Third Edition (September 2002)

This edition applies to Version 1 Release 4 of z/OS (5694-A01) and Version 1 Release 4 of z/OS.e (5655-G52) and to all subsequent releases and modifications until otherwise indicated in new editions.

Publications are not stocked at the address given below. If you want more IBM® publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Software Reengineering
Department G71A/ Bldg 503
Research Triangle Park, NC 27709-9990
U.S.A.

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):
1-800-254-0206

Internet e-mail:
usib2hpd@vnet.ibm.com

World Wide Web:

IBMLink™:
CIBMORCF at RALVM17

IBM Mail Exchange:
tkinlaw@us.ibm.com

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 1990, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
## Contents

<table>
<thead>
<tr>
<th>Figures</th>
<th>xi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>xiii</td>
</tr>
</tbody>
</table>

### About this document

- Who should use this document: xv
- Where to find more information: xv
- Where to find related information on the Internet: xvi
- Accessing z/OS licensed documents on the Internet: xvi
- Using LookAt to look up message explanations: xvii
- How to contact IBM service: xvii
- z/OS Communications Server information: xvii

### Summary of changes
xxv

### Chapter 1. Getting started.

1. Understanding TCP/IP
   - Physical network: 1
   - Protocols: 2
   - Network devices: 2
   - Addresses: 2
   - How TCP/IP uses networks: 2
   - Local and remote nodes: 2
   - Client and server: 2
   - TCP/IP addresses: 3
   - Network names: 3
   - Ports and port numbers: 3

2. Understanding what you can do
   - Logging on to other hosts: 4
   - Transferring data sets between hosts: 4
   - Sending and receiving mail: 4
   - Remote command execution: 5
   - Printing to or from other hosts: 5

3. What you need to get started
   - Testing commands with loopback: 6
   - Obtaining command help: 6

### Chapter 2. Logging on to a host using TELNET

9.

- Using the TSO TELNET command: 9
- TELNET command: 10

13.

- Using the TELNET subcommands:
  - AO—Terminate output display: 13
  - AYT—Query the connection: 14
  - BRK—Send a Break or Attention keystroke to a host: 15
  - HELP or ?—Display help information: 16
  - IP—Interrupt the process: 17
  - PA1—Send the PA1 keystroke to a host: 18
  - QUIT—End the TELNET session: 19
  - SYNCH—Clear the data path: 20
  - c and ‘—Send ASCII control characters to a host in line mode: 21

- Using the TELNET function keys: 22
  - Transparent mode function keys: 23
  - Line mode function keys: 23
FTP client error codes ........................................... 100
Restarting a failed data transfer .......................... 101
Interfacing with JES ............................................. 101
  Steps for submitting a job .................................. 102
  Displaying the status of a job .............................. 103
  Receiving spool output .................................... 104
  Deleting a job .................................................. 106
  Steps for submitting a job and automatically receiving output 106
  Terminating access to JES ................................. 107
  JESINTERFACELEVEL differences .......................... 107
  JES security .................................................. 109
  JES examples ................................................. 111
Performing DB2 SQL queries with FTP ..................... 117
  SQL data types supported by FTP ......................... 117
  Creating the input data set ................................ 118
  Setting the characteristics for the SQL query .......... 118
  Submitting the query ....................................... 121
  Examples of SQL query output ............................ 122
FTP with DBCS support ......................................... 123
  Selecting a DBCS translation table ...................... 123
  Selecting an SBCS translation table ..................... 123
  DBCS subcommands .......................................... 124
  Server commands and client subcommands ................ 124
  Mapping DBCS aliases to CCSIDs .......................... 125

