the time specified by the RETRYDELAY server initialization parameter. After this time has elapsed it can be scheduled again.

If automatic transfer restart is not requested and one of the preceding situations arises, the server that has obtained the file transfer request marks it with the status finished and writes it back to the request queue. It is not scheduled again until you restart it manually.

Because automatic transfer restart is usually desirable, the default value for the automatic transfer restart parameter is YES.

When the file-transfer request is rescheduled, a checkpoint record from the checkpoint data set is used to restart the file transfer. It is restarted from either the recorded checkpoint or the beginning of the file depending on the value you specified for the Restart Point parameter. The sending and the receiving servers position their data sets to match with the checkpoint and resume the file transfer.

The server issues a special message that indicates if the file transfer is started from the recorded checkpoint or the beginning of the file.

Special Considerations for Using Checkpoints during an Automatic Restart of a File Transfer

If one of the errors occurs for which automatic transfer restart can be done, the receiving server writes its checkpoint record to the checkpoint data set to record the current status of the transfer. This is done as part of NetView FTP V2.2.1 MVS’s exceptional checkpointing and cannot be inhibited.

NetView FTP V2.2.1 MVS stores all the parameters it needs to find the checkpoint record by its key into the file-transfer request, although some of these parameters may not have been specified by the request originator. If, due to an error, NetView FTP V2.2.1 MVS cannot find the checkpoint, it restarts the file transfer from the beginning of the data set.

If you use temporary data sets for a transfer, a restart from checkpoint can fail if the data set no longer exists.

If the receiving data set is a single member of a PDS, the restart from a checkpoint can cause problems because NetView FTP V2.2.1 MVS must allocate a large amount of storage for its restart processing with a PDS member. This is especially true for data sets with undefined records.
Automatic Restart of an OSI File Transfer

All information concerning a request is written to a recovery file. This file contains information about requests that have been passed to OSI/File Services and for which NetView FTP V2.2.1 MVS has not yet been notified as being completed. Each OSI server has its own recovery file.

A recovery file keeps information necessary for the OSI server to continue processing the currently active requests following an abnormal termination. If the OSI server terminates abnormally, the current request is set to waiting. If the queue handler terminates, all outstanding requests are set to waiting. When the OSI server is restarted, it knows from its recovery file which file transfer requests are new and which have already been passed to OSI/File Services.

The OSI server uses the recovery file to rebuild its Local Request Table. This table is updated after each significant change of the server status. When the contents of the recovery file are put into the Local Request Table at initialization time, the recovered entries are marked. It does not matter whether the control structures corresponding to each request are destroyed, because each request must be obtained from the queue handler again before the request can be completed.

If one of the recovered requests was canceled by the operator or the filestore owner, the result is still available for the OSI server as the issuer of the request. This ensures that all recovered requests are finished in the server and removed from the Local Request Table.

Manual Restart of an SNA File Transfer

You can manually restart a transfer, to recover from errors that automatic transfer restart cannot handle, or when automatic transfer restart is not desired.

How the Manual Restart Facility Works

The following items characterize NetView FTP V2.2.1 MVS's manual transfer restart facility:

- If NetView FTP V2.2.1 MVS is not able to restart a transfer automatically, or if you specify that you do not want it to try to automatically restart a transfer, the server that obtains the file-transfer request marks it with the status finished and writes it back to the request queue. Both servers tell the appropriate users about the interruption in the file transfer.

The file-transfer request is not rescheduled until you restart it. You can restart only those file-transfer requests that are marked restartable.

When the request is scheduled, a checkpoint record from the checkpoint data set is used to restart the file transfer. It is restarted from either the recorded checkpoint or the beginning of the file depending on the value you specified for the Restart Point parameter. The sending and the receiving servers position their data sets to match with the checkpoint and the file transfer is resumed.

A special message is issued by the server that indicates if the file transfer is started from the recorded checkpoint or the beginning of the file.
If you request a restart from a checkpoint but the checkpoint record cannot be found, NetView FTP V2.2.1 MVS does the following:

- If the requesting system is NetView FTP V2.2.1 MVS or NetView FTP for Workstations, NetView FTP V2.2.1 MVS ends the file transfer with an error.
- If the requesting system is NetView FTP V2.1 MVS, NetView FTP VM, NetView FTP VSE, or NetView FTP/400, NetView FTP V2.2.1 MVS restarts the file transfer from the beginning of the data set.

**Special Considerations for Using Checkpoints during a Manual Restart of a File Transfer**

You can perform a manual transfer restart from checkpoint for most errors. However, there can be error situations, where NetView FTP V2.2.1 MVS cannot perform exceptional checkpointing, for example, if the server subtask that processes the file transfer terminates abnormally.

In these situations NetView FTP V2.2.1 MVS can only use the regularly written checkpoint record when it restarts a file transfer.

If an error occurs during the transfer of a temporary data set, a restart from checkpoint fails if the data set no longer exists.

In the following situations NetView FTP V2.2.1 MVS may not be able to close a data set:

- The system terminates.
- The system detects an overflow condition for the data set.
- The server subtask that is performing the file transfer terminates abnormally.

Under any of these circumstances, NetView FTP V2.2.1 MVS tries to close the data set, but special error conditions in the system can prevent it from succeeding.

When, due to an interruption, NetView FTP V2.2.1 MVS cannot close a PDS member, the PDS directory is not updated, and so does not reflect the changes that the transfer has effected. It is possible that, although records have been written to a new member of a PDS, the member cannot be found after an interrupt in the transfer. In this case a restart from the recorded checkpoint does not work. If you encounter such a problem, submit the file-transfer request again.

When, due to the interruption, NetView FTP V2.2.1 MVS is not able to close a VSAM data set, you may have to use the Access Method Services (AMS) VERIFY command to close the data set manually before you let NetView FTP V2.2.1 MVS continue the file transfer.
When a File-Transfer Request Remains Active in the Request Queue

If the queue handler encounters an error such as abnormal termination or an I/O error in the request queue, it requests all its servers to terminate immediately. In this case the file transfers that are currently being processed are interrupted and checkpoints are taken. However, the file-transfer requests cannot be updated in the request queue. Their status remains active.

When the queue handler is started again and rebuilds the request queue, it changes the status from active to waiting. It remains nondispatchable for 300 seconds. When this time has elapsed, the file-transfer request can be scheduled again.

Scheduling in the Case of a Restart from Checkpoint (SNA only)

When a data set is being transferred between two NetView FTP systems and the transfer must be restarted, any of the local servers that serve the class of the request can obtain it and continue the transfer. Also, the local server can contact any of the remote servers, as far as this is specified by the file-transfer request.

Automatic Logon Retry (SNA only)

Sometimes a server cannot establish a session 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.

When this happens, 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. Later, NetView FTP V2.2.1 MVS tries again to process the request. It keeps trying until it succeeds in establishing a session. This is called using automatic logon retry.

The automatic logon retry facility is not controlled by a parameter in the file-transfer request—NetView FTP V2.2.1 MVS always performs it. A server that fails to establish a session for one of these reasons does not tell the user who originated the file transfer request.
Appendix C. NetView FTP Connectivity, Coexistence, and Migration

This appendix shows the connectivity between NetView FTP V2.2.1 MVS and the other NetView FTP licensed programs and how to migrate to NetView FTP V2.2.1 MVS from NetView FTP V2.2 MVS, NetView FTP V2.1 MVS, or NetView FTP V1 MVS.

Connectivity and Compatibility

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 V2.1 MVS
- NetView FTP V2.2 MVS
- NetView FTP VM
- NetView FTP VSE
- NetView FTP/400
- NetView FTP for Workstations.

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.

Coexistence

NetView FTP V2.2.1 MVS cannot coexist with another installation with the same level on the same system. NetView FTP V2.2.1 MVS can coexist with installations of a lower level if the installations each have their own environment. That is, each installation must have its own SVC number and must have concatenated the link-list library SDVGLMD2 to the STEPLIB or JOBLIB statement. All installations must share the pre-queuing user-exit routine DVGCIROX.

Migrating NetView FTP V1 MVS to NetView FTP V2.2 MVS

When migrating from NetView FTP V1 MVS (with or without the AFF) to NetView FTP V2.2 MVS, be aware of the following:

- You cannot use the same ACF/VTAM definitions. See Chapter 6, “Setting Up and Maintaining the ACF/VTAM Resources Used by NetView FTP V2.2.1 MVS” on page 79.
- You must define a new checkpoint data set.
- You can use the same NetView FTP MVS batch jobs and application programs. However, these control statements are no longer supported:
  - RCMSFID
  - SCMSFID
  - RCMSUSER
  - SCMSUSER
Furthermore, new file-transfer requests should be set up using the NetView FTP V2.2 MVS control statements. For some control statements the default values have been changed. For more information on the NetView FTP V2.2 MVS control statements, refer to the *NetView FTP Parameter Reference*.

- You can use the same user-written exit routines. However, as the APLs for the two transfer programs are different, you must recompile the exit routines before you can use them with the functions of the new transfer program. The name of the pre-queuing exit has changed from DVGIFRQX to DVGCIRQX.

- You must define a new request queue.

- Requests that were created and saved via the interactive interface of NetView FTP V1 MVS can be migrated via the Convert option in the Main Task Selection of the NetView FTP V2.2 MVS interactive interface.

- Check the following initialization parameters that have changed for NetView FTP V2 MVS:

  **REQDELAY**
  The keyword for specifying the requeue delay time has been changed to **RETRYDELAY**.

  **RUNUMCR**
  Specification of checkpoint intervals with **RUNUMCR** is ignored.

  **RUSIZE**
  Specification of request unit size with **RUSIZE** is ignored.

  **SECPAR**
  The values LOCAL and BOTH are no longer supported for **SECPAR**. You must change your requests to use either the YES or NO options. This change has been implemented to improve data security.

  **TRC**
  The control statement for requesting traces has been changed to **TRACE=value**. For a description of the new values refer to “Activating the Trace Facility” on page 137.

- For some queue handler initialization parameters the default values have been changed.

- The macros DVGAPL, DVGQSR, and DVGQAR have several ready-to-use assembler symbols defined for your convenience. Some of them have been changed from equates (EQU) to define constants (DC). Ensure that they are correctly used in your program or user exits.

- The Server Group Table can still be used. However, the specifications for the FTP level and the server running mode are ignored, as NetView FTP V2.2 MVS obtains this information via the Remote Request Checking function.

- File transfers to locations where NetView FTP V1 MVS or FTP V2 is installed are no longer supported.
Migrating NetView FTP V2.1 MVS to NetView FTP V2.2 MVS

When migrating from NetView FTP V2.1 MVS to NetView FTP V2.2 MVS, be aware of the following:

- You can use the same NetView FTP MVS batch jobs and application programs. However, new file-transfer requests should be set up using the NetView FTP V2.2 MVS control statements. It is recommended that you recompile the application programs with a macro library that contains the APL macro source. For more information on the NetView FTP V2.2 MVS control statements, refer to the NetView FTP Parameter Reference.

- Requests that were created and saved via the interactive interface of NetView FTP V2.1 MVS can be migrated via the Convert option in the Main Task Selection of the NetView FTP V2.2 MVS interactive interface.

- The keyword for specifying the requeue delay time has been changed to RETRYDELAY.

  The keyword REQDELAY is supported for migration purposes only.

- The control statement for requesting traces has been changed to TRACE=value. For a description of the new values refer to “Activating the Trace Facility” on page 137.

  The keyword TRC is supported for migration purposes only.

- You can use the same user-written exit routines. However, you should recompile the exit routines with a macro library that includes the Application Program Parameter List (APL) macro source.

- You can use the same user-written file-handler routines. However, you should recompile the file-handler routines with a macro library that includes the User-Written File-Handler Parameter List (UPL) macro source.

- You must define a new request queue.

- You must define a new checkpoint data set.

- For some SNA server initialization parameters the default values have been changed.

- For some queue handler initialization parameters the default values have been changed.

- The Server Group Table can still be used.

- File transfers to locations where NetView FTP V1 MVS or FTP V2 is installed are no longer supported.

- Additions to the SMF product section or data section are appended to avoid changes of SMF record processing programs.
Migrating NetView FTP V2.2.0 MVS to NetView FTP V2.2.1 MVS

When migrating from NetView FTP V2.2.0 MVS to NetView FTP V2.2.1 MVS, be aware of the following:

- You must define a new file transfer request queue.
- Requests that were created and saved via the interactive interface of NetView FTP V2.2.0 MVS can be migrated via the Convert option in the Main Task Selection of the NetView FTP V2.2.1 MVS interactive interface.
- You can use the same user-written exit routines. However, it is recommended to recompile the exit routines with a macro library that includes the Application Program Parameter List (APL) macro source.
- You can use the same NetView FTP MVS batch jobs and application programs. It is recommended that you recompile the application programs with a macro library that contains the APL macro source.
Appendix D. Storage Requirements

The following list shows the storage requirements for each of the components in NetView FTP V2.2.1 MVS. If not explicitly stated, the storage is allocated below 16MB. Storage areas allocated by system or access method routines to satisfy requests from NetView FTP V2.2.1 MVS are not shown in the tables that follow. The given values are rounded to the next higher number. To find out the storage areas allocated by system or access method routines, run RMF* or an equivalent measurement tool.

Specifying Checkpoint Data-Set Parameters

The following examples describe how to define the checkpoint data set parameters. If you need additional information concerning control interval sizes and buffer space, refer to the VSAM documentation.

Note: The checkpoint facility is not used for OSI transfers.

Number of Records

The amount of records depends on the number of restartable file transfers you expect. However, the minimum is determined by the following formula:

Records = Sum of all running server’s MAXSESS values.

Specifying Request Queue Parameters

The following examples show typical values that can be used in the sample job shown in Figure 41 on page 68. Note that the following are only examples. Your system can require different values.

Record Size

The minimum record size for a request is 1352 bytes. However, it is unlikely that the request queue consists entirely of records of minimum size. It is more likely that the request queue consists of records of varying size, because some parameters require additional space. Use the following list to calculate the additional number of bytes, depending on the type of parameters. If one or more parameter group is specified, the minimum record size increases to 1364 bytes.

<table>
<thead>
<tr>
<th>Parameter Group</th>
<th>Number of Additional Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDS</td>
<td>17 per member specified in a selection list</td>
</tr>
<tr>
<td></td>
<td>8 per member specified in an exclusion list</td>
</tr>
<tr>
<td>VSAM Cluster Definition</td>
<td>164</td>
</tr>
<tr>
<td>User-Written File Handler</td>
<td>660</td>
</tr>
<tr>
<td>SFS</td>
<td>16 + 144 = 160</td>
</tr>
<tr>
<td>OS/400 (receiving from)</td>
<td>16 + 32 = 48</td>
</tr>
<tr>
<td>OS/400 (sending to)</td>
<td>16 + 104 = 120</td>
</tr>
<tr>
<td>SMS</td>
<td>16 + 140 = 156</td>
</tr>
</tbody>
</table>
### Calculating the Average Request Record Size

The following explains how to calculate an average value for the record size:

1. Each request has at least 1352 bytes.

2. As the parameters previously listed can cause a larger request queue when several of the parameters are combined, you must estimate which percentage of all requests will have one of these parameters, and add the percentage x additional length to the minimal record size. An example is shown in Figure 103. All values are in bytes.

#### Figure 103. Example of Calculating the Average Request Size

<table>
<thead>
<tr>
<th>Percentage of All Requests</th>
<th>Description</th>
<th>Additional Length</th>
<th>Total Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Partitioned data set with an average of 50 members selected</td>
<td>850 x 8%</td>
<td>68</td>
</tr>
<tr>
<td>2</td>
<td>User-written file handler</td>
<td>660 x 2%</td>
<td>13.2</td>
</tr>
<tr>
<td>10</td>
<td>VSAM cluster definitions</td>
<td>164 x 10%</td>
<td>16.4</td>
</tr>
<tr>
<td>20</td>
<td>SMS</td>
<td>156 x 20%</td>
<td>31.2</td>
</tr>
<tr>
<td>5</td>
<td>SFS</td>
<td>160 x 5%</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Data encryption</td>
<td>28 x 5%</td>
<td>1.4</td>
</tr>
<tr>
<td>20</td>
<td>APPC</td>
<td>40 x 20%</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>OS/400 (sending to)</td>
<td>120 x 10%</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>Job submission</td>
<td>156 x 10%</td>
<td>15.6</td>
</tr>
<tr>
<td>2</td>
<td>Additional user-exit information (average length, 500 bytes)</td>
<td>520 x 2%</td>
<td>10.4</td>
</tr>
<tr>
<td>5</td>
<td>Program call (average length, 1760 bytes)</td>
<td>1760 x 5%</td>
<td>88</td>
</tr>
<tr>
<td>3</td>
<td>Workstation</td>
<td>323 x 3%</td>
<td>9.7</td>
</tr>
<tr>
<td>Total</td>
<td>Average of specifications</td>
<td></td>
<td>Total 281.9</td>
</tr>
</tbody>
</table>

This results in an additional length of 294 bytes. Therefore, the overall length is $1352 + 12 + 282 = 1646$ bytes.
Calculating the Maximum Request Record Size
Take the following into consideration when you decide the maximum request record size:

- For each of the request records of the types that you use, shown in “Record Size” on page 199, add the number of additional bytes to the minimum record size. Some of the request record types can be used in combination, for example VSAM cluster definition, job submission, and additional user-exit information.

- List all combinations you want to allow and calculate the resulting record size. Then take the largest record as the maximum. Notice that for additional user-exit routine input you can specify up to 1220 bytes, for a post-transfer call up to 4122 bytes. This could mean that parameters from other groups cannot be specified. To avoid this you can limit the length of variable length input records in the user-exits.

If you choose a CI size of, for example, 4096 bytes, ensure that a maximum record size of 4096 - 7 = 4089 is sufficient, otherwise choose a larger CI size and record size.

---

Queue Handler

<table>
<thead>
<tr>
<th>Modules</th>
<th>250KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Areas</td>
<td>200KB x number of FSBSLOTS</td>
</tr>
</tbody>
</table>

Dynamic Areas

Server Group Table
28 bytes x number of records in the server group table

Request Queue Directory
108 bytes x number of records in the request queue.
This area is allocated above 16MB.

Server Table
180 bytes x maximum number of servers

Status Display Area
((MAXSRV x 4 - 1) + 5) x 61

FSBX
MIN (C, L)

where:

- L Size of a request queue record, plus the sizes of the RXB and FSBX headers
- C Value of the CSAELIM parameter.
**Server**

<table>
<thead>
<tr>
<th>Modules</th>
<th>1.1MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Areas</td>
<td>380KB x MAXSESS value</td>
</tr>
<tr>
<td>Dynamic Areas</td>
<td></td>
</tr>
</tbody>
</table>

**I/O Buffers**

For QSAM and BSAM/BPAM processing:
Number of active file transfers x block size x 5.

For VSAM processing:
Number depends on the definition of the accessed cluster.