Chapter 5. FTP subcommands ................................. 127
  ! subcommand—Invoke a z/OS UNIX System Services function 130
  ACCOUNT subcommand—Supply account information .... 131
  APPend subcommand—Append a local data set ........... 132
  AScii subcommand—Change the data transfer type to ASCII 134
  BIG5 subcommand—Change the data transfer type to BIG5 .. 135
  BINary subcommand—Change the data transfer type to Image 137
  BLOCK subcommand—Set the block data transfer mode .... 138
  CCc subcommand—Turn off integrity protection .......... 139
  CD subcommand—Change the directory on the remote host 140
  CDUp subcommand—Change to the parent of the working directory 143
  CLEar subcommand—Set the protection level for data transfers to CLEAR 145
  CLOSE subcommand—Disconnect from a remote host ....... 146
  COMPress subcommand—Set the compressed data transfer mode 147
  CProtect subcommand—Set the protection level on commands 148
  DEBug subcommand—Set general trace options ........... 149
  DELEte subcommand—Delete files .......................... 153
  DELimit subcommand—Display the file name delimiter ... 154
  Dlr subcommand—Obtain a list of directory entries ...... 155
  DUMP subcommand—Set extended trace options .......... 159
  EBcdic subcommand—Change the data transfer type to EBCDIC 161
  EUckanji subcommand—Change the data transfer type to EUCKANJI 162
  FFeature subcommand—Query FTP server for features it supports 164
  FFile subcommand—Set the file structure to File ......... 165
  Get subcommand—Copy files ................................ 166
  GLob subcommand—Toggle expansion of metacharacters .... 168
  HAngeul subcommand—Change the data transfer type to HANGEUL 170
  HELP and ? subcommands—Display help information ...... 172
  Ibmkkanji subcommand—Change the data transfer type to IBMKANJI 173
  JIS78kj subcommand—Change the data transfer type to JIS78KJ 174
  JIS83kj subcommand—Change the data transfer type to JIS83KJ 175
Chapter 6. Sending electronic mail using SMTP commands

Interfaces to the SMTP address space

Using the SMTPNOTE command from your terminal

SMTPNOTE command—Send electronic mail to one or more recipients on NJE or TCP networks

Preparing and sending mail
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMCPnn option</td>
<td>Override GDDM multicolor patterns with workstation color names</td>
<td>356</td>
</tr>
<tr>
<td>HostRast option</td>
<td>Perform raster image processing at the System/370 Host</td>
<td>357</td>
</tr>
<tr>
<td>NewLine option</td>
<td>Override the default key mapping for NewLine</td>
<td>358</td>
</tr>
<tr>
<td>XSync option</td>
<td>Request that the X Window system process one request at a time</td>
<td>359</td>
</tr>
<tr>
<td>ZWL option</td>
<td>Tell GDDMXD/MVS to draw all lines using 0-width lines</td>
<td>360</td>
</tr>
<tr>
<td>Keyboard functions</td>
<td>GDDMXD/MVS keyboard functions</td>
<td>361</td>
</tr>
<tr>
<td></td>
<td>GDDMXD/MVS to X Window system keyboard functions</td>
<td>361</td>
</tr>
<tr>
<td></td>
<td>APL2 character set keyboard</td>
<td>362</td>
</tr>
<tr>
<td>Setting up</td>
<td>hlq.GDXAPLCS.MAP</td>
<td>362</td>
</tr>
<tr>
<td>Chapter 10.</td>
<td>Executing commands on a remote host</td>
<td>363</td>
</tr>
<tr>
<td>REXEC command</td>
<td>Execute a command on the remote host and receive the results on your local host</td>
<td>364</td>
</tr>
<tr>
<td>Using the NETRC data set</td>
<td></td>
<td>366</td>
</tr>
<tr>
<td>Submitting REXEC and RSH requests in batch</td>
<td></td>
<td>366</td>
</tr>
<tr>
<td>RSH command</td>
<td>Execute a command on a remote host and receive the results on your local host</td>
<td>370</td>
</tr>
<tr>
<td>RHOSTS.DATA data set</td>
<td></td>
<td>372</td>
</tr>
<tr>
<td>Using remote execution clients in a z/OS UNIX environment</td>
<td></td>
<td>372</td>
</tr>
<tr>
<td>The z/OS UNIX orexec/rexec command</td>
<td>Execute a command on the remote host</td>
<td>373</td>
</tr>
<tr>
<td>Chapter 11.</td>
<td>Using the network database system</td>
<td>375</td>
</tr>
<tr>
<td>NDBCLNT command</td>
<td>Issue SQL statements to a DB2 subsystem</td>
<td>376</td>
</tr>
<tr>
<td>SQL statements</td>
<td></td>
<td>378</td>
</tr>
<tr>
<td>NDB commands</td>
<td></td>
<td>378</td>
</tr>
<tr>
<td>Return codes for NDB</td>
<td></td>
<td>378</td>
</tr>
<tr>
<td>Format of output displayed on the client</td>
<td></td>
<td>381</td>
</tr>
<tr>
<td>Appendix A.</td>
<td>Specifying data sets and files</td>
<td>383</td>
</tr>
<tr>
<td>MVS data sets</td>
<td></td>
<td>383</td>
</tr>
<tr>
<td>Sequential data sets</td>
<td></td>
<td>384</td>
</tr>
<tr>
<td>Partitioned data sets</td>
<td></td>
<td>384</td>
</tr>
<tr>
<td>Transferring data between partitioned and sequential data sets.</td>
<td></td>
<td>385</td>
</tr>
<tr>
<td>Data transfer methods.</td>
<td></td>
<td>386</td>
</tr>
<tr>
<td>Transferring PDS directory information.</td>
<td></td>
<td>386</td>
</tr>
<tr>
<td>AIX and UNIX files</td>
<td></td>
<td>386</td>
</tr>
<tr>
<td>AS/400 operating system.</td>
<td></td>
<td>387</td>
</tr>
<tr>
<td>VM files</td>
<td></td>
<td>388</td>
</tr>
<tr>
<td>Appendix B.</td>
<td>Mapping values for the APL2 character set</td>
<td>389</td>
</tr>
<tr>
<td>Appendix C.</td>
<td>TELNET extensions</td>
<td>395</td>
</tr>
<tr>
<td>Character set cross reference table.</td>
<td></td>
<td>395</td>
</tr>
<tr>
<td>Special key operation for TELNET</td>
<td></td>
<td>397</td>
</tr>
<tr>
<td>Operation of PF and PA keys.</td>
<td></td>
<td>399</td>
</tr>
<tr>
<td>Sense codes for special key operation.</td>
<td></td>
<td>400</td>
</tr>
<tr>
<td>Appendix D.</td>
<td>ICMPv6 types and codes</td>
<td>403</td>
</tr>
<tr>
<td>Appendix E.</td>
<td>How to read a syntax diagram</td>
<td>405</td>
</tr>
<tr>
<td>Symbols and punctuation</td>
<td></td>
<td>405</td>
</tr>
<tr>
<td>Parameters.</td>
<td></td>
<td>405</td>
</tr>
</tbody>
</table>
Figures

1. Sample output of the MORE program—first screen ........................................... 24
2. Sample output of the MORE program—second screen ....................................... 24
3. JCL to run FTP in batch using data sets ......................................................... 93
4. Contents of an INPUT DD data set ................................................................. 93
5. JCL to run FTP in batch without using data sets .............................................. 94
6. Job to create a new GDS in batch .................................................................... 94
7. How to issue the FTP subcommands from a data set ....................................... 95
8. How to issue the FTP subcommands from an HFS file system ......................... 96
9. How FTP subcommands can be issued from an EXEC ....................................... 97
10. SMTP gateway overview ............................................................................... 286
11. Example of preparing and sending mail ......................................................... 289
12. TSO RECEIVE command ............................................................................... 290
13. Example of a nondelivery note ....................................................................... 291
14. Example of an unknown recipient note .......................................................... 291
15. SMTP as a mail gateway .................................................................................. 297
<table>
<thead>
<tr>
<th>Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. TELNET subcommands</td>
</tr>
<tr>
<td>2. ASCII control characters</td>
</tr>
<tr>
<td>3. TELNET function keys in line mode</td>
</tr>
<tr>
<td>4. FTP subcommands for getting help</td>
</tr>
<tr>
<td>5. FTP subcommands for establishing and exiting a connection</td>
</tr>
<tr>
<td>6. FTP subcommands for obtaining status and system information</td>
</tr>
<tr>
<td>7. FTP subcommands for working with directories on the remote host</td>
</tr>
<tr>
<td>8. FTP subcommands for working with directories on the local host</td>
</tr>
<tr>
<td>9. FTP subcommands for preparing the environment</td>
</tr>
<tr>
<td>10. FTP subcommands for transferring data</td>
</tr>
<tr>
<td>11. Recommended methods for data transfer</td>
</tr>
<tr>
<td>12. FTP client search orders</td>
</tr>
<tr>
<td>13. FTP.DATA data set statements for the FTP client</td>
</tr>
<tr>
<td>14. FTP subcommand codes</td>
</tr>
<tr>
<td>15. Internal error codes</td>
</tr>
<tr>
<td>16. FTP subcommands for DBCS support</td>
</tr>
<tr>
<td>17. FTP TYPE subcommand aliases</td>
</tr>
<tr>
<td>18. Mapping of DBCS keywords to CCSIDs</td>
</tr>
<tr>
<td>19. FTP subcommands</td>
</tr>
<tr>
<td>20. SMTP commands</td>
</tr>
<tr>
<td>21. Remote printing commands</td>
</tr>
<tr>
<td>22. Supported graphics presentation space sizes</td>
</tr>
<tr>
<td>23. GDDMXD/MVS options</td>
</tr>
<tr>
<td>24. GColors.</td>
</tr>
<tr>
<td>25. Recommended methods for data transfer</td>
</tr>
<tr>
<td>26. Mapping values for the APL2 character set.</td>
</tr>
<tr>
<td>27. TCP/IP character set cross reference</td>
</tr>
<tr>
<td>28. Special key conversions.</td>
</tr>
<tr>
<td>29. Sense codes</td>
</tr>
<tr>
<td>30. ICMP types and codes</td>
</tr>
<tr>
<td>31. ICMPv6 types and codes</td>
</tr>
<tr>
<td>32. IP information APARs.</td>
</tr>
<tr>
<td>33. SNA information APARs.</td>
</tr>
<tr>
<td>34. Non-document information APARs</td>
</tr>
</tbody>
</table>
About this document