**Note:** If the server restarts an interrupted file transfer of a single PDS member, additional storage is required. To read the incomplete PDS member temporarily into its own storage, NetView FTP V2.2.1 MVS needs the number of records already transferred in the PDS member x maximum logical record length.

For undefined records, the record length is assumed to be 32,760 bytes.

**Batch Job Interface Routine**

<table>
<thead>
<tr>
<th>Modules</th>
<th>350KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Areas</td>
<td>310KB</td>
</tr>
<tr>
<td>Dynamic Areas</td>
<td>32KB per user + maximum RXB size.</td>
</tr>
</tbody>
</table>

**Application Program Interface Routine**

<table>
<thead>
<tr>
<th>Modules</th>
<th>200KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Areas</td>
<td>50KB</td>
</tr>
<tr>
<td>Dynamic Areas</td>
<td>32KB per user + maximum RXB size.</td>
</tr>
</tbody>
</table>

**Interactive Interface Routine**

<table>
<thead>
<tr>
<th>Modules</th>
<th>500KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Areas</td>
<td>90KB</td>
</tr>
<tr>
<td>Dynamic Areas</td>
<td>32KB per user + maximum RXB size.</td>
</tr>
</tbody>
</table>

**Note:** The table handling of the interactive interface routine requires additional storage, depending on the size of the tables.
Common Storage Requirements

SVC Routine

<table>
<thead>
<tr>
<th>Modules</th>
<th>5800 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The storage for the module resides in the PLPA.</td>
<td></td>
</tr>
<tr>
<td>This routine can run above the 16MB boundary.</td>
<td></td>
</tr>
<tr>
<td>Automatic Areas</td>
<td>250 bytes</td>
</tr>
<tr>
<td>Dynamic Areas</td>
<td>Size of FSBX required</td>
</tr>
</tbody>
</table>

The storage for these areas is allocated in the address space of the caller of the SVC routine.

Static Common Service Area Requirements

Storage for the FSB is allocated by the queue handler in subpool 231. The number of bytes allocated is:

\[(\text{FSBSLOTS value} \times (1124 + 176)) + 40\]

This control block is allocated above the 16MB boundary.

The storage for this control block is freed when the queue handler terminates.

Dynamic Common Service Area Requirements

Storage for the FSBX is allocated when any component requires additional CSA space.

The queue handler allocates its own additional CSA storage. The SVC routine allocates additional CSA storage for the user interface routines and servers.

The storage is freed when it is no longer needed. The components of NetView FTP V2.2.1 MVS must allocate additional CSA storage for an FSBX whenever a queue handler command is to pass data to or from a NetView FTP V2.2.1 MVS component and the RCB is not large enough to contain all the data.
If any of the following apply to your installation, you must calculate and specify the amount of additional CSA storage required. This is important if your system suffers from storage constraints.

- You carry out transfers involving:
  - User-written file handlers parameters
  - Dynamic VSAM cluster definition parameters
  - PDS member selection or exclusion list parameters
  - SFS parameters
  - OS/400 parameters
  - SMS parameters
  - Data encryption
  - APPC security parameters
  - User-exit routine input
  - Job submission parameters
  - Program call parameters
  - Workstation parameters.

- You are not prepared to accept a limited status display.

Calculating the Amount of Additional CSA Storage Required at Your Installation
The following factors determine how much additional CSA storage is required at a particular installation:

- The maximum number of concurrent FSBXs allowed
- The actual size of the FSBX required for each queue handler command.

**Specifying the Maximum Number of Concurrent FSBXs Allowed:** The maximum number of FSBXs that can exist concurrently is determined by and is equal to the number of commands that the queue handler can process concurrently. This number is installation-dependent and is set via the FSBSLOTS queue handler initialization parameter.

As the maximum FSBSLOTS parameter value is 32, the maximum number of FSBXs that can exist at any one time is also 32.

**Determining the Size of an FSBX:** The following factors determine the actual size of an FSBX:

- If the FSBX is allocated for use with a user query command, its size depends on the number of user query records to be passed to the user.

- If the FSBX is allocated for use with an administrator query command, its size depends on the number of administrator query records to be passed to the administrator.

- If the FSBX is allocated for use with an operator DISPLAY command, its size depends on the amount of data returned from the component or components specified in the command. An FSBX of 2042 bytes is needed to hold all possible data.
If the FSBX, containing an RXB, is allocated for use with an ADD, OBTAIN, or NOTIFY command, its data area size depends on the parameter or parameters in the request that need additional space. The number of additional bytes shown in “Record Size” on page 199, are also the number of bytes necessary for the corresponding FSBX. The maximum request queue record size is set during the installation of NetView FTP V2.2.1 MVS.

You can specify either of the following types of record length:

Fixed  The minimum size that can be set is 1124 bytes; this allows only requests that do not need an RXB to be processed.

Variable  If you specify variable request queue record size, you must specify an average and a maximum size.

The following examples show the type of parameters that can be processed with various maximum request queue record sizes; use them as a guide to select the optimum value for your installation.

**Example 1:** Maximum record size set to 1845 bytes.

This allows one of the following to be processed in one file-transfer request:

- User-written file handler parameters
- VSAM cluster definition parameters
- Up to 39 selected PDS members
- Up to 84 excluded PDS members
- SMS parameters
- Data encryption parameters
- APPC security parameters.

**Example 2:** Maximum record size set to 1653 bytes.

This allows one of the following to be processed in one file-transfer request:

- VSAM cluster definition parameters
- Up to 25 selected PDS members
- Up to 52 excluded PDS members
- SFS parameters
- Data encryption parameters
- OS/400 parameters.

The value specified in the CSAELIM queue handler initialization parameter limits the maximum total amount of additional CSA storage that can be allocated for FSBXs at any one time.

Each RXB is embedded in the FSBX and passed to or from the queue handler via the CSA. NetView FTP V2.2.1 MVS tracks the total amount of additional CSA storage allocated for FSBXs and rejects any request for further storage when the amount of available storage is less than the difference between the request queue record size (RECSIZE) and the sum of the RCB and RXB header sizes (1148 bytes).
Therefore, the maximum RXB size is:

\[ RXBMAX = \text{MIN} (A, B) \]
\[ A = \text{RECSIZE} - (\text{RCBSIZE} + \text{TRQHEAD}) \]
\[ B = \text{CSAESIZE} - (\text{FSBXHEAD} + \text{RXBHEAD}) \]

where:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXBMAX</td>
<td>The maximum amount of CSA storage that can be allocated for the current request</td>
</tr>
<tr>
<td>RECSIZE</td>
<td>The maximum request queue record size</td>
</tr>
<tr>
<td>RCBSIZE</td>
<td>The size of the RCB (1124 bytes)</td>
</tr>
<tr>
<td>TRQHEAD</td>
<td>The size of the RXB header on the transfer request queue (12 bytes)</td>
</tr>
<tr>
<td>RXBHEAD</td>
<td>The size of the RXB header (24 bytes)</td>
</tr>
<tr>
<td>CSAESIZE</td>
<td>(Value of CSAELIM parameter - CSA storage already allocated for FSBXs)</td>
</tr>
<tr>
<td>FSBXHEAD</td>
<td>The size of the FSBX header (12 bytes).</td>
</tr>
</tbody>
</table>

The queue handler passes the value of the maximum FSBX size that can be allocated to the other NetView FTP V2.2.1 MVS components as follows:

User Interfaces Via the GETMXLEN queue handler command
Servers Via the OPEN queue handler command.

**Selecting the Optimum CSAELIM Value**

It is the NetView FTP V2.2.1 MVS system administrator responsibility to select and specify the CSAELIM value best suited to a particular installation.

The following examples show a range of possible CSAELIM values and discuss the advantages and disadvantages of each.

**Example 1:** CSAELIM value = number of FSBSLOTS \times (RECSIZE - 1148)

This is the largest CSAELIM value that ever needs to be specified. No queuing bottlenecks can occur, as the queue handler can allocate the maximum additional CSA storage for every request.

If the CSA storage value is large, it reduces the storage that would otherwise be available for other tasks.

**Example 2:** CSAELIM value > (RECSIZE - 1148) < number of FSBSLOTS \times (RECSIZE - 1148)

CSAELIM values in this range let the administrator consider the advantages of saving CSA storage against the risks of possible queuing bottlenecks.
**Example 3:** CSAELIM value = (RECSIZE - 1148 bytes)

This CSAELIM value is acceptable for installations where requests that need an RXB occur rarely, and the saving in CSA storage is worth the risk of an occasional queuing bottleneck.

**Example 4:** CSAELIM value < (RECSIZE - 1148 bytes)

CSAELIM values in this range are not recommended, for the following reasons:

- DASD space is wasted.
- Frequent queuing bottlenecks can occur.
- Requests whose required RXB is greater than the CSAELIM value might never be processed.

It is the system programmer's responsibility to ensure that the amount of CSA storage specified in the CSAELIM parameter value is available. The amount of storage required for each FSBX must be contiguous, but FSBXs need not be contiguous to each other.

If sufficient CSA storage cannot be allocated when the queue handler attempts to carry out a command, the following occurs:

- **ADD** The request is rejected.
- **QUERY** The request is processed without additional CSA storage.
- **OBTAIN** The request is deferred.
- **NOTIFY** The RXB on the request queue is not updated.
- **STATUS** The display is truncated.

When the queue handler is started or restarted, and the request queue directory is rebuilt, all requests that need an RXB are checked to see if they are dispatchable. If the CSAELIM value is less than the RXB size for a request, then the CSAELIM value must have been decreased since the request was added to the queue; in this case the request is marked as blocked and does not become dispatchable until the CSAELIM value is increased to a value greater than the RXB size.
Appendix E. Factors That Affect the Performance of NetView FTP V2.2.1 MVS

This appendix describes the factors and the parameters whose values affect the performance of NetView FTP V2.2.1 MVS.

FSBSLOTS Queue Handler Initialization Parameter Value

The values specified for this parameter determine the following:

- The total number of subtasks that the queue handler can process concurrently.
- The number of subtasks that are reserved for TSO users.

The following factors must be taken into consideration when selecting these values:

- The larger the value, the less the wait time for a free FSB slot.
- The larger the value, the more wasted CSA storage during periods of low queue handler activity.
- The total number of FSBSLOTS determines the maximum allowed number of FSBXs. When selecting these values, the system administrator should consider the following factors against each other:
  - The probability of CSA shortages, due to a requirement for relatively large numbers of concurrent FSBXs.
  - The probability of queuing bottlenecks, due to relatively large numbers of deferred requests.
- The ratio of slots reserved for TSO users determines the relative response times for TSO users and for other subtasks.

CSAELIM Queue Handler Initialization Parameter Value

The value specified for this parameter determines the total amount of additional CSA space that can be allocated for FSBXs at any one time. When selecting this value the system administrator must consider the following factors against each other:

- The amount of CSA space that can be allocated without affecting the general system performance.
- The probability of queuing bottlenecks, due to a relatively large additional CSA space requirement.

Unless you have common service area constraints, you can use the default (NOLIM). However, take into consideration the maximum value at your installation.
TIMECR and BYTECR Server Initialization Parameter Values

The value specified for TIMECR or BYTECR determines how often NetView FTP V2.2.1 MVS takes a checkpoint during a file transfer. These parameters do not apply for OSI transfers.

The receiving server cannot receive any data while it is writing data to the checkpoint data set. When selecting a value for this parameter the system administrator must consider the following factors:

- The need for frequent checkpoints, to reduce the amount of data that must be retransferred when an error occurs.
- The amount of I/O time used in writing checkpoints frequently.

The faster the speed of the line the less important is the need for regular checkpointing. You can specify BYTECR=0 to stop regular checkpointing for smaller data sets. However, if regular checkpointing is used and PDS members are transferred, a checkpoint is written after each member. This may cause severe performance degradation if the PDS members are small.

The Maximum Number of File Transfers That NetView FTP V2.2.1 MVS Can Process Concurrently

This number is determined by a combination of the following:

- The MAXSRV queue handler initialization parameter value. The value specified for this parameter determines the maximum number of servers that can be started at any one time.
- The number of servers that are started. The operator can change this number.
- The MAXSESS server initialization parameter value. The value specified for this parameter determines the maximum number of file transfers that any server can process simultaneously. For an SNA server the value limits the maximum number of subtasks for that server. You cannot change the value during processing.

An OSI server works asynchronously and does no subtasking. MAXSESS specifies the maximum number of requests that the OSI server can handle at the same time. It is used to limit the storage used to keep information about requests until they are completed.

- The AVLSESS, AVLLRH, and AVLRRH server initialization parameter values. The values specified for these parameters determine the number of file transfers that the relevant server can process concurrently, as follows:
  
  **AVLSESS**  Total number of file transfers the relevant server can process concurrently
  
  **AVLLRH**  Number of local file transfers

  For OSI servers this parameter value must be the same as the AVLSESS parameters as OSI servers can only handle locally initiated sessions.
AVLRRH  Number of remote file transfers

For an OSI server this parameter must be set to zero because an OSI server is only used for locally initiated file transfers.

When selecting values for the preceding parameters, the system administrator must consider the following factors:

- The total of the amount of automatic and dynamic storage per server that must be allocated, as follows:
  - Automatic space for each possible concurrent file transfer; determined by the MAXSESS values for the server.
  - Dynamic space for each active file transfer.
- The probability of queuing bottlenecks, due to the server or servers for a particular request class not having enough sessions available to process the requests in that class.

The Compression Method Used

The originator of a file-transfer request can specify any of the following types of compression:

- **NONE**  No compression is used
- **ADAPT**  Adaptive compression is used
- **SNA**  SNA-type compression is used.

The type of data to be transferred determines which of these methods is the most effective (see the NetView FTP Parameter Reference for more information about compression methods). The system administrator must take the following into consideration when determining which type of compression should be used on the system:

- No compression uses the least CPU time, but usually involves the transfer of the greatest amount of data.
- Adaptive compression uses the most CPU time, but usually involves the transfer of the least amount of data. This is the most effective method for most types of data.
- SNA compression involves less CPU time than adaptive compression, but usually involves the transfer of more data.

The system administrator can then advise the users on the type of compression they should use.

Server Classes

When setting up the classes of request that the servers on a system can process, the system administrator must take the following into consideration:

- The general requirements of the installation.
- The speed with which requests of a particular class are processed depends on the total number of local request handlers that are available to process requests of that class. This is determined by the sum of the AVLLRHH initialization parameter values of all started servers that can process requests of that class.
The OBTINTVL Server Initialization Parameter Value

The value specified for this parameter determines the interval after a server unsuccessfully tries to obtain a request from the queue handler, or an OSI server unsuccessfully tries to get a result from OSI/File Services, before the server tries to get a new file transfer request from the queue handler or a result from OSI/File Services. When selecting a value for this parameter, the system administrator must take the following into consideration:

- The lower the OBTINTVL value, the less the wait time between trying to obtain a request from the queue handler.
- Every time a server tries to obtain a request from the queue handler an FSB slot is used; a low OBTINTVL value can cause bottlenecks, if the FSBSLOTS value is low and other NetView FTP V2.2.1 MVS components are waiting for FSB slots.

Request Unit

Request-Unit Size

The value for the RU size is taken from the VTAM logon mode table specified in server initialization parameters. The value specified for this parameter determines the size of the RUs transferred. NetView FTP V2.2.1 MVS supports any RU size that is supported by the installed version of VTAM; this has the following effects on performance:

- Controls the number of ACF/VTAM cycles when sending and receiving the RUs.
- Gives a better network tuning capability for the user. You can define the best RU size for the network, depending on the applications, control units, and other resources.

For customization and tuning an RU size of 2KB is recommended.

Negative impacts of large RU sizes are:

- More system storage for each session is needed.
- They could decrease the overall performance of the system.
- Severe ACF/VTAM buffer problems could occur.
Pacing

How Pacing Affects Throughput

Pacing, as specified for the VPACING operand of the ACF/VTAM APPL definition statement, can have a considerable effect on throughput. However, the best value for pacing is difficult to calculate with precision, because you can alter both RU size and pacing to affect the throughput.

A small RU size and a pacing value of one causes the least effect on other components of the network, but gives the slowest throughput. Conversely, a large RU size and a large pacing value have a greater effect, but give the best performance (assuming the ACF/NCP/VS or ACF/VTAM components can handle the volume of data). Often the best values are probably intermediate values, unless the installation has special requirements.

For customization and tuning, start with a VPACING value of 4.

Negative impacts of no pacing or a large pacing window are:

- Severe ACF/VTAM buffer problems could occur.
- The buffers of the LU and the intermediate routing nodes could overflow.
- More system storage must be reserved for each session.

On links where SNA class-of-service is used, you can adjust this parameter. If pacing is not used, severe ACF/VTAM buffer problems might occur.

Pacing Specifications

Pacing specifications can be particularly important when controlling the flow of data in a network, especially to avoid a Network Control Program (ACF/NCP/VS) “slow down.” The following explanations are useful when defining pacing values for the NetView FTP V2.2.1 MVS BIND image in the logon mode table entry:

**PSNDPAC and SRCVPAC are Zero**

The value of VPACING from the APPL is used. If the NetView FTP V2.2.1 MVS sending server is the primary LU, VPACING controls pacing.

**PSNDPAC and SRCVPAC are Nonzero**

The values are unchanged, that is, VPACING is not used. If the NetView FTP V2.2.1 MVS sending server is the primary LU, the value in the BIND, not VPACING, controls pacing.

**SSNDPAC is Zero**

The values are unchanged, that is, VPACING is not used. If the NetView FTP V2.2.1 MVS sending server is the secondary LU, the value in the BIND image, not VPACING, controls pacing. In other words, there is no pacing. This might cause severe buffer problems in ACF/VTAM.

**SSNDPAC is Nonzero**

The value of VPACING from the APPL is used. If the NetView FTP V2.2.1 MVS sending server is the secondary LU, VPACING controls pacing.
Rules for Defining Pacing Values in a Logon Mode Table

When defining your pacing values, you should follow these general rules:

- To permit pacing control, always define a logon mode table entry for all servers at all locations where NetView FTP is installed.
- Always set SSNDPAC to a nonzero value.
- Set PSNDPAC and SRCVPAC to zero.
- Always define VPACING in your APPL definition statements.

By using these rules, you ensure that a server paces all data transfers as specified in the VPACING operand in the APPL statement. This permits you to make simple changes to pacing values without having to assemble and link-edit the logon mode table again.

Note: If no session parameters are found via the default logon mode table entry, NetView FTP V2.2.1 MVS sets all pacing fields to zero in the BIND image that it creates.

Machine Cycles versus Transfer Time

Normally, the limiting factor in performance is the transfer time on the line rather than the number of cycles used by the processing unit to process the data.

For example, assume that NetView FTP V2.2.1 MVS is transferring 120-character records and that it is compressing each one to about 100 characters. Assuming an RU size of 1KB and a telecommunication link that operates at 9600 bits per second (about 1200 bytes per second), it takes slightly less than one second to transfer the block of data.