This document describes how to use the applications available in z/OS™ Communications Server to perform the following functions:

- Log on to a remote host
- Transfer data sets
- Send and receive electronic mail
- Print on remote printers
- Display IBM GDDM®/MVS™ graphics on X Window System workstations
- Run a command on another host

The information in this document supports both IPv6 and IPv4. Unless explicitly noted, information describes IPv4 networking protocol. IPv6 support is qualified within the text.

A companion to this document is the z/OS Communications Server: IP System Administrator’s Commands, which contains specific system administrator commands used to monitor the network, manage resources, and maintain performance of z/OS CS V1R4.

This document supports z/OS.e™.

Who should use this document

This document is written for users who want to use the applications that are available in z/OS CS V1R4.

Before using this document, you should be familiar with the IBM Multiple Virtual Storage (MVS) operating system, the IBM Time Sharing Option (TSO), and z/OS UNIX System Services and the z/OS UNIX shell. In addition, z/OS CS V1R4 should already be installed and customized for your network. For information about installing, refer to the z/OS Program Directory. For information about customizing, refer to the z/OS Communications Server: IP Configuration Reference.

Where to find more information

This section contains:

- Pointers to information available on the Internet
- Information about licensed documentation
- Information about LookAt, the online message tool
- A set of tables that describes the documents in the z/OS Communications Server (z/OS CS) library, along with related publications

Where to find related information on the Internet

z/OS

z/OS Internet Library

IBM Communications Server product

IBM Communications Server product support
DNS web sites
For more information about DNS, see the following USENET news groups and mailing:

USENET news groups:
comp.protocols.dns.bind

For BIND mailing lists, see:
- http://www.isc.org/ml-archives/
  - BIND Users
    - Subscribe by sending mail to bind-users-request@isc.org.
    - Submit questions or answers to this forum by sending mail to bind-users@isc.org.
  - BIND 9 Users (Note: This list may not be maintained indefinitely.)
    - Subscribe by sending mail to bind9-users-request@isc.org.
    - Submit questions or answers to this forum by sending mail to bind9-users@isc.org.

For definitions of the terms and abbreviations used in this document, you can view or download the latest IBM Glossary of Computing Terms at the following Web address:
http://www.ibm.com/ibm/terminology

Note: Any pointers in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Accessing z/OS licensed documents on the Internet
z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:
http://www.ibm.com/servers/resourcelink

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.

To obtain your IBM Resource Link user ID and password, log on to:
http://www.ibm.com/servers/resourcelink

To register for access to the z/OS licensed documents:
1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select User Profiles located on the left-hand navigation bar.
Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either z/OS Licensed Product Library CD-ROM or IBM Resource Link to print licensed documents.

**Using LookAt to look up message explanations**

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:


or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the z/OS Collection (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your z/OS Collection (SK3T-4269) or from the News section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

**How to contact IBM service**

For immediate assistance, visit this Web site:


Most problems can be resolved at this Web site, where you can submit questions and problem reports electronically, as well as access a variety of diagnosis information.