In fast processing units, the data could become available much faster than the line can handle it, unless the line is extremely fast. Normally, the way to improve performance when using NetView FTP V2.2.1 MVS would be to examine those aspects that could improve the transfer time, such as the introduction of faster lines, more lines (that is, use of transfer groups), increase of pacing value, reduction of contention on lines, and increase of RU size.

Simultaneous File Transfers per Server versus Several Servers

The system administrator must carefully consider the file transfer workload:

- When a particular file is to be transferred.
- The size of the files to be transferred.
- Subsequent job dependencies.

The file transfer should be classified, that is, a server class is to be given for each file transfer, for example:

Class O Default class
Class T Tape mount required
Class L Long runtime, large datasets
Class S Short runtime, small datasets
Class N  Night shift
Class C  Low checkpointing frequency
Class P  High checkpointing frequency
Class R  Server reserved to serve remote requests only
Class Z  Maintenance, system support.

Unless otherwise specified, all file transfers run with the same dispatching priority. However, you can specify a different service profile for each server by defining them in the IEAIPSxx and IEAICSxx members of SYS1.PARMLIB.

You can use as many simultaneous file transfers per server until each server is 100% busy.

Generally, if small data sets with small block sizes are transferred, and if the RU size being used is small, a server can run more file transfers simultaneously. However, if large data sets with large block sizes are transferred, and the RU size is large, the server is kept busy and the number of simultaneous file transfers is reduced. Additionally, if adaptive compression is used, the consumed CPU time per transferred record can increase. Therefore, for some servers, the number of available simultaneous file transfers (AVLSESS) might be set to a low value.

The more NetView FTP is accessed and the more resources are allocated to NetView FTP, the more attention must be given to performance. NetView FTP is a subsystem and must be treated like other subsystems, and defined parameters may need modifying occasionally to optimize performance. To help make the correct decision about which parameters to modify, specify the writing of SMF records and analyze them using data reduction programs, for example SLR.

Additional Considerations

The more NetView FTP is accessed and the more resources are allocated to NetView FTP, the more attention must be given to performance. NetView FTP is a subsystem and must be treated like other subsystems, and defined parameters may need modifying occasionally to optimize performance. To help make the correct decision about how many servers to be started, and which parameters to modify to attain a balanced workload, use any system diagnostic tool to analyze them.
Appendix F. Fixed Parameters and Constants Used in NetView FTP V2.2.1 MVS

This appendix lists the processing constants used in the queue handler. These constants cannot be specified by the NetView FTP V2.2.1 MVS installation.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time interval waited by the queue handler for all servers to deliver their current status information in case of a DISPLAY command.</td>
<td>30 seconds</td>
</tr>
<tr>
<td>Time interval waited by the queue handler main task for all sub-tasks to finish their processing and for all started servers to stop in case of an immediate termination situation.</td>
<td>180 seconds</td>
</tr>
<tr>
<td>Amount of time that a file-transfer request that had the status ACTIVE when the queue handler rebuilt the request queue at startup time has to wait before it can be scheduled again.</td>
<td>300 seconds</td>
</tr>
</tbody>
</table>
Appendix G. NetView FTP V2.2.1 MVS Log Files

NetView FTP V2 MVS maintains three log files:
- Batch Job Interface Log File
- Queue Handler Log File
- Server Log File.

Each file consists of a chronological series of messages that document NetView FTP processing. The messages are described in *NetView FTP V2 MVS Messages and Codes*.

This appendix includes a sample of each of the three log files.

### Batch Job Interface Log File

The following is a sample of a Batch Job Interface Log File.

```
DVG550I NETVIEW FTP V2R2M1; BATCH JOB INTERFACE STARTED ON 94/10/17 AT 17:18:31; MAINTENANCE LEVEL =
DVG550I FTP V2 R2 M1 - 94/10/04 09:00 FIT
DVG02I * Parameters to add a file transfer request
DVG02I FUNCTION=ADD
DVG02I XMODE=TO
DVG02I CLASS=1
DVG02I PRIORITY=1
DVG02I RMTNODE=MVS
DVG02I SSECURP='*'
DVG02I SFILEID='DVG.NFTP23/.*.SDVGSAM0'
DVG02I MEMBER=(R,DVGMXSO5)
DVG02I SFILEORG=PO
DVG02I RNOTIFY=UID
DVG02I RSECURP='*'
DVG02I RFILEID='DVG.NFTP230..Receive.Data.Set1'
DVG02I RSTATOPT=MNE
DVG02I REOPOK=CATLG
DVG02I RFILEORG=PO
DVG02I ++++++++++++++++ Question: Batch Job Interface Log File
```

---

Figure 105. Batch Job Interface Log File
Queue Handler Log File

The following is a sample of a Queue Handler Log File.

```
DVG4101 NETVIEW FTP V2R2M1; QUEUE HANDLER STARTED ON 94/10/17 AT 13:14:55; MAINTENANCE LEVEL = NFTP
DVG4101 MVS INITIAL VERSION 94/10/04 09:00 FIT
DVG4101 **********************************************************
DVG4101 -- NETVIEW FILE TRANSFER PROGRAM VERSION 2.2.1 --
DVG4101 -- Generated by macro DVGMXJG2 --
DVG4101 -- Member: DVQQHP --
DVG4101 -- Parameter: --
DVG4101 -- ADMPW SRTF5 --
DVG4101 -- SVNPRG DVG230 --
DVG4101 -- Function: --
DVG4101 -- INITIALIZATION PARAMETERS FOR THE NETVIEW --
DVG4101 -- QUEUE HANDLER DVG230QH --
DVG4101 **********************************************************
DVG4101 ADMIN=SRTF5
DVG4101 NFTPNAME=TESTSYS
DVG4101 MAXSRV=04
DVG4101 SVRPREF=DVG230
DVG4101 STARTSV1=(01,02)
DVG4101 NVEVENT=ALL
DVG4101 REQUEST QUEUE INITIALIZED WITH A MAXIMUM NUMBER OF 99 REQUESTS
DVG4101 QUEUE HANDLER READY
DVG4241 NETVIEW FTP ACCEPTED COMMAND TO START SERVER DVG23001
DVG4241 NETVIEW FTP ACCEPTED COMMAND TO START SERVER DVG23002
DVG4501 DVG23001 ISSUED QUEUE HANDLER COMMAND 01 FOR REQUEST NUMBER 0 AT 94/10/17 ON 13:15:14; RC = 0 ; RSN = 0
DVG4501 DVG23002 ISSUED QUEUE HANDLER COMMAND 01 FOR REQUEST NUMBER 0 AT 94/10/17 ON 13:15:14; RC = 0 ; RSN = 0
DVG4501 UID ISSUED QUEUE HANDLER COMMAND 00 FOR REQUEST NUMBER 0 AT 94/10/17 ON 13:15:43; RC = 0 ; RSN = 0
DVG4501 UID ISSUED QUEUE HANDLER COMMAND 10 FOR REQUEST NUMBER 1 AT 94/10/17 ON 13:15:44; RC = 0 ; RSN = 0
DVG4321 NETVIEW FTP ACCEPTED OPERATOR SUBCOMMAND D 01
DVG4501 DVG23001 ISSUED QUEUE HANDLER COMMAND 90 FOR REQUEST NUMBER 0 AT 94/10/17 ON 13:16:28; RC = 0 ; RSN = 0
DVG4591 DISPLAY OF SERVER DVG23001
DVG4491 0 OF 0 SESSIONS DISPLAYED
DVG4501 DVG23001 ISSUED QUEUE HANDLER COMMAND 05 FOR REQUEST NUMBER 1 AT 94/10/17 ON 13:16:29; RC = 0 ; RSN = 0
DVG4501 DVG23001 ISSUED QUEUE HANDLER COMMAND 06 FOR REQUEST NUMBER 1 AT 94/10/17 ON 13:16:38; RC = 0 ; RSN = 0
DVG4501 DVG23001 ISSUED QUEUE HANDLER COMMAND 06 FOR REQUEST NUMBER 1 AT 94/10/17 ON 13:16:50; RC = 0 ; RSN = 0
DVG4331 NETVIEW FTP ACCEPTED OPERATOR SUBCOMMAND TO STOP QUEUE HANDLER
DVG4071 QUEUE HANDLER TERMINATION IN PROGRESS
DVG4501 DVG23002 ISSUED QUEUE HANDLER COMMAND 02 FOR REQUEST NUMBER 0 AT 94/10/17 ON 13:17:24; RC = 0 ; RSN = 0
DVG4501 DVG23001 ISSUED QUEUE HANDLER COMMAND 02 FOR REQUEST NUMBER 0 AT 94/10/17 ON 13:17:24; RC = 0 ; RSN = 0
DVG4031 2 OF 2 SERVERS HAVE TERMINATED
DVG4041 NETVIEW FTP V2R2M1; QUEUE HANDLER ENDED ON 94/10/17 AT 13:17:31; RETURN CODE = 0
```

Figure 106. Queue Handler Log File
The following is a sample of a Server Log File.

```plaintext
DVG0011 NETVIEW FTP V2R2M1; SERVER DVG23001 STARTED ON 94/10/17 AT 13:15:12; MAINTENANCE LEVEL = NFTP
DVG0011 MVS INITIAL VERSION 94/10/04 09:00 FIT
DVG0011
DVG0011 --- NETVIEW FILE TRANSFER PROGRAM VERSION 2.2.1 ---
DVG0011 Generated by macro DVGMXJG2
DVG0011 --- MEMBER : DVGSVP01 ---
DVG0011 --- Parameter: ---
DVG0011 --- APLSVP F2CA6V ---
DVG0011 --- FUNCTION : NFTP server startup parameters ---
DVG0011 ******************************************************
DVG0011 LUNAME=F2CA6V01
DVG0011
DVG0011 CLASS=1
DVG0011
DVG0011 SRVMODE=CONT
DVG0011
DVG0011 * no termination time specified, server runs
DVG0011 * indefinitely
DVG0011 * TTIME='23:59' stop before midnight
DVG0011 * TTIME=('23:59','99/12/31')
DVG0011
DVG0011 DLOGMOD=FTPBIND
DVG0011
DVG0011 TIMECR=180 = 3 min between regulary checkpoints
DVG0011 * BYTECR=32K
DVG0011 *
DVG0011 GID='DVG.SV1'
DVG0011 *
DVG0011 OBTINTVL=120
DVG0011 *
DVG0011 RETRYDELAY=300
DVG0011 *
DVG0011 REPCLASS=B
DVG0011 *
DVG0011 * PRETRAN=
DVG0011 * POSTTRAN=
DVG0011 * POSTCONV=
DVG0011 *
DVG0011 MAXSESS=16
DVG0011 AVLSESS=16
DVG0011 AVLNRH=14
DVG0011 AVLRRH=14
DVG0011 SECPAR=YES
DVG0011 *
DVG0011 * NVEVENT=1 I/O error on the data set
DVG0011 * NVEVENT=(64,65) File tranfer start, end
DVG0011 * NVEVENT=ALL
DVG0011 *
DVG0011 SUPSTOW=NO
DVG0011 *
DVG0011 * Dont use trace normally,
DVG0011 * severely degrades the performance!
DVG0011 *
DVG0011 *
DVG0011 * TRACE=(GROUP1,CL1) entry exit trace for file transfer manager
DVG0011 * TRACE=(GROUP2,CL1) entry exit trace for file handler
DVG0011 * TRACE=(GROUP3,CL4,CL5) User-Exit Routine Interface
DVG0011 *
DVG0011 ********
```

Figure 107 (Part 1 of 2). Server Log File
DVG0661 PROCESSING STARTED FOR REQUEST NUMBER = 1 ; REQUEST NAME = AKEMVAD1
DVG0691 CONVERSATION ALLOCATED WITH F2CA6V02 AT 13:16:30
DVG0921 FILE TRANSFER STARTED ON LOCAL REQUEST HANDLER FOR REQUEST NUMBER 1 ON 94/10/17 AT 13:16:30
DVG0951 END OF PROCESSING OPTION CHANGED FROM NOT-SPECI TO KEEP
DVG0951 FILE ACCESS OPTION CHANGED FROM NOT-SPECI TO SHARED
DVG2431 DATA SET IS ALLOCATED WITH FILE STATUS OPTION MUSTEXIST AND FILE ACCESS OPTION SHARED
DVG2311 FILE TRANSFER STARTED FROM BEGINNING OF DATA SET
DVG2011 --------
DVG3671 1 MEMBERS COMPLETELY TRANSMITTED
DVG2231 REQUEST ORIGINATOR = UID ; REQUEST RECEIVED ON 94/10/17 AT 13:15:43; SERVER CLASS = 1
DVG3641 CHECKPOINTING INTERVAL = 180 SEC
DVG2551 TRANSFER STARTED ON 94/10/17 AT 13:16:30; TRANSFER ENDED ON 94/10/17 AT 13:16:49
DVG2361 NUMBER OF RECORDS TRANSMITTED = 1000
DVG2541 NUMBER OF BYTES TRANSMITTED = 80031
DVG3591 COMPRESSION METHOD = ADAPT; TRANSMISSION RATE = 5335 ; COMPRESSION PERCENTAGE = 14
DVG5141 DATA ENCRYPTION = NO
DVG12012 CHARACTER DATA CONVERSION = NO
DVG2261 INFORMATION FOR LOCAL SYSTEM; LU NAME = F2CA6V01
DVG2301 TRANSFER PARAMETERS; LOGICAL UNIT TYPE = P; TRANSFER MODE = S
DVG2331 RECORD FORMAT = FB ; BLOCK SIZE = 6160 ; LOGICAL RECORD LENGTH = 80
DVG3601 DATA SET INFORMATION; DATA SET NAME = DVG.NFTP23.SDVGSAM ; DD NAME = DVGFI01 ; FILE ORGANIZATION
DVG3601 = PO ; LABEL TYPE = ; DYNAMIC ALLOCATION REQUESTED = Y
DVG2391 VOLUME SERIAL NUMBER 1 = MUSRT3
DVG2441 SECURITY PARAMETERS; USER ID = UID ; GROUP ID = PUBUSER
DVG2011 --------
DVG2261 INFORMATION FOR REMOTE SYSTEM; LU NAME = F2CA6V02; SERVER GROUP = MVS ; NETVIEW FTP NAME = TESTSYS
DVG2301 TRANSFER PARAMETERS; LOGICAL UNIT TYPE = S; TRANSFER MODE = R
DVG3631 DATA SET INFORMATION; DATA SET NAME = DVG.NFTP23.RECEIVE.DATA.SE1 ; FILE ORGANIZATION = PO ; LABEL TYPE = ;
DVG2391 VOLUME SERIAL NUMBER 1 = MUSRT3
DVG2441 SECURITY PARAMETERS; USER ID = UID ; GROUP ID = PUBUSER
DVG2011 --------
DVG0031 FILE TRANSFER ENDED FOR REQUEST NUMBER 1 ON 94/10/17 AT 13:16:49; RC = 0 ; RSN = 0
DVG2011 --------
DVG3121 SERVER TERMINATION REQUESTED BY QUEUE HANDLER; TERMINATION TYPE = I
DVG0021 NETVIEW FTP V2R2M1; SERVER DVG23001 ENDED ON 94/10/17 AT 13:17:25; RETURN CODE = 0

Figure 107 (Part 2 of 2). Server Log File
Appendix H. Layout of the SMF Records

This chapter contains General-use Programming Interface and Associated Guidance information.

An SMF record consists of the following:

- **Record Header**
  This standard SMF record header is prefixed to all SMF record data section subtypes.

- **Header Extension**
  This optional extension to the SMF record header comprises a variable number of *triplets*. Each triplet contains the offset to a product section, the length of the product section, and the number of product sections.

- **Product Section**
  This section contains details of the product (NetView FTP) and the operating system (MVS) that the product runs on.

- **Data Section**
  This section contains subtypes that report the activity of the following (all SMF subtypes are hexadecimal values):
  - **User interface routines**
    Consists of data section subtypes for the following queue handler commands that are issued by the user interface routines:
    
    1. Add
    2. Delete
    3. Query
    4. Change
    5. Force
    6. Restart.
  
  - **Queue handler**
    Consists of data section subtypes for the following queue handler operations:
    
    11. Rebuild request queue
  
  - **Servers**
    Consists of data section subtypes for the following queue handler commands that are issued by servers:
    
    21. Open
    22. Obtain
    23. Getluname
    24. Notify
    25. Close.

All of the above data sections are written by NetView FTP’s queue handler component and thus activated by the appropriate queue handler startup parameter SMFREC.
If you want SMF data section subtype X'51' to be written, code the SMFREC control statement in the server initialization parameters.

Activities reported directly by the server are as follows:

X'51' Server finished processing a file transfer request

To find out how these commands are used, refer to “How NetView FTP V2.2.1 MVS Works” on page 6. Figure 108 on page 226 shows the layout of an SMF record.

### Interpreting SMF Records

Fields relating to each other in the SMF record subtypes can be identified by their names. For example, the field DVGADSNN contains the data-set name in an ADD record and the field DVGNDSNN contains the data-set name in a NOTIFY record.

**Note:** The ADD and OBTAIN records are written before the server starts processing the files in the request. However, the NOTIFY records are written after the server has started processing the requests.

During ADD and OBTAIN processing, the server has not checked the type of data set it is to process. When it does, it changes the field DVGADSNN, which is specified as Q1.Q2.Q3(MEMBER) to the name of the data set in DVGNDSNN(Q1.Q2.Q3). The server then places the member name into DVGNDMEM(MEMBER) in the NOTIFY record.

The same occurs with GDG data sets, with the exception that the server puts the relative generation specified in the request into DVGNDMEM instead of the member name. If this is not done, the relative generation would no longer be accessible. The field DVGADSNN GDGNAME(+1) would be changed to DVGNDSNN(GDGNAME.G00004V00) and DVGNDMEM (+1) in the NOTIFY record.

The following SMF record sequences can be expected during the processing of:

- Normal transfer request
- Restart transfer request (temporary and permanent error situation)
- Manual restart request.

The contents of DVGNTYPE and DVGNRTYP have been added to distinguish between the different situations.

The normal sequence is:

1. ADD record
2. OBTAIN record
3. NOTIFY record DVGNTYPE=S DVGNRTYP=0
4. NOTIFY record DVGNTYPE=F DVGNRTYP=0

The return and reason code indicates whether the file transfer ended successfully or not.
The sequence in the case of a temporary error is:

1. ADD record
2. OBTAIN record
3. NOTIFY record DVGNTYPE=S DVGNRTYP=0
4a. NOTIFY record DVGNTYPE=S DVGNRTYP=A
4b. OBTAIN record
4c. NOTIFY record DVGNTYPE=S DVGNRTYP=0
5. NOTIFY record DVGNTYPE=F DVGNRTYP=0

In this sequence, item 4 is repeated as often as the request is restarted. The return and reason code indicates whether the file transfer ended successfully or not.

The sequence in the case of manual restart is:

1. ADD record
2. OBTAIN record
3. NOTIFY record DVGNTYPE=S DVGNRTYP=M
4. NOTIFY record DVGNTYPE=F DVGNRTYP=0
5. RESTART record
6. OBTAIN record
7. NOTIFY record DVGNTYPE=S
8. NOTIFY record DVGNTYPE=F

The return and reason code indicates whether the file transfer ended successfully or not. In this case, a transfer request for the same sending and receiving data set has been interrupted and has terminated with a return and reason code. This has left a checkpoint-restart record at the receiving system. If the value of the Restart from Checkpoint parameter is YES, NetView FTP V2.2.1 MVS tries to restart the transfer from the last recorded checkpoint. This sequence corresponds to the restart of the transfer from the last checkpoint.

The sequence in the case of a permanent error that occurs before the first checkpoint is taken is:

1. ADD record
2. OBTAIN record
3. NOTIFY record DVGNTYPE=F DVGNRTYP=0

The return and reason code indicates an unsuccessful file transfer.

The sequence in the case of a permanent error that occurs after the first checkpoint is taken is:

1. ADD record
2. OBTAIN record
3. NOTIFY record DVGNTYPE=S DVGNRTYP=0
4. NOTIFY record DVGNTYPE=F DVGNRTYP=0

The return and reason code indicates an unsuccessful file transfer.

When looking at the SMF records written by the server (subtype X‘51’), you find a record for each time a transfer was handled. If a transfer is restarted (more than once), you find one record for each transfer attempt. The byte count and processor microsecond values always denote one single transfer attempt or partial transfer. Note that SMF records for remotely initiated transfers do not necessarily contain all information that is available for locally initiated transfers, for example, a specified server group name is not available on the responding system.
### Standard SMF Header

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>24</td>
<td>SMFHDR(/zerodot)</td>
<td>Standard SMF record header</td>
</tr>
</tbody>
</table>

### Standard SMF Record Header for Subtype Records

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SMFXXLEN</td>
<td>Record length</td>
</tr>
<tr>
<td>2</td>
<td>SIGNED</td>
<td>2</td>
<td>SMFXXSEG</td>
<td>Segment descriptor</td>
</tr>
<tr>
<td>4</td>
<td>BITSTRING</td>
<td>1</td>
<td>SMFXXFLG</td>
<td>System indicator flag byte</td>
</tr>
<tr>
<td>5</td>
<td>BITSTRING</td>
<td>1</td>
<td>SMFXXRTY</td>
<td>Record type XX in HEX</td>
</tr>
<tr>
<td>6</td>
<td>CHARACTER</td>
<td>4</td>
<td>SMFXXTME</td>
<td>Time, in hundredths of a second, record was moved to SMF buffer</td>
</tr>
<tr>
<td>10</td>
<td>(A) 4</td>
<td>SMFXXDTE</td>
<td>Date record was moved to SMF buffer in the form /zerodot/zerodotYYDDDF where F is the sign</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CHARACTER</td>
<td>4</td>
<td>SMFXXSID</td>
<td>System identification (taken from SID parameter)</td>
</tr>
<tr>
<td>18</td>
<td>CHARACTER</td>
<td>4</td>
<td>SMFXXSSI</td>
<td>Subsystem identification ('NFTP')</td>
</tr>
<tr>
<td>22</td>
<td>SIGNED</td>
<td>2</td>
<td>SMFXXSTY</td>
<td>Subtype in HEX (same as decimal infix XX)</td>
</tr>
</tbody>
</table>

- .....1 DVGADD "X'01'" ADD subtype
- .....1 DVGDEL "X'02'" DELETE subtype
- .....1 DVGQRY "X'03'" QUERY subtype
- .....1 DVGCHA "X'04'" MODIFY subtype
- .....1 DVGFOR "X'05'" Force subtype
- .....1 DVGRES "X'06'" Restart subtype
- ...1 DVGQRB "X'11'" QUEUE REBUILD subtype
- ...1 DVGMOD "X'12'" OPERATOR MODIFY subtype
- ..1 DVGOPN "X'21'" OPEN subtype
- ..1 DVGOBT "X'22'" OBTAIN subtype
- ..1 DVGGLU "X'23'" GETLUNAME subtype
- ..1 DVGNOT "X'24'" NOTIFY subtype
- ..1 DVGCLS "X'25'" CLOSE subtype

*Figure 108 (Part 1 of 22). SMF Record Layout*
### Standard Header Extension of NetView FTP SMF Record

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>(18) CHARACTER</td>
<td>20</td>
<td>SMFHDX(0)</td>
<td>Record header extension</td>
</tr>
<tr>
<td>24</td>
<td>(18) SIGNED</td>
<td>2</td>
<td>SMFXTRN</td>
<td>Number of triplets in this record. A triplet is a set of offset/length/number values that defines a section of this record.</td>
</tr>
<tr>
<td>26</td>
<td>(1A) SIGNED</td>
<td>2</td>
<td>SMFXRS1</td>
<td>Filler</td>
</tr>
<tr>
<td>28</td>
<td>(1C) SIGNED</td>
<td>4</td>
<td>SMFXPRS</td>
<td>Offset to product section</td>
</tr>
<tr>
<td>32</td>
<td>(20) SIGNED</td>
<td>2</td>
<td>SMFXPRL</td>
<td>Length of product section</td>
</tr>
<tr>
<td>34</td>
<td>(22) SIGNED</td>
<td>2</td>
<td>SMFXPRN</td>
<td>Number of product sections</td>
</tr>
<tr>
<td>36</td>
<td>(24) SIGNED</td>
<td>4</td>
<td>SMFXDSS</td>
<td>Offset to data section</td>
</tr>
<tr>
<td>40</td>
<td>(28) SIGNED</td>
<td>2</td>
<td>SMFXDSL</td>
<td>Length of data section</td>
</tr>
<tr>
<td>42</td>
<td>(2A) SIGNED</td>
<td>2</td>
<td>SMFXDSN</td>
<td>Number of data sections</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>SMFXLRH</td>
<td>&quot;*/SMFCDXX&quot; Length of record header</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0) STRUCTURE</td>
<td>32</td>
<td>SMFXPRO</td>
<td>SMF product section Based on product section pointer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0) CHARACTER</td>
<td>4</td>
<td>SMFXVER</td>
<td>NetView FTP Version, Release, Modification level. The contents of this field is any of the following: - 210 for NetView FTP V2R1M - 220 for NetView FTP V2R2M</td>
</tr>
<tr>
<td>4</td>
<td>(4) CHARACTER</td>
<td>12</td>
<td>SMFXPRD</td>
<td>Product name ('NetView FTP')</td>
</tr>
<tr>
<td>16</td>
<td>(10) CHARACTER</td>
<td>4</td>
<td>SMFXRLS</td>
<td>OS release number and level in the form nnll</td>
</tr>
<tr>
<td>20</td>
<td>(14) CHARACTER</td>
<td>8</td>
<td>SMFXMVS</td>
<td>MVS software level</td>
</tr>
<tr>
<td>28</td>
<td>(1C) CHARACTER</td>
<td>4</td>
<td>SMFXOSI</td>
<td>'OSI ' for OSI request, blanks for SNA request.</td>
</tr>
<tr>
<td>.1</td>
<td>.1</td>
<td>SMFXLRP</td>
<td>&quot;*/SMFXXPRO&quot; Length of product section</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 108 (Part 2 of 22). SMF Record Layout**
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>8</td>
<td>SMFXXADD</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>DBL WORD</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

NetView FTP SMF Record Data Sections

---

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>148</td>
<td>SMFADD()</td>
<td>The originator of the ADD queue handler command:</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGAORIG</td>
<td>If DVGAORIG is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A: Jobname</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T: TSO user ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B: Jobname or TSO user ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>I: TSO user ID</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGARORI</td>
<td>Requestor identification:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A = Application program invoked via batch job</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T = Application program invoked via TSO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B = Batch interface</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>I = Interactive interface</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGAMOD</td>
<td>Transfer mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S = Sending</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R = Receiving</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGARETC</td>
<td>Return code of the ADD queue operation</td>
</tr>
<tr>
<td>12</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGAREAC</td>
<td>Reason code of the ADD queue operation if return code not zero</td>
</tr>
<tr>
<td>14</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGAPRTY</td>
<td>Queuing priority (0 - 9)</td>
</tr>
<tr>
<td>15</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGACLAS</td>
<td>Request class (0 - 9,A - Z)</td>
</tr>
</tbody>
</table>

Figure 108 (Part 3 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGAURI</td>
<td>Request number assigned by queue handler for this request</td>
</tr>
<tr>
<td>20</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGASTIM(0)</td>
<td>Command enter time</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>4</td>
<td>DVGASDTE</td>
<td>Date in the form 00YDDDF where F is the sign</td>
</tr>
<tr>
<td>24</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGASTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>28</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGAORIR</td>
<td>The originator of the file-transfer request</td>
</tr>
<tr>
<td>36</td>
<td>CHARACTER</td>
<td>44</td>
<td>DVGADSNN</td>
<td>File ID (data set name) to be transferred OSI: local filename</td>
</tr>
<tr>
<td>80</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGARDAT(0)</td>
<td>Request receipt time on queue handler</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>4</td>
<td>DVGARDTE</td>
<td>Date in the form 00YDDDF where F is the sign</td>
</tr>
<tr>
<td>84</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGARTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>88</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGAPDS</td>
<td>Number of PDS members selected or excluded</td>
</tr>
<tr>
<td>92</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGARNAM</td>
<td>Remote node ID or remote LU name specified in the request OSI: Application Entity Title</td>
</tr>
<tr>
<td>100</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGANBTD(0)</td>
<td>Not-before date and time: User specified that the request should not be processed before that time</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>4</td>
<td>DVGABDTE</td>
<td>Date in the form 00YDDDF where F is the sign</td>
</tr>
<tr>
<td>104</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGABTME</td>
<td>Time, in hundredths of a second</td>
</tr>
</tbody>
</table>

Figure 108 (Part 4 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>108</td>
<td>(6C) BITSTRING</td>
<td>8</td>
<td>DVGANATD(0)</td>
<td>Not-after date and time: User specified that the request should not be processed after that time</td>
</tr>
<tr>
<td>108</td>
<td>(6C) BITSTRING</td>
<td>4</td>
<td>DVGAADTE</td>
<td>Date in the form /zerodot/zerodotYYDDDF where F is the sign</td>
</tr>
<tr>
<td>112</td>
<td>(70) BITSTRING</td>
<td>4</td>
<td>DVGAATME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>116</td>
<td>(74) CHARACTER</td>
<td>8</td>
<td>DVGADMEM(0)</td>
<td>Member name if a single PDS member name was specified within the DS name or via the PDSMEM control statement or,</td>
</tr>
<tr>
<td>116</td>
<td>(74) CHARACTER</td>
<td>8</td>
<td>DVGADGDG</td>
<td>Relative generation of a GDG data set if it was specified within the DS name</td>
</tr>
<tr>
<td>124</td>
<td>(7C) CHARACTER</td>
<td>2</td>
<td>DVGADTYP(0)</td>
<td>Data set type (file type) Primary part of file type:</td>
</tr>
<tr>
<td></td>
<td>(7C) CHARACTER</td>
<td>1</td>
<td>DVGATYP1</td>
<td>V = VSAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S = Sequential</td>
<td>O = Partitioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U = User</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>(7E) CHARACTER</td>
<td>1</td>
<td>DVGATYP2</td>
<td>Secondary part of file type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>P = PDS</td>
<td>L = Labeled tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U = Unlabeled tape</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>(80) BITSTRING</td>
<td>4</td>
<td>DVGAWAIT</td>
<td>Number of current requests on queue waiting to be processed</td>
</tr>
<tr>
<td>132</td>
<td>(84) BITSTRING</td>
<td>4</td>
<td>DVGAREST</td>
<td>Number of current requests on queue waiting to be processed due to restart</td>
</tr>
<tr>
<td>136</td>
<td>(88) BITSTRING</td>
<td>4</td>
<td>DVGAACT</td>
<td>Number of requests currently processed by a server (Active)</td>
</tr>
<tr>
<td>140</td>
<td>(8C) BITSTRING</td>
<td>4</td>
<td>DVGAFIN</td>
<td>Number of current requests on queue already finished</td>
</tr>
<tr>
<td>144</td>
<td>(90) BITSTRING</td>
<td>4</td>
<td>DVGAUQS</td>
<td>Number of current unused queue slots</td>
</tr>
<tr>
<td>1..1</td>
<td></td>
<td>..</td>
<td>DVGADDLN</td>
<td>&quot;*.SMFXXADD&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 5 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>40</td>
<td>SMFDEL(0)</td>
<td>The originator of the DELETE queue handler command:</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>8</td>
<td>DVGDORIG</td>
<td>The originator of the DELETE queue handler command:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>If DVGDORI is:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A: Jobname</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: TSO user ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B: Jobname or TSO user ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>if submitted via TSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I: TSO user ID</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>1</td>
<td>DVGDORI</td>
<td>Requestor identification:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A = Application program invoked via batch job</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T = Application program invoked via TSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B = Batch Interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I = Interactive interface</td>
</tr>
<tr>
<td>9</td>
<td>(9)</td>
<td>1</td>
<td>DVGDS1</td>
<td>Filler</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>2</td>
<td>DVGDRETC</td>
<td>Return code of the DELETE queue operation</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>2</td>
<td>DVGDREAC</td>
<td>Reason code of the DELETE queue operation if return code not zero</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>2</td>
<td>DVGDRS2</td>
<td>Filler</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>4</td>
<td>DVGDURI</td>
<td>The request identified by this number is to be deleted</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>8</td>
<td>DVGDSTIM(0)</td>
<td>Command enter time</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>4</td>
<td>DVGDSDTE</td>
<td>Date in the form 00YYDDDF where F is the sign</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>4</td>
<td>DVGDSTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>8</td>
<td>DVGDORID</td>
<td>The originator ID for which requests should be deleted</td>
</tr>
<tr>
<td>36</td>
<td>(24)</td>
<td>4</td>
<td>DVGDDEL#</td>
<td># of requests deleted on DELETE ALL command</td>
</tr>
<tr>
<td>..1. 1..</td>
<td></td>
<td></td>
<td></td>
<td>&quot;*-SMFXXDEL&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Note: Records of subtype DELETE are not marked for OSI
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXQRY</td>
<td>SUBTYPE 3: QUERY Queue Operation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>40</td>
<td>SMFQRY(0)</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGQORIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: TSO user ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B: Jobname or TSO user ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>if submitted via TSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I: TSO user ID</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGQORI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A = Application program invoked via batch job</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T = Application program invoked via TSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B = Batch interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I = Interactive interface</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>(9)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGQRS1</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGQRETC</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGQREAC</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGQRS2</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGQURI</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGQSTIM(0)</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td></td>
<td>4</td>
<td>DVGQSDTE</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGQSTME</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGQORID</td>
</tr>
<tr>
<td>36</td>
<td>(24)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGQQRY#</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&quot;*-SMFXQRY&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 7 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>36</td>
<td>SMFCHA</td>
<td>The originator of the MODIFY queue handler command:</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>8</td>
<td>DVGCORIG</td>
<td>If DVGCORI is</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>1</td>
<td>DVGCROI</td>
<td>Requestor identification:</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>1</td>
<td>DVGCRS1</td>
<td>Filler</td>
</tr>
<tr>
<td></td>
<td>(A)</td>
<td>2</td>
<td>DVGCRETC</td>
<td>Return code of the MODIFY queue operation</td>
</tr>
<tr>
<td></td>
<td>(C)</td>
<td>2</td>
<td>DVGCREAC</td>
<td>Reason code of the MODIFY queue operation if return code not zero</td>
</tr>
<tr>
<td></td>
<td>(E)</td>
<td>1</td>
<td>DVGCPRTY</td>
<td>Priority to which the request is to be changed</td>
</tr>
<tr>
<td></td>
<td>(F)</td>
<td>1</td>
<td>DVGCCLAS</td>
<td>Class to which the request is to be changed</td>
</tr>
<tr>
<td></td>
<td>(10)</td>
<td>4</td>
<td>DVGCURI</td>
<td>The request identified by this number is to be modified</td>
</tr>
<tr>
<td></td>
<td>(14)</td>
<td>8</td>
<td>DVGCSTIM</td>
<td>Command enter time</td>
</tr>
<tr>
<td></td>
<td>(14)</td>
<td>4</td>
<td>DVGCSDTE</td>
<td>Date in the form 00YYDDDF where F is the sign</td>
</tr>
<tr>
<td></td>
<td>(18)</td>
<td>4</td>
<td>DVGCSTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td></td>
<td>(1C)</td>
<td>8</td>
<td>DVGCORID</td>
<td>The originator ID for which requests should be changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DVGCALN</td>
<td>&quot;*=SMFXXCHA&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 8 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>36</td>
<td>SMFFOR(0)</td>
<td>The originator of the FORCEDEL queue handler command:</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>8</td>
<td>DVGFORIG</td>
<td>If DVGORORI is A: Jobname T: TSO user ID B: Jobname or TSO user ID if submitted via TSO I: TSO user ID</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>1</td>
<td>DVGRORI</td>
<td>Requestor identification: A = Application program invoked via batch job T = Application program invoked via TSO B = Batch Interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I = Interactive interface</td>
</tr>
<tr>
<td>9</td>
<td>(9)</td>
<td>1</td>
<td>DVGRS1</td>
<td>Filler</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>2</td>
<td>DVGFRET</td>
<td>Return code of the DELETE queue operation</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>2</td>
<td>DVGFREAC</td>
<td>Reason code of the DELETE queue operation if return code not zero</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>2</td>
<td>DVGRS2</td>
<td>Filler</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>4</td>
<td>DVGFURI</td>
<td>The request identified by this number is to be deleted</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>8</td>
<td>DVGFSTIM(0)</td>
<td>Command enter time</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>4</td>
<td>DVGFSDTE</td>
<td>Date in the form /zerodot/zerodotYYDDDF where F is the sign</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>4</td>
<td>DVGFSTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>8</td>
<td>DVGFORID</td>
<td>The originator ID for which requests should be deleted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DVGFORLN</td>
<td>&quot;*-SMFXXFOR&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 9 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(B)</td>
<td>55</td>
<td>SMFRES(0)</td>
<td>The originator of the FORCEDEL queue handler command:</td>
</tr>
<tr>
<td>0</td>
<td>(B)</td>
<td>8</td>
<td>DVGTORIG</td>
<td>If DVGDRORI is A: Jobname, T: TSO user ID, B: Jobname or TSO user ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>if submitted via TSO, I: TSO user ID</td>
</tr>
<tr>
<td>8</td>
<td>(B)</td>
<td>1</td>
<td>DVGTRORI</td>
<td>Requestor identification:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A = Application program invoked via batch job</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T = Application program invoked via TSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B = Batch Interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I = Interactive interface</td>
</tr>
<tr>
<td>9</td>
<td>(B)</td>
<td>1</td>
<td>DVGTRS1</td>
<td>Filler</td>
</tr>
<tr>
<td>10</td>
<td>(B)</td>
<td>2</td>
<td>DVGTRETC</td>
<td>Return code of the DELETE queue operation</td>
</tr>
<tr>
<td>12</td>
<td>(B)</td>
<td>2</td>
<td>DVGTREAC</td>
<td>Reason code of the DELETE queue operation if return code not zero</td>
</tr>
<tr>
<td>14</td>
<td>(B)</td>
<td>2</td>
<td>DVGTRS2</td>
<td>Filler</td>
</tr>
<tr>
<td>16</td>
<td>(B)</td>
<td>4</td>
<td>DVGTURI</td>
<td>The request identified by this number is to be deleted</td>
</tr>
<tr>
<td>20</td>
<td>(B)</td>
<td>8</td>
<td>DVGTSTIM(0)</td>
<td>Command enter time</td>
</tr>
<tr>
<td>20</td>
<td>(B)</td>
<td>4</td>
<td>DVGTSDTE</td>
<td>Date in the form 00YDDD where F is the sign</td>
</tr>
</tbody>
</table>