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

If you would like to provide feedback on this publication, see “Communicating Your Comments to IBM” on page 443.

**z/OS Communications Server information**

This section contains descriptions of the documents in the z/OS Communications Server library.
z/OS Communications Server publications are available:

- Online at the z/OS Internet Library web page at http://www.ibm.com/servers/eserver/zseries/zos/bkserv
- In softcopy on CD-ROM collections.

**Softcopy information**
Softcopy publications are available in the following collections:

<table>
<thead>
<tr>
<th>Titles</th>
<th>Order Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS V1R4 Collection</td>
<td>SK3T-4269</td>
<td>This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R4, in both BookManager® and PDF formats.</td>
</tr>
<tr>
<td>z/OS Software Products Collection</td>
<td>SK3T-4270</td>
<td>This CD includes, in both BookManager and PDF formats, the libraries of z/OS software products that run on z/OS but are not elements and features, as well as the Getting Started with Parallel Sysplex® bookshelf.</td>
</tr>
<tr>
<td>z/OS V1R4 and Software Products DVD Collection</td>
<td>SK3T-4271</td>
<td>This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager and PDF format. This collection combines SK3T-4269 and SK3T-4270.</td>
</tr>
<tr>
<td>z/OS Licensed Product Library</td>
<td>SK3T-4307</td>
<td>This CD includes the licensed documents in both BookManager and PDF format.</td>
</tr>
<tr>
<td>System Center Publication IBM S/390® Redbooks™ Collection</td>
<td>SK2T-2177</td>
<td>This collection contains over 300 ITSO redbooks that apply to the S/390 platform and to host networking arranged into subject bookshelves.</td>
</tr>
</tbody>
</table>

**z/OS Communications Server library**

z/OS V1R4 Communications Server documents are available on the CD-ROM accompanying z/OS (SK3T-4269 or SK3T-4307). Unlicensed documents can be viewed at the z/OS Internet library site.

Updates to documents are available on RETAIN® and in information APARs (info APARs). See Appendix G, “Information APARs” on page 419 for a list of the documents and the info APARs associated with them.

- Info APARs for OS/390® documents are in the document called OS/390 DOC APAR and PTF ++HOLD Documentation which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IIDOCMST/CCONTENTS

- Info APARs for z/OS documents are in the document called z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation which can be found at http://publibz.boulder.ibm.com:80/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS

**Planning and migration:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Communications Server</td>
<td>GC31-8774</td>
<td>This document is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This document also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.</td>
</tr>
</tbody>
</table>
### Resource definition, configuration, and tuning:

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Communications Server: IP Migration</td>
<td>GC31-8773</td>
<td>This document is intended to help you plan for TCP/IP Services, whether you are migrating from a previous version or installing IP for the first time. This document also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with TCP/IP Services.</td>
</tr>
<tr>
<td>z/OS Communications Server: IPv6 Network and Application Design Guide</td>
<td>SC31-8885</td>
<td>This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server’s support of IPv6, coexistence with IPv4, and migration issues.</td>
</tr>
<tr>
<td>z/OS Communications Server: IP Configuration Guide</td>
<td>SC31-8775</td>
<td>This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX® System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document in conjunction with the z/OS Communications Server: IP Configuration Reference.</td>
</tr>
</tbody>
</table>
| z/OS Communications Server: IP Configuration Reference | SC31-8776| This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the z/OS Communications Server: IP Configuration Guide. The information in this document includes:  
  - TCP/IP configuration data sets  
  - Configuration statements  
  - Translation tables  
  - SMF records  
  - Protocol number and port assignments |
| z/OS Communications Server: SNA Network Implementation Guide | SC31-8777| This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the z/OS Communications Server: SNA Resource Definition Reference. |
| z/OS Communications Server: SNA Resource Definition Reference | SC31-8778| This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document in conjunction with the z/OS Communications Server: SNA Network Implementation Guide. |
| z/OS Communications Server: SNA Resource Definition Samples | SC31-8836| This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions. |
| z/OS Communications Server: AnyNet SNA over TCP/IP | SC31-8832| This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.                                                                                                            |
| z/OS Communications Server: AnyNet Sockets over SNA | SC31-8831| This guide provides information to help you install, configure, use, and diagnose sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA. |
| z/OS Communications Server: IP Network Print Facility | SC31-8833| This document is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services. |