Figure 108 (Part 10 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGTSTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>28</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGTORID</td>
<td>The originator ID for which requests should be deleted</td>
</tr>
<tr>
<td>36</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGTAEBF(0)</td>
<td>Command enter time</td>
</tr>
<tr>
<td>36</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGTADETE</td>
<td>Date in the form /YYDDDD where F is the sign</td>
</tr>
<tr>
<td>40</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGTNTME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>44</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGTNAFT(0)</td>
<td>Command enter time</td>
</tr>
<tr>
<td>44</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGTADTE</td>
<td>Date in the form /YYDDDD where F is the sign</td>
</tr>
<tr>
<td>48</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGTATME</td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>52</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGTRPNT</td>
<td>Specified restart point</td>
</tr>
<tr>
<td>53</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGTPRTY</td>
<td>Priority to which the request is to be modified</td>
</tr>
<tr>
<td>54</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGTCLAS</td>
<td>Class to which the request is to be modified</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DVGRESLN</td>
<td>&quot;*-SMFXXRES&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 11 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXXQRB</td>
<td></td>
</tr>
</tbody>
</table>

SUBTYPE 11 : Queue Rebuild

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>44</td>
<td>SMFQRB(0)</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGRQHNA</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGRSLOT</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGRFINB</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGRFREB</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGRSTIM(0)</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td></td>
<td>4</td>
<td>DVGRSDTE</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGRSTME</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGRFREA</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGRFINA</td>
</tr>
<tr>
<td>36</td>
<td>(24)</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGRETIM(0)</td>
</tr>
<tr>
<td>36</td>
<td>(24)</td>
<td></td>
<td>4</td>
<td>DVGREDTE</td>
</tr>
<tr>
<td>40</td>
<td>(28)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGREME</td>
</tr>
<tr>
<td>..1. 11..</td>
<td></td>
<td></td>
<td></td>
<td>DVGQRBLN</td>
</tr>
</tbody>
</table>

*Figure 108 (Part 12 of 22). SMF Record Layout*
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXXMOD</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>59</td>
<td>SMFMOD(0)</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGMSVN</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGMRORI</td>
</tr>
<tr>
<td>9</td>
<td>(9)</td>
<td>CHARACTER</td>
<td>5</td>
<td>DVGMRS1</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGMPRTY</td>
</tr>
<tr>
<td>15</td>
<td>(F)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGMCLAS</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGMURI</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>BITSTRING</td>
<td>3</td>
<td>DVGMSVV(0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGMNTAS</td>
</tr>
<tr>
<td>21</td>
<td>(15)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGMNLOC</td>
</tr>
<tr>
<td>22</td>
<td>(16)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGMNRMT</td>
</tr>
<tr>
<td>23</td>
<td>(17)</td>
<td>CHARACTER</td>
<td>36</td>
<td>DVGMNCLS</td>
</tr>
</tbody>
</table>

..11 1.11  DVGMODLN   "*-SMFXXMOD" Length of this data section

Figure 108 (Part 13 of 22): SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXOPN</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>68</td>
<td>SMFOPN(0)</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGOJNAM</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGORORI</td>
</tr>
<tr>
<td>9</td>
<td>(9)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGORS1</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGORETC</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGOREAC</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>CHARACTER</td>
<td>6</td>
<td>DVGORS2</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGOLUNA</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGOSVV(0)</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGoitms</td>
</tr>
<tr>
<td>29</td>
<td>(1D)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGoitas</td>
</tr>
<tr>
<td>30</td>
<td>(1E)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGoiLoc</td>
</tr>
<tr>
<td>31</td>
<td>(1F)</td>
<td>BITSTRING</td>
<td>1</td>
<td>DVGoiRmt</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>CHARACTER</td>
<td>36</td>
<td>DVGoiCls</td>
</tr>
<tr>
<td>.1.</td>
<td>.1.</td>
<td></td>
<td>DVGopnln</td>
<td>&quot;=SMFXOPN&quot;</td>
</tr>
</tbody>
</table>

Figure 108 (Part 14 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0) STRUCTURE</td>
<td>8</td>
<td>SMFXOBT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0) CHARACTER</td>
<td>81</td>
<td>SMFOBT(0)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0) CHARACTER</td>
<td>8</td>
<td>DVGBJNAM</td>
<td>Jobname of the server that issued OBTAIN queue handler command</td>
</tr>
<tr>
<td>8</td>
<td>(8) CHARACTER</td>
<td>1</td>
<td>DVGBRORI</td>
<td>Requestor identification: S = Server</td>
</tr>
<tr>
<td>9</td>
<td>(9) CHARACTER</td>
<td>1</td>
<td>DVGBXMOD</td>
<td>Transfer mode: S = Sending R = Receiving</td>
</tr>
<tr>
<td>10</td>
<td>(A) BITSTRING</td>
<td>2</td>
<td>DVGBRETC</td>
<td>Return code of the OBTAIN queue operation</td>
</tr>
<tr>
<td>12</td>
<td>(C) BITSTRING</td>
<td>2</td>
<td>DVGBREAC</td>
<td>Reason code of the OBTAIN queue operation if return code not zero</td>
</tr>
<tr>
<td>14</td>
<td>(E) CHARACTER</td>
<td>1</td>
<td>DVGBPRTY</td>
<td>Queuing priority (0 - 9)</td>
</tr>
<tr>
<td>15</td>
<td>(F) CHARACTER</td>
<td>1</td>
<td>DVGBCLAS</td>
<td>Request class (0 - 9, A - Z)</td>
</tr>
<tr>
<td>16</td>
<td>(10) BITSTRING</td>
<td>4</td>
<td>DVGBURI</td>
<td>Request number the queue handler found eligible to be served by this server</td>
</tr>
<tr>
<td>20</td>
<td>(14) CHARACTER</td>
<td>8</td>
<td>DVGBLUNA</td>
<td>LU-name of the server that issued the OBTAIN. This is the local server that processes the file-transfer request OSI: Name of server</td>
</tr>
<tr>
<td>28</td>
<td>(1C) CHARACTER</td>
<td>8</td>
<td>DVGBRS1</td>
<td>Filler</td>
</tr>
<tr>
<td>36</td>
<td>(24) CHARACTER</td>
<td>44</td>
<td>DVGBDSNN</td>
<td>Data set name to be transferred OSI: local filename</td>
</tr>
<tr>
<td>80</td>
<td>(50) CHARACTER</td>
<td>1</td>
<td>DVGBCOMP</td>
<td>Compression method used: N = None S = Standard SNA A = Adaptive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DVGObTln</td>
<td>&quot;*-SMFXOBT&quot; Length of this data section</td>
</tr>
</tbody>
</table>

*Figure 108 (Part 15 of 22). SMF Record Layout*
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXXGLU</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>30 SMFGLU(0)</td>
<td>Jobname of the server that issued GETLUNAM queue handler command</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8 DVGGJNAM</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(A)</td>
<td>BITSTRING</td>
<td>2 DVGGRETC</td>
<td>Return code of the GETLUNAM queue operation</td>
</tr>
<tr>
<td>9</td>
<td>(9)</td>
<td>CHARACTER</td>
<td>1 DVGGRORI</td>
<td>Requestor identification: S = Server</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>BITSTRING</td>
<td>2 DVGGREAC</td>
<td>Reason code of the GETLUNAM queue operation if return code not zero</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>CHARACTER</td>
<td>8 DVGGNODE</td>
<td>Node ID</td>
</tr>
<tr>
<td>22</td>
<td>(16)</td>
<td>CHARACTER</td>
<td>4 DVGGOPSX</td>
<td>Remote operating system type</td>
</tr>
<tr>
<td>26</td>
<td>(1A)</td>
<td>CHARACTER</td>
<td>3 DVGLVL</td>
<td>Remote FTP level</td>
</tr>
<tr>
<td>29</td>
<td>(1D)</td>
<td>CHARACTER</td>
<td>1 DVGGSMOD</td>
<td>Remote server running mode</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td>&quot;*=SMFXXGLU&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 16 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXXNOT</td>
<td>SUBTYPE 24 : NOTIFY queue operation</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER 148</td>
<td>SMFNOT(0)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER 8</td>
<td>DVGNJNAM</td>
<td>Jobname of the server that issued NOTIFY queue handler command</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER 1</td>
<td>DVGNRORI</td>
<td>Requestor identification:</td>
</tr>
<tr>
<td>9</td>
<td>(9)</td>
<td>CHARACTER 1</td>
<td>DVGNXMOD</td>
<td>Transfer mode:</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>BITSTRING 2</td>
<td>DVGNRETC</td>
<td>Return code of the NOTIFY queue operation</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>BITSTRING 2</td>
<td>DVGNREAC</td>
<td>Reason code of the NOTIFY queue operation</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>CHARACTER 1</td>
<td>DVGNPRTY</td>
<td>Queuing priority (0 - 9)</td>
</tr>
<tr>
<td>15</td>
<td>(F)</td>
<td>CHARACTER 1</td>
<td>DVGNCLAS</td>
<td>Request class (0 - 9,A - Z)</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>BITSTRING 4</td>
<td>DVGNURI</td>
<td>Request number currently being processed by this server</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>CHARACTER 8</td>
<td>DVGNLUNA</td>
<td>LU name of the server that issued the NOTIFY.</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>BITSTRING 4</td>
<td>DVGNBYTE</td>
<td>Transfer byte rate (data bytes per second)</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>BITSTRING 2</td>
<td>DVGNTRRC</td>
<td>Return code of the finished file transfer</td>
</tr>
</tbody>
</table>

Figure 108 (Part 17 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGNTRSC</td>
<td>Reason code of the finished file transfer if return code not zero</td>
</tr>
<tr>
<td>36</td>
<td>CHARACTER</td>
<td>44</td>
<td>DVGNDSNN</td>
<td>Data set name transferred OS1: local filename</td>
</tr>
<tr>
<td>80</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGNCOMP</td>
<td>Compression method used: N = none S = Standard SNA A = Adaptive</td>
</tr>
<tr>
<td>81</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGNCFAC</td>
<td>Compression factor. This value is a binary integer from 0 to 100 used as a percentage expression.</td>
</tr>
<tr>
<td>82</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGNTYPE</td>
<td>Type of NOTIFY: S = Transfer start (NOTIFY restartable) F = Transfer finished (NOTIFY finished)</td>
</tr>
<tr>
<td>83</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGNRTYP</td>
<td>Type of restart: If transfer was restarted then: A = Automatic transfer restart or logon retry M = Manual transfer restart</td>
</tr>
<tr>
<td>84</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGNRLUN</td>
<td>LUNAME of remote server currently involved in the file-transfer request OS1: Applic.Entity Title</td>
</tr>
<tr>
<td>92</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGNRNOD</td>
<td>Remote node name specified by the request originator</td>
</tr>
<tr>
<td>100</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGNXMST(0)</td>
<td>Time and date the file-transfer request started</td>
</tr>
</tbody>
</table>

Figure 108 (Part 18 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>(64)</td>
<td>4</td>
<td>DVGNSDTE</td>
<td>Date in the form 00Y0DDDF where F is the sign</td>
</tr>
<tr>
<td>104</td>
<td>(68)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNSTME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>108</td>
<td>(6C)</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGNXMET(0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time and date the file-transfer request ended</td>
</tr>
<tr>
<td>108</td>
<td>(6C)</td>
<td>4</td>
<td>DVGNEDTE</td>
<td>Date in the form 00Y0DDDF where F is the sign</td>
</tr>
<tr>
<td>112</td>
<td>(70)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNETME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time, in hundredths of a second</td>
</tr>
<tr>
<td>116</td>
<td>(74)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGNDMEM(0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Member name if a single PDS member name was specified within the DS name or via the PDSMEM control statement or</td>
</tr>
<tr>
<td>116</td>
<td>(74)</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGNDGDG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Relative generation of a GDG data set if it was specified within the DS name</td>
</tr>
<tr>
<td>124</td>
<td>(7C)</td>
<td>CHARACTER</td>
<td>2</td>
<td>DVGNDTYP(0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Data set type (file type)</td>
</tr>
<tr>
<td>124</td>
<td>(7C)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGNTYP1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Primary part of file type: V = VSAM S = Sequential O = Partitioned U = User</td>
</tr>
<tr>
<td>125</td>
<td>(7D)</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGNTYP2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Secondary part of file type: P = PDS L = Labeled tape U = Unlabeled tape</td>
</tr>
<tr>
<td>126</td>
<td>(7E)</td>
<td>CHARACTER</td>
<td>2</td>
<td>DVGNRS1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Filler</td>
</tr>
<tr>
<td>128</td>
<td>(80)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNWAIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of current requests on queue waiting to be processed</td>
</tr>
<tr>
<td>132</td>
<td>(84)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNREST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of current requests on queue waiting to be processed due to restart</td>
</tr>
<tr>
<td>136</td>
<td>(88)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNACT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of requests currently processed by a server</td>
</tr>
<tr>
<td>140</td>
<td>(8C)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNFIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of current requests on queue already finished</td>
</tr>
<tr>
<td>144</td>
<td>(90)</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGNUQS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of current unused queue slots</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&quot;*-SMFXXN0T&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 19 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>SMFXXCLS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>CHARACTER</td>
<td>SMFCLS(0)</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>DVGLJNAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>DVGLRORI</td>
<td>Requestor identification: S = Server</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>BITSTRING</td>
<td>DVGLRS1</td>
<td>Filler</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>BITSTRING</td>
<td>DVGLREAC</td>
<td>Return code of the CLOSE queue handler command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>DVGCLSLN</td>
<td>&quot;*-SMFXXCLS&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 20 of 22). SMF Record Layout
## SMF Record Layout

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>190</td>
<td>SMFSFR(0)</td>
<td>Jobname of the server</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGSJNAM</td>
<td>Jobname of the server</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSXMOD</td>
<td>Transfer mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- S = Sending</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- R = Receiving</td>
</tr>
<tr>
<td>9</td>
<td>CHARACTER</td>
<td>3</td>
<td>DVGSROLE</td>
<td>Role of the server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- REQ = requesting server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- RSP = responding server</td>
</tr>
<tr>
<td>12</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGSURI</td>
<td>Request number</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSPRTY</td>
<td>Queueing priority (0-9)</td>
</tr>
<tr>
<td>17</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSCLAS</td>
<td>Request class (0-9,A-Z)</td>
</tr>
<tr>
<td>18</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGSQNMM</td>
<td>Request name</td>
</tr>
<tr>
<td>26</td>
<td>CHARACTER</td>
<td>10</td>
<td>DVGSROI</td>
<td>Originator ID</td>
</tr>
<tr>
<td>36</td>
<td>CHARACTER</td>
<td>17</td>
<td>DVGSLLUNA</td>
<td>LU name of lcl. server</td>
</tr>
<tr>
<td>53</td>
<td>CHARACTER</td>
<td>17</td>
<td>DVGSRLUN</td>
<td>LU name of rmt. server</td>
</tr>
<tr>
<td>70</td>
<td>CHARACTER</td>
<td>10</td>
<td>DVGSNOD</td>
<td>Server group name</td>
</tr>
<tr>
<td>80</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGSTRC</td>
<td>Transfer Reason Code</td>
</tr>
<tr>
<td>82</td>
<td>BITSTRING</td>
<td>2</td>
<td>DVGSTRC</td>
<td>Transfer Reason Code</td>
</tr>
<tr>
<td>84</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSRTYP</td>
<td>Type of transfer:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- I = Initial transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- A = Automatic restart</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- M = Manual restart</td>
</tr>
<tr>
<td>85</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSRPNT</td>
<td>Restart Point:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- B = from beginning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- C = from checkpoint</td>
</tr>
<tr>
<td>86</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSENC</td>
<td>Encryption active (Y=Yes)</td>
</tr>
<tr>
<td>87</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSNCR</td>
<td>Encryption active (Y=Yes)</td>
</tr>
<tr>
<td>88</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSNCR</td>
<td>Encryption active (Y=Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N = none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S = Standard SNA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A = Adaptive</td>
</tr>
<tr>
<td>89</td>
<td>CHARACTER</td>
<td>1</td>
<td>DVGSDCFAC</td>
<td>Compression method used:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N = none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S = Standard SNA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A = Adaptive</td>
</tr>
<tr>
<td>90</td>
<td>CHARACTER</td>
<td>2</td>
<td>DVGSFIL1</td>
<td>Filler</td>
</tr>
<tr>
<td>92</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGSBYTE</td>
<td>Transfer rate (Bytes/sec)</td>
</tr>
<tr>
<td>96</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGSBCNT</td>
<td>Transfer byte count</td>
</tr>
<tr>
<td>104</td>
<td>CHARACTER</td>
<td>8</td>
<td>DVGSCPUT</td>
<td>CPU time used (Microsec)</td>
</tr>
<tr>
<td>112</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGSXMT(0)</td>
<td>Transfer Start Time+Date</td>
</tr>
<tr>
<td>112</td>
<td></td>
<td></td>
<td></td>
<td>Date, format: 00YYDDDF</td>
</tr>
<tr>
<td>116</td>
<td>BITSTRING</td>
<td>4</td>
<td>DVGSSTME</td>
<td>Time, in 100ths of a sec</td>
</tr>
<tr>
<td>120</td>
<td>BITSTRING</td>
<td>8</td>
<td>DVGSXMT(0)</td>
<td>Transfer End Time &amp; Date</td>
</tr>
</tbody>
</table>

Figure 108 (Part 21 of 22). SMF Record Layout
<table>
<thead>
<tr>
<th>Offsets</th>
<th>Type</th>
<th>Length</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>(78)</td>
<td>4</td>
<td>DVGSEDTE</td>
<td>Date in the form 00YYDDDF</td>
</tr>
<tr>
<td>124</td>
<td>(7C)</td>
<td>4</td>
<td>DVGSETME</td>
<td>Time, in 100ths of a sec</td>
</tr>
<tr>
<td>128</td>
<td>(80)</td>
<td>8</td>
<td>DVGSDDNM</td>
<td>Local DDName</td>
</tr>
<tr>
<td>136</td>
<td>(88)</td>
<td>44</td>
<td>DVGSDSNN</td>
<td>Local data set name</td>
</tr>
<tr>
<td>180</td>
<td>(B4)</td>
<td>8</td>
<td>DVGSDEMEM(0)</td>
<td>Member name</td>
</tr>
<tr>
<td>188</td>
<td>(B4)</td>
<td>8</td>
<td>DVGSDDG DG</td>
<td>Rel. generation of a GDG</td>
</tr>
<tr>
<td>188</td>
<td>(B4)</td>
<td>2</td>
<td>DVGSDDTYP(0)</td>
<td>Data set type (file type)</td>
</tr>
<tr>
<td>189</td>
<td>(BD)</td>
<td>1</td>
<td>DVGSTYP1</td>
<td>1st part of file type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- V = VSAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- S = Sequential</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- O = Partitioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- U = User</td>
</tr>
<tr>
<td>189</td>
<td>(BD)</td>
<td>1</td>
<td>DVGSTYP2</td>
<td>2nd part of file type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- P = PDS Member</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- L = Labeled Tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- U = Unlabeled Tape</td>
</tr>
<tr>
<td>1.11</td>
<td>111.</td>
<td></td>
<td>DVGSFRLN</td>
<td>&quot;*-SMFXXSFR&quot; Length of this data section</td>
</tr>
</tbody>
</table>

Figure 108 (Part 22 of 22). SMF Record Layout
Appendix I. Using NetView in Conjunction with NetView FTP V2.2.1 MVS

This appendix describes how you can use NetView in conjunction with NetView FTP V2.2.1 MVS.

Introduction

To enable automated operations, NetView FTP V2.2.1 MVS writes messages for each of the events described in “Automated Operation of NetView FTP V2.2.1 MVS” by issuing multiline Write to Operator (WTO) messages. These messages have the following special properties:

- They are issued to the operator’s console
- They are written with multi-line WTO
- They have a fixed, well defined, and automatically interpretable structure that is not subject to changes or translation due to National Language Support.

These properties do not make the messages easy to read by an operator, so a means is provided to suppress the messages when there is no NetView Command List (CLIST) to filter them out.

You can partly automate the operation of NetView FTP V2.2.1 MVS if you have active NetView CLIST or REXX procedures that can interpret messages and perform the corresponding action in NetView FTP V2.2.1 MVS.

This appendix describes only the multiline WTOs written by NetView FTP V2.2.1 MVS and not the NetView CLIST or REXX procedures.

Automated Operation of NetView FTP V2.2.1 MVS

Messages are provided in the form of multiline WTOs for the following events provided to the NetView CLIST. All of the events are divided into three groups. These are the events issued by the queue handler, the server, or both. The following lists all of the events issued by NetView FTP V2.2.1 MVS:

1. Events issued by the queue handler and the SNA server:

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I/O error on the data set</td>
</tr>
<tr>
<td>2</td>
<td>VSAM error on the request queue, checkpoint data set, sending data set, or receiving data set</td>
</tr>
<tr>
<td>3</td>
<td>Abend during I/O.</td>
</tr>
</tbody>
</table>
2. Events issued by the queue handler only:

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Queue handler ready</td>
</tr>
<tr>
<td>22</td>
<td>Queue handler stopped</td>
</tr>
<tr>
<td>23</td>
<td>Abnormal termination of the queue handler</td>
</tr>
<tr>
<td>24</td>
<td>Request queue status</td>
</tr>
<tr>
<td>25</td>
<td>Server status</td>
</tr>
<tr>
<td>26</td>
<td>Request queue rebuilt</td>
</tr>
<tr>
<td>27</td>
<td>File-transfer request added.</td>
</tr>
</tbody>
</table>

3. Events issued by the SNA server only:

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>Server ready</td>
</tr>
<tr>
<td>62</td>
<td>Server stopped</td>
</tr>
<tr>
<td>63</td>
<td>Abnormal termination of a server</td>
</tr>
<tr>
<td>64</td>
<td>File transfer started</td>
</tr>
<tr>
<td>65</td>
<td>File transfer ended</td>
</tr>
<tr>
<td>66</td>
<td>Security program error during user validation</td>
</tr>
<tr>
<td>67</td>
<td>Dynamic allocation error during file access.</td>
</tr>
</tbody>
</table>

**Note:** OSI servers do not issue events.

---

**The Startup Parameter**

NVEVENT is the startup parameter to activate specific events for the queue handler and server. NVEVENT is specified as follows:

- `NVEVENT = ALL`
- `NVEVENT = Event_No`
- `NVEVENT = (Event_No,...,Event_No)`

The Event_No is an identifier for each event to be issued. How it is defined is described in “Description of Multiline WTOs” on page 251. Valid values for Event_IDs are 1 through 3 for events issued by both the queue handler and the server, 21 through 27 for events issued by the queue handler only, and 61 through 67 for events issued by the server only.

Specifying this statement is optional. If it is not specified, NetView FTP V2.2.1 MVS provides no NetView event.
Description of Multiline WTOs

The following describes the form of each of the multiline WTOs being issued as a NetView event:

All events issued by NetView FTP V2.2.1 MVS are of the same form:

```
  HEADER  Event_ID  PARAMETERS
```

where:

**HEADER**  
This is a uniquely defined word DVG670I and is an identifier for a CLIST or REXX procedure that scans the messages written to the operator's console for any of the events issued by NetView FTP V2.2.1 MVS, which are described in "Automated Operation of NetView FTP V2.2.1 MVS" on page 249.

**Event_ID**  
This specifies the special event issued by NetView FTP V2.2.1 MVS.

**PARAMETERS**  
These pass the information in the form of keyword(value) to the CLIST or REXX procedure that is to perform the automated operation. The parameters must be separated from one another by at least one blank.

The Event_No denotes the identification number for the startup parameter. All keywords and field names are explicitly listed. A comment is given in parenthesis for each parameter.

Events Issued by NetView FTP V2.2.1 MVS

The following describes the events issued by NetView FTP V2.2.1 MVS.

Events Issued by the Queue Handler and the SNA Server

**I/O Error on the Data Set**  
This event is issued if the SYNAD error analysis routine of the file handler gets control.

```
  Event_ID  (Event_No)
  IOERR(1)
```
Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>(name of the job issuing the event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT</td>
<td>(unit address of device in error)</td>
</tr>
<tr>
<td>DEVC</td>
<td>(device type)</td>
</tr>
<tr>
<td></td>
<td>DA - direct access device</td>
</tr>
<tr>
<td></td>
<td>TA - magnetic tape device</td>
</tr>
<tr>
<td></td>
<td>UR - unit record device</td>
</tr>
<tr>
<td>DDNAME</td>
<td>(DD name of data set)</td>
</tr>
<tr>
<td>OPER</td>
<td>(operation attempted)</td>
</tr>
<tr>
<td>BLCK</td>
<td>(device type dependent)</td>
</tr>
<tr>
<td></td>
<td>if DA - actual track address and block number in hexadecimal</td>
</tr>
<tr>
<td></td>
<td>if TA - relative block number in decimal</td>
</tr>
<tr>
<td></td>
<td>if UR - asterisks.</td>
</tr>
<tr>
<td>ACCESS</td>
<td>(access method)</td>
</tr>
</tbody>
</table>

The values of the UNIT, DEVC, DDNAME, OPER, BLCK, and ACCESS keywords are results of the SYNADAF error analysis macroinstruction.

VSAM Error
This event is issued if an error occurs in a VSAM data set. It can be in any of the following:

- The request-queue of the queue handler
- The checkpoint data set of the server
- The sending or receiving data set.

Event_ID (Event_No)
VSAMER(2)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>(name of the job issuing the event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPER</td>
<td>(operation attempted)</td>
</tr>
<tr>
<td>RC</td>
<td>(VSAM return code)</td>
</tr>
<tr>
<td>DDNAME</td>
<td>(DD name of data set)</td>
</tr>
</tbody>
</table>

Abend During I/O
This event is issued if an abnormal termination occurs during an I/O operation.

Event_ID (Event_No)
IOABEND(3)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>(name of the job issuing the event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPER</td>
<td>(operation attempted)</td>
</tr>
<tr>
<td>LOCDSN</td>
<td>(data set name)</td>
</tr>
<tr>
<td>SYSABCD</td>
<td>(system abend code)</td>
</tr>
</tbody>
</table>
Events Issued by the Queue Handler Only

**Queue Handler Ready**
This event is issued upon completion of initialization of the queue handler. It passes the queue handler startup parameters.

**Event_ID (Event_No)**
QHREADY(21)

**Keyword (Value)**
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>MAXSRV</td>
<td>(maximum number of servers allowed (decimal))</td>
</tr>
<tr>
<td>NFTPNAME</td>
<td>(installation name)</td>
</tr>
<tr>
<td>SRVPREF</td>
<td>(server prefix)</td>
</tr>
<tr>
<td>DELAY</td>
<td>(minimum amount of time requests are to be left in the request queue after they have been given the status finished (in hours, decimal))</td>
</tr>
<tr>
<td>STRSV1..5</td>
<td>(suffixes of automatically started servers, separated by commas)</td>
</tr>
<tr>
<td>FSBSLOTS</td>
<td>(total and TSO user reserved FSB slots (decimal, separated by commas))</td>
</tr>
<tr>
<td>CSAELIM</td>
<td>(CSA storage limit for requests generating an FSBX (in bytes, decimal))</td>
</tr>
<tr>
<td>SMFREC</td>
<td>(SMF record number, decimal)</td>
</tr>
<tr>
<td>TRC</td>
<td>(ON</td>
</tr>
<tr>
<td>NVEVENT</td>
<td>(event numbers of the NetView events issued by the queue handler (separated by commas))</td>
</tr>
</tbody>
</table>

**Queue Handler Stopped**
This event is issued upon termination of the queue handler.

**Event_ID (Event_No)**
QHTERM(22)

**Keyword (Value)**
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>RC</td>
<td>(queue handler return code, decimal)</td>
</tr>
</tbody>
</table>

**Abnormal Termination of the Queue Handler**
This event is issued if an abnormal termination occurs in the queue handler.

**Event_ID (Event_No)**
QHABEND(23)

**Keyword (Value)**
The event passes the following parameter:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
</tbody>
</table>
Request Queue Status
This event reflects the status of the request queues. It is issued whenever the request queue status changes.

Event_ID (Event_No)
TRQST(24)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>TRQUN</td>
<td>(number of unused request queue slots)</td>
</tr>
<tr>
<td>TRQFR</td>
<td>(number of finished jobs)</td>
</tr>
<tr>
<td>TRQRS</td>
<td>(number of waiting requests due to start)</td>
</tr>
<tr>
<td>TRQRC</td>
<td>(number of requests due for remote checking)</td>
</tr>
<tr>
<td>TRQWR1</td>
<td>(number of waiting requests, classes 0 - 9)</td>
</tr>
<tr>
<td>TRQWR2</td>
<td>(number of waiting requests, classes A - J)</td>
</tr>
<tr>
<td>TRQWR3</td>
<td>(number of waiting requests, classes K - T)</td>
</tr>
<tr>
<td>TRQWR4</td>
<td>(number of waiting requests, classes U - Z)</td>
</tr>
<tr>
<td>TRQAR1</td>
<td>(number of active requests, classes 0 - 9)</td>
</tr>
<tr>
<td>TRQAR2</td>
<td>(number of active requests, classes A - J)</td>
</tr>
<tr>
<td>TRQAR3</td>
<td>(number of active requests, classes K - T)</td>
</tr>
<tr>
<td>TRQAR4</td>
<td>(number of active requests, classes U - Z)</td>
</tr>
</tbody>
</table>

General Comments
All numbers are given in decimal. If several values are related to one keyword, they are separated by commas.

The event that NetView FTP V2.2.1 MVS issued when the queue handler started providing information about the initial state of the request queue. When a new request is entered, a request is deleted, or a request is scheduled, the status of the request queue changes and that change is passed with this event.

There are several applications for this event:

- Network-wide request-queue status
- Tailoring of server classes depending on the load in the different request classes.

The event is issued each time the TRQ is updated.

Server Status
This event is issued by the queue handler after a server has completed its initialization procedure, and whenever the operator issues a request to modify a server.

Event_ID (Event_No)
SVST(25)
Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>SVNAME</td>
<td>(server job name)</td>
</tr>
<tr>
<td>LUNAME</td>
<td>(server LU name)</td>
</tr>
<tr>
<td>AVLSESS</td>
<td>(maximum number of available sessions)</td>
</tr>
<tr>
<td>AVLRRH</td>
<td>(maximum number of available primary sessions)</td>
</tr>
<tr>
<td>AVLRRH2</td>
<td>(maximum number of available secondary sessions)</td>
</tr>
<tr>
<td>CLASS</td>
<td>(classes served, given as one character string not separated by commas)</td>
</tr>
<tr>
<td>STATUS</td>
<td>(server status H(old) or R(elease))</td>
</tr>
</tbody>
</table>

Request Queue Rebuilt
This event is issued by the queue handler after the operator issued a request to rebuild the request queue.

Event_ID (Event_No)
TRQRB(26)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>TRQUN</td>
<td>(number of unused request queue slots)</td>
</tr>
<tr>
<td>TRQFR</td>
<td>(number of finished requests)</td>
</tr>
<tr>
<td>DELAY</td>
<td>(current delay time used, in hours)</td>
</tr>
</tbody>
</table>

All numbers are given in decimal.
File-Transfer Request Added
This event is issued after the queue handler has added a request to the request queue.

Event_ID (Event_No)
FTADD(27)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>(name of the job issuing the event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQNO</td>
<td>(request number, decimal)</td>
</tr>
<tr>
<td>REQNAM</td>
<td>(request name)</td>
</tr>
<tr>
<td>ORIG</td>
<td>(request originator)</td>
</tr>
<tr>
<td>XMODE</td>
<td>(transfer Mode, S (sending), R (receiving))</td>
</tr>
<tr>
<td>REMLUN</td>
<td>(remote LU_name)</td>
</tr>
<tr>
<td>LOCDSN</td>
<td>(local data set name)</td>
</tr>
<tr>
<td>REMDSN</td>
<td>(remote data set name)</td>
</tr>
<tr>
<td>REMNODE</td>
<td>(remote node name)</td>
</tr>
<tr>
<td>LOCLUN</td>
<td>(local LU name)</td>
</tr>
<tr>
<td>CLASS</td>
<td>(class of the file-transfer request)</td>
</tr>
<tr>
<td>PRTY</td>
<td>(priority of the file-transfer request)</td>
</tr>
<tr>
<td>NOTBEFORE</td>
<td>(not before time)</td>
</tr>
<tr>
<td>NOTAFTER</td>
<td>(not after time)</td>
</tr>
<tr>
<td>REMCHECK</td>
<td>(remote check option (Y or N))</td>
</tr>
<tr>
<td>SSRVMODE</td>
<td>(sending server’s running mode S (single), C (continuous), G (continuous, GDG))</td>
</tr>
<tr>
<td>RSRVMODE</td>
<td>(receiving server’s running mode S (single), C (continuous), G (continuous, GDG))</td>
</tr>
<tr>
<td>REQST</td>
<td>(status of the file-transfer request, R(elease) or H(old))</td>
</tr>
</tbody>
</table>

Events Issued by the SNA Server Only

Server Ready
This event is issued on completion of a server’s initialization. The parameters passed reflect the server’s startup parameters.

Event_ID (Event_No)
SVSTRT(61)
Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword (Value)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>LUNAME</td>
<td>(LU_name of server)</td>
</tr>
<tr>
<td>TTIME</td>
<td>(server termination date and time in the format YY/MM/DD, HH:MM:SS)</td>
</tr>
<tr>
<td>RUSIZE</td>
<td>(size of RU (decimal, in bytes))</td>
</tr>
<tr>
<td>BYTECR</td>
<td>(number of bytes between two checkpoint restart records in decimal)</td>
</tr>
<tr>
<td>TIMECR</td>
<td>(elapsed time between two checkpoint restart records (in seconds, decimal))</td>
</tr>
<tr>
<td>PRETRAN</td>
<td>(module name of pre-transfer user exit)</td>
</tr>
<tr>
<td>POSTTRAN</td>
<td>(module name of post-transfer user exit)</td>
</tr>
<tr>
<td>POSTCONV</td>
<td>(module name of post-conversation user exit)</td>
</tr>
<tr>
<td>GID</td>
<td>(default 1st qualifier for data set name creation)</td>
</tr>
<tr>
<td>CLASS</td>
<td>(server classes (specified as one character string))</td>
</tr>
<tr>
<td>TRC</td>
<td>(ON</td>
</tr>
<tr>
<td>MAXSESS</td>
<td>(maximum number of subtasks)</td>
</tr>
<tr>
<td>SRVMODE</td>
<td>(server's running mode S (single), C (continuous), G (continuous, GDG))</td>
</tr>
<tr>
<td>OBTINTVL</td>
<td>(obtain interval (in seconds, decimal))</td>
</tr>
<tr>
<td>REQDELAY</td>
<td>(requeue delay time (in seconds, decimal))</td>
</tr>
<tr>
<td>NVEVENT</td>
<td>(NetView events issued by server (separated by commas))</td>
</tr>
</tbody>
</table>

Server Stopped
This event is issued upon termination of the server.

Event_ID (Event_No)
SVTERM(52)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Keyword (Value)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>RC</td>
<td>(server return code)</td>
</tr>
</tbody>
</table>

Abnormal Termination of a Server
This event is issued if an abnormal termination occurs in the server.

Event_ID (Event_No)
SVABEND(63)

Keyword (Value)
The event passes the following parameter:

<table>
<thead>
<tr>
<th>Keyword (Value)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
</tbody>
</table>
File Transfer Started
This event is issued at the beginning of a file transfer.

Event_ID (Event_No)
FTSTRT(64)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>REQNO</td>
<td>(request number, decimal)</td>
</tr>
<tr>
<td>REQNAM</td>
<td>(request name)</td>
</tr>
<tr>
<td>ORIG</td>
<td>(request originator)</td>
</tr>
<tr>
<td>XMODE</td>
<td>(transfer Mode, S (sending), R (receiving))</td>
</tr>
<tr>
<td>REMLUN</td>
<td>(remote LU_name)</td>
</tr>
<tr>
<td>LOCDSN</td>
<td>(local data set name)</td>
</tr>
<tr>
<td>REMDSN</td>
<td>(remote data set name)</td>
</tr>
<tr>
<td>ROLE</td>
<td>(role during file transfer, REQUESTOR, RESPONDER)</td>
</tr>
</tbody>
</table>

File Transfer Ended
This event is issued at the end of a file transfer.