### Operation:

About this document  xix
<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>z/OS Communications Server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP User’s Guide and Commands</td>
<td>SC31-8780</td>
<td>This document describes how to use TCP/IP applications. It contains requests that allow a user to log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.</td>
</tr>
<tr>
<td>IP System Administrator’s Commands</td>
<td>SC31-8781</td>
<td>This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator’s commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.</td>
</tr>
<tr>
<td><strong>SNA Operation</strong></td>
<td>SC31-8779</td>
<td>This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.</td>
</tr>
<tr>
<td><strong>Quick Reference</strong></td>
<td>SX75-0124</td>
<td>This document contains essential information about SNA and IP commands.</td>
</tr>
<tr>
<td><strong>Customization:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>z/OS Communications Server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNA Customization</td>
<td>LY43-0092</td>
<td>This document enables you to customize SNA, and includes the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Communication network management (CNM) routing table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Logon-interpreter routine requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Logon manager installation-wide exit routine for the CLU search exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• TSO/SNA installation-wide exit routines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SNA installation-wide exit routines</td>
</tr>
<tr>
<td><strong>Writing application programs:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>z/OS Communications Server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Application Programming Interface Guide</td>
<td>SC31-8788</td>
<td>This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP.</td>
</tr>
<tr>
<td>IP CICS Sockets Guide</td>
<td>SC31-8807</td>
<td>This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS® using z/OS TCP/IP.</td>
</tr>
<tr>
<td>IP IMS Sockets Guide</td>
<td>SC31-8830</td>
<td>This document is for programmers who want application programs that use the IMS™ TCP/IP application development services provided by IBM’s TCP/IP Services.</td>
</tr>
<tr>
<td>Title</td>
<td>Number</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: IP Programmer’s Reference</strong></td>
<td>SC31-8787</td>
<td>This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: SNA Programming</strong></td>
<td>SC31-8829</td>
<td>This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: SNA Programmer’s LU 6.2 Guide</strong></td>
<td>SC31-8811</td>
<td>This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: CSM Guide</strong></td>
<td>SC31-8808</td>
<td>This document describes how applications use the communications storage manager.</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: CMIP Services and Topology Agent Guide</strong></td>
<td>SC31-8828</td>
<td>This document describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The document provides guide and reference information about CMIP services and the SNA topology agent.</td>
</tr>
</tbody>
</table>

**Diagnosis:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>z/OS Communications Server: IP Diagnosis</strong></td>
<td>GC31-8782</td>
<td>This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures</strong> and <strong>z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT</strong></td>
<td>LY43-0088 LY43-0089</td>
<td>These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.</td>
</tr>
<tr>
<td><strong>z/OS Communications Server: SNA Data Areas Volume 1 and SNA Data Areas Volume 2</strong></td>
<td>LY43-0090 LY43-0091</td>
<td>These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.</td>
</tr>
</tbody>
</table>

**Messages and codes:**
<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8790</td>
<td>This document describes the ELM, IKT, IST, ISU, IUT, IVT, and USS messages. Other information in this document includes:</td>
</tr>
<tr>
<td>SNA Messages</td>
<td></td>
<td>• Command and RU types in SNA messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Node and ID types in SNA messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supplemental message-related information</td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8783</td>
<td>This volume contains TCP/IP messages beginning with EZA.</td>
</tr>
<tr>
<td>IP Messages Volume 1 (EZA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8784</td>
<td>This volume contains TCP/IP messages beginning with EZB.</td>
</tr>
<tr>
<td>IP Messages Volume 2 (EZB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8785</td>
<td>This volume contains TCP/IP messages beginning with EZY.</td>
</tr>
<tr>
<td>IP Messages Volume 3 (EZY)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8786</td>
<td>This volume contains TCP/IP messages beginning with EZZ and SNM.</td>
</tr>
<tr>
<td>IP Messages Volume 4 (EZZ-SNM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8791</td>
<td>This document describes codes and other information that appear in z/OS Communications Server messages.</td>
</tr>
<tr>
<td>IP and SNA Codes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**APPC Application Suite:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8809</td>
<td>This document describes the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APIING facilities of the</td>
</tr>
<tr>
<td>APPC Application Suite User's</td>
<td></td>
<td>APPC application suite. Although its primary audience is the end user, administrators and application programmers may also find it useful.</td>
</tr>
<tr>
<td>Guide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8835</td>
<td>This document contains the information that administrators need to configure the APPC application suite and to manage the APIING, ANAME, AFTP,</td>
</tr>
<tr>
<td>APPC Application Suite</td>
<td></td>
<td>and A3270 servers.</td>
</tr>
<tr>
<td>Administration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z/OS Communications Server:</td>
<td>SC31-8834</td>
<td>This document provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their</td>
</tr>
<tr>
<td>APPC Application Suite</td>
<td></td>
<td>application programs.</td>
</tr>
<tr>
<td>Programming</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Redbooks**