Event_ID (Event_No)
FTEND(65)

Keyword (Value)
The event passes the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>(name of the job issuing the event)</td>
</tr>
<tr>
<td>REQNO</td>
<td>(request number, decimal)</td>
</tr>
<tr>
<td>REQNAM</td>
<td>(request name)</td>
</tr>
<tr>
<td>ORIG</td>
<td>(request originator)</td>
</tr>
<tr>
<td>XMODE</td>
<td>(transfer mode, S (sending), R (receiving))</td>
</tr>
<tr>
<td>REMLUN</td>
<td>(remote LU_Name)</td>
</tr>
<tr>
<td>LOCDSN</td>
<td>(local data set name)</td>
</tr>
<tr>
<td>REMDSN</td>
<td>(remote data set name)</td>
</tr>
<tr>
<td>RC</td>
<td>(transfer return code)</td>
</tr>
<tr>
<td>RSN</td>
<td>(transfer reason code)</td>
</tr>
<tr>
<td>REQST</td>
<td>(request status, F (finished), W (waiting))</td>
</tr>
<tr>
<td>ROLE</td>
<td>(role during file transfer, REQUESTOR, RESPONDER)</td>
</tr>
</tbody>
</table>
Security Program Error During User Validation
This event is issued if the required security parameters of the request are invalid.

**Event ID (Event No)**
RACFUV(66)

**Keyword (Value)**
The event passes the following parameters:

- **JOBNAME** (name of the job issuing the event)
- **LOCUID** (local user ID)
- **LOCGID** (local group ID)
- **REQNO** (request number, decimal)
- **REQNAM** (request name)
- **XMODE** (transfer mode, S (sending), R (receiving))
- **RC** (RACF return code)
- **RSN** (RACF reason code)

**General Comments**
The event provides information about an unauthorized attempt to log on to a system.

Dynamic Allocation Error During File Access
This event is issued if a dynamic allocation error occurs during file access.

**Event ID (Event No)**
DYNALL(67)

**Keyword (Value)**
The event passes the following parameters:

- **JOBNAME** (name of the job issuing the event)
- **LOCDSN** (data set name)
- **XMODE** (transfer mode, S (sending), R (receiving))
- **REQNO** (request number, decimal)
- **REQNAM** (request name)
- **RC** (dynamic allocation return code (decimal))
- **S99ERR** (dynamic allocation error code (hexadecimal))
- **S99INFO** (dynamic allocation info code (hexadecimal))

**General Comments**
The event provides information about a dynamic allocation error for the sending or receiving data set.

Unauthorized attempts to gain access to a data set is given by error code S99ERR (47AC). For more information, refer to MVS/ESA System Programming Library: Application Development Guide MVS System/Product: JES2 Version 2 JES3 Version 3.
Sample Log Files with NetView Events

The following is a sample of a Queue Handler Log File.

```
DVG670I TRQST JOBNAME(DVG230QH) TRQUN(98) TRQRS(/) TRQFR(1) TRQRC(/)
TRQWR1(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQWR2(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQWR3(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQWR4(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
TRQAR1(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQAR2(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQAR3(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQAR4(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
DVG670I QHTERM JOBNAME(DVG230QH) TRQUN(98) TRQRS(/) TRQFR(1) TRQRC(/)
TRQWR1(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQWR2(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQWR3(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQWR4(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
TRQAR1(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQAR2(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQAR3(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) TRQAR4(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
```

```
Figure 109. Queue Handler Log File with NetView Events

The following is a sample of a Server Log File.

```
DVG670I SVSTRT JOBNAME(DVG23001) LUNAME(F2CA6V01) AVLSESS(16) AVLRLR(14) AVLRR(14) CLASS(1) STATUS(R)
DVG670I SVSTRT JOBNAME(DVG23002) LUNAME(F2CA6V02) AVLSESS(16) AVLRLR(14) AVLRR(14) CLASS(2) STATUS(R)
DVG670I SVTERM JOBNAME(DVG23001) RC(/)
DVG670I SVTERM JOBNAME(DVG23002) RC(/)
```

```
Figure 110. Server Log File with NetView Events
```
Appendix J. Macro List

The macros identified in this appendix are provided as programming interfaces for customers by NetView FTP V2.2.1 MVS.

**Warning:** Do not use as programming interfaces any NetView FTP V2.2.1 MVS macros other than those identified in this appendix.

List of executable macros:
- DVG$GMC
- DVG$GMD
- DVG$QIS
- DVGCALL

List of mapping macros:
- DVG$UPL
- DVGAPL
- DVGAPX
- DVGQAR
- DVGQSR
- DVGSMF
- DVGSMPL
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, NOTAFTER=('21:34','94/12/25') is a control statement that assigns the value ('21:34','94/12/25') to the parameter represented by the keyword 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.

**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/VS.** 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.

**initialization file.** A file that contains parameters that determine how NetView FTP for Workstations starts running.

**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 AIX.  NetView File Transfer Program Server for AIX and NetView File Transfer Program Client for AIX.

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.

**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 SSCP s, 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 internetwork 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.

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, GH12-5480
- 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**

**VTAM Publications**
- VTAM Installation and Resource Definition Version 3, SC23-0111
- ACF/VTAM Version 3 Programming, SC23-0115
- ACF/VTAM Version 3 Customization, SC23-0112

**MVS Publications**
- MVS/Extended Architecture System Programming Library: System Macros and Facilities, Volume 1, GC28-1150
- MVS/Extended Architecture System Programming Library: System Macros and Facilities, Volume 2, GC28-1151
- MVS/Extended Architecture Supervisor Services and Macro Instructions, GC28-1154

---

3 NetView FTP/400 V1 only.
OS/VS Publications

OS/VS SPL: Supervisor, GC28-1046
OS/VS1 and OS/VS2 MVS Programmed

RACF Publications

System Programming Library: Resource Access Control Facility (RACF), SC28-1343

ISPF Publications

ISPF and ISPF/PDF Version 2 for MVS Diagnosis, SC34-4020
ISPF Version 2 for MVS Dialog Management Services, SC34-4021

OSI Publications

OSI/File Services General Information Manual, GH19-6636
OSI/File Services System/370 Programming Guide, SH19-6640
MVS and VM OSI/Communications Subsystem Configuration and Administration Guide, SL23-0168
OSI/File Services Installation and Customization Guide MVS, SH19-6639
## Index

### Numerics

16MB boundary 6

### A

**ABEND**
- codes 164
- dumps 165
- type-of-failure keyword 168

**abnormal termination**
- handling 153
- queue handler 155
- recovery 192
- server 157
- user interface routines 155

**ACBNAME** 80

**ACF/VTAM**
- application program node, defining server as 79
- conversation 8
- MODETAB keyword 82
- resources, setting up 22
- resources, setting up and maintaining 79
- terminating a conversation 10

**activating the trace facility** 137

**ACTIVE status** 9

**ADD command** 7

**ADD, SMF record** 224

**ADMIN, queue handler initialization parameter** 101

**administration, master password** 101

**AMS** 67, 69, 73

**APAR**
- description 163
- submitting 173

**APF authorized library** 19, 26

**APPC parameter** 80

**APPL**
- definition 79, 81
- definition statement 79
- name 80

**application program interface routine** 164, 202
- storage requirements
  - automatic areas 202
  - dynamic areas 202
  - modules 202

**area-of-failure** 169

**area-of-failure keywords** 166

**authorized library** 19, 26

**authorized program analysis report (APAR)** 163

**automated operations** 249

**automatic** (continued)
- OSI transfers 91
- rebuild of request queue 71
- transfer restart 189, 191
- transfer restart checkpointing 190

**automatic areas**
- storage requirements for application program interface routine 202
- storage requirements for batch job interface routine 202
- storage requirements for interactive interface routine 202
- storage requirements for queue handler 201
- storage requirements for server 202
- storage requirements for supervisor call routine 203

**AUTOSES parameter** 80

**AVLLRH server initialization parameter, effect on performance** 210

**AVLRRH, OSI server initialization parameter** 127

**AVLRRH, server initialization parameter** 120

**AVLRRH server initialization parameter, effect on performance** 210

**AVLRRH, OSI server initialization parameter** 127

**AVLRRH, server initialization parameter** 120

**AVLSESS server initialization parameter, effect on performance** 210

**AVLSESS, OSI server initialization parameter** 126

**AVLSESS, server initialization parameter** 120

**avoiding checkpoint data set overflow** 78

### B

**batch job**
- create and add a request 30, 33
- interface routine 164
- interface routine, storage requirements 202
- modify a request 35
- query a request 31, 34
- query administrator function 37
- query requests 37

**boundary, 16MB** 6

**byte checkpointing interval** 117

**BYTECR server initialization parameter, effect on performance** 210

**BYTECR, server initialization parameter** 117

### C

**CANCEL command** 61

**cataloged procedure**
- starting a server 47
- starting the queue handler 45
CDRM 81
CDRSC definition 81
changing message text 129
checking data sets 179
checkpoint
  avoiding data set overflow 78
data set 185
data set full 76
data set printing 78
facility 185
frequency 117
initial values 73
interval, byte 117
interval, time 117
overflow of data set 76
record 185, 186
serialized access to data set 76
checkpoint data set
  defining 73
deleting 73
initializing 73
rebuild 78
setting up 73
specifying parameters 199
checkpointing 10
  after abnormal termination 192
  automatic transfer restart 190
  exceptional 187
  manual transfer restart 192
  regular 187
checksum, file transfer 179
class
  file-transfer report spool 119
  keyword 115
  OSI server initialization parameter 128
  request 115
  server 115
  server initialization parameter 115
closing a file 10
cms users, messages to 10
codes 164
coeexistence with other installations of NetView FTP
MVS 195
commands
  MODIFY 45, 48
  OBTAIN, queue handler 7
  operator 45
  START 45, 47
  STOP 45
  VERIFY 153
common storage, MVS 6
compatibility
  Program Temporary Fix (PTF) 195
  with NetView FTP VM 195
  with NetView FTP VSE 195
  with NetView FTP/2 195
compatibility (continued)
  with NetView FTP/400 195
component identification 166
compression
data 9
  effect on performance 211
constants and fixed parameters 217
conversation
  establishing 8
COPY function, OSI/File Services 11
create
  and add a request, batch job 30, 33
  logon mode table 82
  server group table 85
CRRDATA parameter 76
CSA storage 203
CSAELIM queue handler initialization parameter, effect
  on performance 209
CSAELIM, queue handler initialization parameter 102
D
data compression and decompression 9
data integrity 179
DATA parameter 69
data security 179, 180
data set
  checking 179
  checkpoint 107, 185
  checkpoint printing 78
  checkpoint, full 76
  file-transfer report file 108
  initial values for the checkpoint data set 73
  initial values for the request queue 69
  queue handler initialization 98
  queue handler messages 97
  recovery 110
  request queue 97
  request queue printing 72
  server group table 98
  server messages 109
  server parameters 109
  SNAP dump 97, 108
  system dump 96, 109
database
  information systems 166
  known problems 163
DBCS 132
decompression, data 9
default first qualifier 118
defining
  a bigger request queue 71
  load libraries 19
  messages in another language 106
  server as ACF/VTAM application program node 79
  server as cross-domain resource 81
defining (continued)
  the checkpoint data set 73
  the request queue 67
degradation, performance 169
delay time, retry 118
DELAY, queue handler initialization parameter 100
deleting
  entries from the OSI/FS results file 93
  file-transfer requests 71
DEQ parameter 76
description of the symptom record 174
DLOGMOD parameter 80
DLOGMOD, server initialization parameter 117
documentation error 169
double-byte characters, messages requiring 132
DSESSLIM parameter 80
dump
  ABEND 165
  general 165
  SNAP 97, 165
DVG230OH job 28
DVGCCMDx 96
DVGCDRSC 22
DVGCRQX 20
DVGCPPARM 23
DVCPPROC 23
DVGCR 107
DVGCRQY 26
DVGCRREP 23
DVGCSVC 21
DVGEXEC 107
DVGJSMPE 22
DVGLOG 97, 109
DVGMCQI 20, 21
DVGMCXRD 73
DVGMSJSMPE 21
DVGMODTB 22
DVGMOQFTP 156
DVGMOFUN 157
DVGMSAM 158
DVGMSAX 159
DVGMVAD1 30
DVGMVAD2 33
DVGMVADM 37
DVGMBBFV 29
DVGMVMY2 35
DVGMVCR1 31
DVGMVCR2 34
DVGMXAPF 19
DVGMXCR1 23, 73
DVGMXCR2 77
DVGMXLPB 21
DVGMXR1I 25
DVGMXRQ1 23, 67
DVGMXSV 21
DVGNMT 98
DVGP2CRD 78
DVGP2QEU 72
DVQHFTP 96
DVQREP 108
DVQRF 110
DVQSNAP 97
DVQSVAPP 22
DVQSVFTP 106, 107
DVQSVG 23
DVQSVPSI 26
DVQVTRQ 97
dynamic
  allocation 9
  CSA storage requirements 203
  data security 180
  storage requirements 203
dynamic areas, storage requirements for
  application program interface routine 202
  batch job interface routine 202
  interactive interface routine 202
  queue handler 201
  server 202
  supervisor call routine 203
E
EHIINI function, OSI/File Services 11
ENQ parameter 76
entry and exit trace 136
entry traces 136
environment identification keywords 166
environment identification keywords, identifying 167
establish
  a conversation 8
  a security environment 9
ESTAE recovery routines 153, 165
evaluation, initial 163
events, issued by the
  queue handler and the server 249
  queue handler only 249
  server only 249
example
  symptom record 147
  trace dump 147
exceptional checkpointing 187
EXEC interface routine 164
exit traces 136
F
failure, area of 169
file
  avoiding checkpoint overflow 78
  closing a 10
  dynamic allocation 9
file (continued)
   handler 9
   opening 9
   protection 9
file-transfer
   checksum 179
   report 10
   report spool class 119
file-transfer request
   ACTIVE 9
   classes 115
   FINISHED 10
   introduction 4
   obtaining a new 10
   parameter checking 179
   WAITING 7
FINISHED status 10
first qualifier, default 118
fixed parameters and constants 217
FMID 167
formula to avoid request queue overflow 70
frequency
   checkpointing 117
   of queue handler enter/exit trace 149
   of server enter/exit trace 150
FSB storage requirements 203
FSBSLOTS, queue handler initialization parameter 101
FSBSLOTS, queue handler initialization parameter, effect on performance 209

G
   gaining access to the interactive interface routine of the Japanese Language Feature 27
general register save area dump, example 147
generated jobs, during installation 16
GET_RESULT command, OSI/File Services 12
gid, server initialization parameter 118
glossary 265
group, trace 137

H
   handling problems in NetView FTP V2.2.1 MVS 163

I
   IBM Support Center 163
   ICCF users, messages to 10
   id of request queue 185
   identify
      component identification keywords 167
      operating system 167
   IEBDG 67, 73
incorrect output 169
INCORROUT, type-of-failure keyword 168
INDEX parameter 69
Info-Sys 166
information systems 166
initial evaluation 163, 164
initial values
   checkpoint data set 73
   request queue 69
initialization parameters
   queue handler 99
   sample for a server 124
   sample for the queue handler 104
   server 113
installation verification
   overview 29
   preparing for 28
installing
   creating the checkpoint data set 23
   creating the request queue 23
   defining load libraries 19
   log file 23
   NetView FTP V2.2.1 MVS 15
   OSI part 15
   performing IPL 22
   RACF authorization 22
   report file 23
   setting up ACF/VTAM resources 22
   SVC routine 21
   unloading the distribution tape 15
   unloading the distribution tape for NLS 15
   verifying the installation 28
integrity, data 179
interactive interface
   special considerations 165
interactive interface routine
   gaining access to 24
   preparing access 24
   storage requirements
      automatic areas 202
      dynamic areas 202
      modules 202
   verifying access 39
interactive interface routine, Japanese Language Feature 27
Interactive System Productivity Facility (ISPF) 165
interface routine 6
internal trace 123, 128, 166
interpreting SMF records 224
interval
   obtain 118
time checkpoint 117
IPL, after loading 22
ISPF library definition 25
Japanese Language Feature
interactive interface routine 27
preparing 26
preparing the NetView FTP V2.2.1 MVS jobs 26

job
DVG230nn 29
DVG230OH 28
DVGC3VC 21
DVGJSMPE 22
DVGMJSMPE 21
JOBLIB 96, 107

keyword
ADMIN master password 101
ADMIN, queue handler initialization parameter 101
AVLLRH, OSI server initialization parameter 127
AVLRRH, server initialization parameter 120
AVLRRRH, OSI server initialization parameter 127
AVLRRH, server initialization parameter 120
AVLSESS, OSI server initialization parameter 126
AVLSESS, server initialization parameter 120
BYTECR, server initialization parameter 117
CLASS, OSI server initialization parameter 128
CLASS, server initialization parameter 115
CSAELIM, queue handler initialization parameter 102
DELAY, queue handler initialization parameter 100
DLOGMOD, server initialization parameter 117
FSBSLOTS, queue handler initialization parameter 101
GID, server initialization parameter 118
LUNAME, server initialization parameter 115
MAXSESS, OSI server initialization parameter 126
MAXSESS, server initialization parameter 120
MAXSRV, queue handler initialization parameter 100
MODETAB, ACF/VTAM 82
NFTPNAME, queue handler initialization parameter 99
NVEVENT, queue handler initialization parameter 102
NVEVENT, server initialization parameter 122
OBTINTVL, OSI server initialization parameter 126
OBTINTVL, server initialization parameter 118
OIAAPLID, OSI server initialization parameter 125
OITAMID, OSI server initialization parameter 125
OUNISRPW, OSI server initialization parameter 126
OIRECDEL, OSI server initialization parameter 127
OISVNAME, OSI server initialization parameter 125
OUSERID, OSI server initialization parameter 125
OUSRNPW, OSI server initialization parameter 126

POSTCONV, server initialization parameter 119
POSTTRAN, server initialization parameter 119
PPEEXIT, server initialization parameter 119
PRETRAN, server initialization parameter 119
REPCCLASS, server initialization parameter 119
RETRYDELAY, OSI server initialization parameter 127
RETRYDELAY, server initialization parameter 118
SECPAR, server initialization parameter 121
SMFREC, queue handler initialization parameter 103, 123
SRVNAME, OSI server running mode 125
SRVNAME, server initialization parameter 116
SRVPREF 105
SRVPREF, queue handler initialization parameter 100
STARTSV, queue handler initialization parameter 101
STARTSVx 47
SUPSTOW, server initialization parameter 123
TIMECR, server initialization parameter 117
TRACE, OSI server initialization parameter 128
TRACE, queue handler initialization parameter 103
TRACE, server initialization parameter 123
TTIME, OSI server initialization parameter 128
TTIME, server initialization parameter 116

keyword (continued)

known problems, searching for 163

language message model 96
language support, queue handler 96
library
APF authorized 19, 26
link list, MVS 20
MVS link list 20
limit to the maximum number of simultaneous requests 126
link-list library, MVS 20
linklib 96
list of macros 261
load
checkpoint data set 73
request queue 69
load module, server 106
Local Resource Directory, OSI/File Services 11
local user password 126
log file, queue handler 97
logical unit name, server 115
logical unit password, server 115
logon mode table 82
logon retry, automatic 193
LOOP, type-of-failure keyword 168

Index 285
LPA library 96, 107
LRD user name of OSI server 125
LU 6.2 parameters 83
LUNAME keyword 115
LUNAME server initialization parameter 115

M
macro list 261
macroinstruction
  OPNDST 80
  RACROUTE 180, 181
  SETRP 156
  SNAP 135, 157, 158
main task recovery
  queue handler 156
  server 158
maintaining
  ACF/VTAM resources 79
  server group table 85
manual
  OSI transfers 91
  transfer restart 191
master password, ADMIN 101
maximum number of active transfer requests 126
maximum number of local request handler sessions 127
MAXSESS 126
  OSI server initialization parameter 126
  server initialization parameter 120
  server initialization parameter, effect on performance 210
MAXSRV queue handler initialization parameter, effect on performance 210
MAXSRV, queue handler initialization parameter 100
member
  DVGCSVC 21
  DVGMCXR1 73
  DVGMCXR2 77
  DVGMRQ1 67
  DVGPRQU 72
message
  changing text 129
  error 164
  language definition 106
  multiline WTO 249
  user 10
migration
  from NetView FTP V1 MVS 195
  from NetView FTP V2.1 MVS 197
  from NetView FTP V2.2.0 MVS 198
mode name, server 117
mode, server running 116
MODETAB keyword, ACF/VTAM 82
MODETAB parameter 80
modification level 167
MODIFY command 45, 48
modifying a NetView FTP V2.2.1 MVS file-transfer request 35
modifying a request 35
module
  groups 138, 139
  module traces 136
modules
  DVGCCMDx 96
  DVGCMCIQ 20
  DVGGMQFTP 156
  DVGMQFUN 157
  DVGMSAM 158
  DVGMSAX 159
  DVGQHFTP 96
  DVGSVFFTP 107
  storage requirements for application program interface routine 202
  storage requirements for batch job interface routine 202
  storage requirements for interactive interface routine 202
  storage requirements for queue handler 201
  storage requirements for server 202
  storage requirements for supervisor call routine 203
MOVE function, OSI/File Services 11
multiline WTO 251
multiline WTO messages 249
MVS common storage 6
MVS link-list library 20

N
NetView FTP V1 MVS, migration from 195
NetView FTP V2.1 MVS, migration from 197
NetView FTP V2.2.0 MVS, migration from 198
NetView FTP V2.2.1 MVS
  messages 164
  startup procedures 95
NetView FTP V2.2.1 MVS Japanese Language Feature
  verifying installation 40
NetView FTP V2.2.1 MVS OSI part
  verifying installation 39
NetView FTP VM compatibility 195
NetView FTP VSE compatibility 195
NetView FTP/2 compatibility 195
NetView FTP/400 compatibility 195
NetView, using to operate NetView FTP V2.2.1 MVS 249
network ID 185
new local user password 126
NFTPNAME, queue handler initialization parameter 99
non-native network connection, VTAM 83
notations, operator commands 43
NOTIFY, SMF record 224
number of file transfers, effect on performance 210
NVEVENT, queue handler initialization parameter 102
NVEVENT, server initialization parameter 122

O

OBTAIN
queue handler command 7
SMF record 224
obtain interval 118, 126
obtaining a new file-transfer request 10
OBTINTVL server initialization parameter, effect on performance 212
OBTINTVL, OSI server initialization parameter 126
OBTINTVL, server initialization parameter 118
OIAPPLID, OSI server initialization parameter 125
OIFTMID, OSI server initialization parameter 125
OINUSRPW, OSI server initialization parameter 126
OIRECDEL, OSI server initialization parameter 127
OISVNAME, OSI server initialization parameter 125
OIUSERID, OSI server initialization parameter 125
OUSRPWD, OSI server initialization parameter 126
Open Systems Interconnection 3, 91
opening a file 9
operating NetView FTP V2.2.1 MVS 43, 249
operating system, identifying 167
operations, automated 249
operator commands 45
OPNDST macroinstruction 80
OSI 3
log file entries 161
network 4
security 183
server 4, 6
server name 125
server recovery 160
OSI file transfers 94
administering 91
automatic 92
manual 92
recovery 94
recovery file 191
OSI part 15
OSI server 11
preparing 26
OSI/File Services
COPY function 11
EHINI function 11
GET_RESULT command 12
Local Resource Directory 11
MOVE function 11
overflow of request queue 71

P

pacing specifications 213
parameter
ADMIN, queue handler initialization 101
APPC 80
AUTOSES 80
AVLLRH, OSI server initialization 127
AVLLRH, server initialization 120
AVLRRH, OSI server initialization 127
AVLRRH, server initialization 120
AVLSESS, OSI server initialization 126
AVLSESS, server initialization 120
BYTECR, server initialization 117
CLASS, OSI server initialization 128
CLASS, server initialization 115
CRRDATA 76
CSAILIM, queue handler initialization 102
DATA 69
DELAY, queue handler initialization 100
DEQ 76
DLOGMOD 80
DLOGMOD, server initialization 117
DSESLSLIM 80
ENQ 76
FBSLOPTS, queue handler initialization 101
GID, server initialization 118
INDEX 69
LUNAME, server initialization 115
MAXSESS, OSI server initialization 126
MAXSESS, server initialization 120
MAXSRV, queue handler initialization 100
MODETAB 80
NFTPNAME, queue handler initialization 99
NVEVENT, queue handler initialization 102
NVEVENT, server initialization 122
OBTINTVL, server initialization 118, 126
OIAPPLID, OSI server initialization 125
OIFTMID, OSI server initialization 125
OINUSRPW, OSI server initialization 126
OIRECDEL, OSI server initialization 127
OISVNAME, OSI server initialization 125
OIUSERID, OSI server initialization 125
OUSRpwd, OSI server initialization 126
PARM 96
PARSESS 80
POSTCONV, server initialization 119
POSTTRAN, server initialization 119
PPEXIT, server initialization 119
PRETRAN, server initialization 119
QUANTITY 69
RECORD 69, 71
REGION 96
REPCCLASS, server initialization 119
RETRYDELAY, OSI server initialization 127
RETRYDELAY, server initialization 118
parameter (continued)
SECACPT  81
SECPAR, server initialization 121
server class 115
server initialization 113
server LU name 115
server mode name 117
server termination time 116
SHAREOPTIONS 76
SMFREC, queue handler initialization 103, 123
SRVMODE, OSI server running mode 125
SRVMODE, server initialization 116
SRVPREF, queue handler initialization 100
STARTSV, queue handler initialization 101
SUPSTOW, server initialization 123
SYSIN 69, 75
TIMECR, server initialization 117
TRACE, OSI server initialization 128
TRACE, queue handler initialization 103
TRACE, server initialization 123
TTIME, OSI server initialization 128
TTIME, server initialization 116
UNIT 69
VOLUMES 69
VPACING 80
PARM parameter 96
PARSESS parameter 80
password
  master 101
  protection, VSAM 183
  VSAM 183
patch-area traces 136
PERFM, type-of-failure keyword 168
performance
degradation 169
  factors affecting 209
post-conversation user-exit routine 119
post-transfer user-exit routine 10, 119
POSTCONV, server initialization parameter 119
POSTTRAN, server initialization parameter 119
PPEXIT, server initialization parameter 119
pre-queuing user-exit routine 7
pre-transfer user-exit routine 9, 119, 183
prefix, server 100
preparing
  interactive interface prepare 24
  the installation of NetView FTP V2.2.1 MVS 15
  preparing for the installation verification test 28
preparing the Japanese Language Feature of NetView FTP V2.2.1 MVS 26
preparing the NetView FTP V2.2.1 MVS jobs for the Japanese Language Feature 26
preparing the NetView FTP V2.2.1 MVS OSI server 26
PRETRAN, server initialization parameter 119
printing checkpoint data set 78
printing request queue 72
problem
description 166
evaluation 163, 164
reporting 163
  verifying the description 163
product
  modification level 167
  release 167
  version 167
program logic error ABEND codes 164
PSNDPAC 213
PTF (Program Temporary Fix) 195
put level 167
Q
qualifier, default first 118
QUANTITY parameter 69
query requests, batch job 31, 34, 37
queue handler 6, 164
  abnormal termination 155
  events issued by 249
  frequency of enter/exit 149
  initialization parameters 99
ADMIN 101
CSAELIM 102
CSAELIM, effect on performance 209
DELAY 100
FSBSLOTS 101
FSBSLOTS, effect on performance 209
MAXSRV 100
MAXSRV, effect on performance 210
NFTPNAME 99
NVEVENT 102
sample of 104
SMFREC 103, 123
SRVPREF 100
STARTSV 101
TRACE 103
language support 96
log file 97
main task recovery 156
modify 45
processing constants 217
service subtask recovery 157
setting up 99
starting 28, 45
startup procedure 45
stop 45
storage requirements
  automatic areas 201
dynamic areas 201
  modules 201
tracing 139
R
RACF (see SAF) 180
RACF authorization of started tasks 22
RACROUTE macroinstruction 180, 181
reason codes 164
rebuild request queue
    automatic 71
receiving server, determining 9
record format, server group table 85
RECORD parameter 69, 71
record size, specifying for request queue 199
recovery
    delay time 127
    ESTAE routine 153
    OSI server 160
REGION parameter 96
register dump as output from tracing 147
regular checkpointing 187
release 167
release a request 35
release level 167
remote server 8
reorganizing the checkpoint data set 77
REPCLASS, server initialization parameter 119
reporting problems to IBM 163
REQDELAY 197
request
    classes 115
request queue
    automatic rebuild 71
    avoiding overflows 71
    defining 67
    defining, with greater number of records 71
    formula 70
    initial values 69
    initializing 69
    overflow 71
    printing 72
    redefining 71
    setting up 67
    specifying parameters 199
    specifying record size 199
request unit
    size, effect on performance 212
retry delay time 118, 127
RETRYDELAY 197
RETRYDELAY, OSI server initialization parameter 127
RETRYDELAY, server initialization parameter 118
return codes 164
routines
    ESTAE recovery 153
    interface 6
    post-transfer user-exit 10
routines (continued)
    pre-queuing user exit 7
    pre-transfer user-exit 9
    SVC 153, 157, 203
rules
    changing message text 130
running mode
    OSI server 125
    server 116
S
SAF (System Authorization Facility) 180
SAF, server authorization 180
sample
    DVGCDRSC 22
    DVGCPPARM 23
    DVGCPPROC 23
    DVGCRRREP 23
    DVGMVAD1 30
    DVGMVAD2 33
    DVGMVADM 37
    DVGMVBFV 29
    DVGMVMY2 35
    DVGMVQOR1 31
    DVGMVQOR2 34
    DVGMXAPF 19
    DVGMXCR1 23
    DVGMXLPA 21
    DVGMXRII 25
    DVGMXRQ1 23
    DVGMXSVC 21
    DVGPCRD 78
    DVGSVGRP 23
    modifying a request 35
    OSI server startup job 111
    OSI server startup procedure 109
    SNA server startup job 110, 112
    SNA server startup procedure 105
sample initialization parameters
    for a server 124
    queue handler 104
scheduling
    restart from checkpoint 193
search
    known problems 163
    similar problems 186
    strategy 172
SECACPT parameter 81
SECPAR 196
SECPAR, server initialization parameter 121
security
    checking 183
    establishing 9
    security, data 179
sending server, determining 9
serialized access to checkpoint data set 76
server 85, 164
abnormal termination 157
authorization, SAF 180
automatic start 47
batch job start 48
class 115
CLASS initialization parameter 115
defining as application program node 79
defining as cross-domain resource 81
determining sending and receiving 9
DLOGMOD, initialization parameter 117
events issued by 249
frequency of enter/exit 150
initialization parameters 113
   AVLLRH 120
   AVLLRH, effect on performance 210
   AVLRRH 120
   AVLRRH, effect on performance 210
   AVLSESS 120
   AVLSESS, effect on performance 210
   BYTECR 117
   BYTECR, effect on performance 210
   CLASS 115
   DLOGMOD 117
   GID 118
   LUNAME 115
   MAXSESS 120
   MAXSESS, effect on performance 210
   NEVENT 122
   OB TINTVL, effect on performance 212
   OBTINTVL 118
   POSTCONV 119
   POSTTRAN 119
   PPEXIT 119
   PRETRAN 119
   REPCLASS 119
   RETRYDELAY 118
   sample of 124
   SECPAR 121
   SRVMODE 116
   SUPSTOW 123
   TIMECR 117
   TIMECR, effect on performance 210
   TRACE 123
   TTIME 116
   xSECU RP 121
initialization parameters (OSI)
   AVLLRH 127
   AVLRRH 127
   AVLSESS 126
   CLASS 128
   MAXSESS 126
   OB TINTVL 126
   OIAPIPLID 125
   OIFTAMID 125
   SETRP macroinstruction 156
server (continued)
   initialization parameters (OSI) (continued)
   OINUSRPW 126
   OIRECDEL 127
   OISVNAME 125
   OIUSERID 125
   OIUSRPWD 126
   RETRYDELAY 127
   TRACE 128
   TTIME 128
   load module 106
   logical unit name 115
   logical unit password 115
   LU name 8
   LUNAME initialization parameter 115
   main task recovery 158
   mode 116
   mode name 117
   remote 8
   running mode 116
   running mode (OSI)
      SRVMODE 125
   sample startup job
      OSI 111
      SNA 110, 112
   sample startup procedure
      OSI 109
      SNA 105
   starting 29, 45, 47
   starting automatically 47
   starting using MODIFY command 48
   stopping 61
   stopping using MODIFY command 61
   storage requirements
      automatic areas 202
      dynamic areas 202
      modules 202
   submitting a job to start 105
   subtask recovery 159
   termination time 116
   tracing 140
   TTIME initialization parameter 116
   writing a procedure to start 105
server class 128
server classes, effect on performance 211
server group table 85, 88
server group table, sample 87
server groups, special considerations 87
server initialization parameters 113
server name 125
server prefix 100
service subtask recovery, queue handler 157
setting up
   the checkpoint data set 73
   the request queue 67

290  NetView FTP V2 MVS Installation, Operation, and Administration
setting up ACF/VTAM resources 22
setting up and maintaining
   a server group table 85
      ACF/VTAM resources 79
setting up the queue handler 99
shared block storage requirements 203
SHAREOPTIONS parameter 76
similar problems, searching for 166
SMF recording 103, 123
SMF records
   ADD 224
   interpreting 224
   NOTIFY 224
   OBTAIN 224
SMFREC, queue handler initialization parameter 103, 123
SMP 167
SNA 3
SNA network 4
SNA server 4, 6, 8
SNAP dump 153, 165
SNAP macroinstruction 135, 157, 158
special considerations for server groups 87
specific module trace 136
specifying checkpoint data-set parameters 199
specifying request queue parameters 199
spool class, file-transfer report 119
SRCVPAC 213
SRVMODE, OSI server running mode 125
SRVMODE, server initialization parameter 116
SRVPREF keyword 105
SRVPREF, queue handler initialization parameter 100
SSNDPAC 213
START command 45, 47, 63
start options, VTAM 83
started tasks, RACF authorization 22
starting a server 47, 48
   automatically 47
   using a batch job 48
   using the MODIFY command 48
   using the START command 47
starting the queue handler 28
STARTSV, queue handler initialization parameter 101
STARTSVx keyword 47
startup procedures for NetView FTP V2.2.1 MVS 95
statement
   VBUILD 79
static storage requirements 203
status
   ACTIVE 9
   FINISHED 10
   WAITING 7
STEPLIB 96, 107
STOP command 45, 63
stopping a server 61

stopping NetView FTP V2.2.1 MVS 12
storage area dump 149
storage requirements
   calculating additional CSA storage 204
   for application program interface routine 202
   for batch job interface routine 202
   for interactive interface routine 202
   for queue handler 201
   for server 202
   for supervisor call routine 203
string development sheet 175
submitting an APAR 163, 173
subtask recovery, server 159
supervisor call routine storage requirements
   automatic areas 203
   dynamic areas 203
   modules 203
SUPSTOW, server initialization parameter 123
SVC routine 153, 157, 203
SVC routine, installing 21
symptom keywords
   area-of-failure 166
   component identification 166
   environment identification 166
   initial evaluation 163
   similar problems, searching for 166
   transfer-failure 166
   type-of-failure 166
symptom record
   description 174
   example 147
   general 165
   initial evaluation 163
   output from tracing 145
symptom series
   initial evaluation 163
SYNAD 251
SYSIN 98, 99, 109
SYSIN parameters 69, 75
System Authorization Facility (SAF) 180
system modification program (SMP/E) 167
Systems Network Architecture 3
SYSUDUMP 96, 109

T

table entry, logon mode 82
taking checkpoints 10
terminating an ACF/VTAM conversation 10
termination time 128
termination time, server 116
termination, abnormal 153
test data set
   DVGMVBVFV 29
text, changing message 129
time checkpoint interval 117
TIMECR server initialization parameter, effect on performance 210
TIMECR, server initialization parameter 117
trace
activating 137
data example 147
dump example 147
general 135
group 137
internal 123, 166
points 135
types 136
when to take 135
TRACE, OSI server initialization parameter 128
TRACE, queue handler initialization parameter 103
TRACE, server initialization parameter 123
tracing 135
enter/exit queue handler frequency 149
enter/exit server frequency 150
entry and exit 136
output 145
output produced 147, 149
output to register dump 147
output to symptom record 145
queue handler 139
queue handler entry and exit trace 136
server 140
specific module 136
specific server modules 140
transfer error recovery 185
transfer restart
automatic 189, 191
manual 191
transfer-failure keywords 166
transfer-failure, describing 171
TRC 196, 197
TSO logon procedure 24
TSO startup CLIST 24
TTIME, OSI server initialization parameter 128
TTIME, server initialization parameter 116
type-of-failure keywords 166
type-of-failure keywords, identifying 168
types of traces 136

user-exit routines (continued)
pre-transfer 9, 119
user-written exit routines
post-conversation 119
post-transfer 10
pre-queuing 7
pre-transfer 9
request queue 7
utility IEBDG 67, 73

V
values
for the checkpoint data set, initial 73
for the request queue, initial 69
VBUILD statement 79
VERIFY command 153
verifying
installation of OSI part 39
installation of the Japanese Language Feature 40
interactive interface access 39
the installation 28
the installation of NetView FTP V2.2.1 MVS 15
verifying the problem description 163
version 167
VOLUMES parameter 69
VPACING parameter 80
VTAM LU name of OSI/Files Services 125
VTAM LU name of the server 125
VTAM non-native network connection 83
VTAM start options 83

W
wait condition 168
WAIT, type-of-failure keyword 168
WAITING status 7
when to take traces 135
writing a procedure to start a server 105
WTO, multiline 251
WTO, multiline messages 249

X
xSECURP 121
Your comments, please ...

NetView
File Transfer Program
Version 2 for MVS
Installation, Operation, and Administration
Release 2.1
Publication No. SH12-5657-04

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 IBMMAIL.

Please include the title and publication number (as shown above) in your reply.