The following Redbooks may help you as you implement z/OS Communications Server.

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP Tutorial and Technical</td>
<td>GG24–3376</td>
</tr>
<tr>
<td>Overview</td>
<td></td>
</tr>
<tr>
<td>SNA and TCP/IP Integration</td>
<td>SG24–5291</td>
</tr>
<tr>
<td>IBM Communications Server for</td>
<td>SG24–5227</td>
</tr>
<tr>
<td>OS/390 V2R10 TCP/IP Implementation Guide:</td>
<td></td>
</tr>
<tr>
<td>Volume 1: Configuration and Routing</td>
<td></td>
</tr>
<tr>
<td>IBM Communications Server for</td>
<td>SG24–5228</td>
</tr>
<tr>
<td>OS/390 V2R10 TCP/IP Implementation Guide:</td>
<td></td>
</tr>
<tr>
<td>Volume 2: UNIX Applications</td>
<td></td>
</tr>
<tr>
<td>IBM Communications Server for</td>
<td>SG24–5229</td>
</tr>
<tr>
<td>OS/390 V2R7 TCP/IP Implementation Guide:</td>
<td></td>
</tr>
<tr>
<td>Volume 3: MVS Applications</td>
<td></td>
</tr>
<tr>
<td>Secureway Communications Server</td>
<td>SG24–5631</td>
</tr>
<tr>
<td>for OS/390 V2R8 TCP/IP: Guide to</td>
<td></td>
</tr>
<tr>
<td>Enhancements</td>
<td></td>
</tr>
<tr>
<td>TCP/IP in a Sysplex</td>
<td>SG24–5235</td>
</tr>
<tr>
<td>Managing OS/390 TCP/IP with SNMP</td>
<td>SG24–5866</td>
</tr>
</tbody>
</table>

xxii  z/OS V1R4.0 CS: IP User’s Guide and Commands
### Related information

For information about z/OS products, refer to [z/OS Information Roadmap](SA22-7500). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, as well as describing each z/OS publication.

Relevant RFCs are listed in an appendix of the IP documents. Architectural specifications for the SNA protocol are listed in an appendix of the SNA documents.

The table below lists documents that may be helpful to readers.

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Security in OS/390–based TCP/IP Networks</strong></td>
<td>SG24–5383</td>
</tr>
<tr>
<td><strong>IP Network Design Guide</strong></td>
<td>SG24–2580</td>
</tr>
<tr>
<td><strong>Migrating Subarea Networks to an IP Infrastructure</strong></td>
<td>SG24–5957</td>
</tr>
<tr>
<td><strong>IBM Communication Controller Migration Guide</strong></td>
<td>SG24–6298</td>
</tr>
</tbody>
</table>

### About this document

xxiii
Determining if a publication is current
As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager softcopy are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. The following information describes how to determine if you are looking at the most current copy of a publication:

- At the end of a publication’s order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.

- If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.

- To compare softcopy publications, you can check the last two characters of the publication’s filename (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.
Summary of changes
for SC31-8780-02
z/OS Version 1 Release 4

This document contains information previously presented in SC31-8780-01, which supports z/OS Version 1 Release 2. The information in this document supports both IPv6 and IPv4. Unless explicitly noted, information describes IPv4 networking protocol. IPv6 support is qualified within the text.

New information:
• FTP Enhancements include the following:
  – FTP support for substitution characters during EBCDIC/ASCII single-byte translations
    - FTP.DATA data set statements (see page 70)
    - LOCSite command (see page 192)
    - Site command (see page 196)
  – The following areas contain information in support for Codepage GB18030:
    - New section describing using different MBCS translation tables (see page 85)
    - FTP.DATA data set statements (see page 67)
    - LOCSite command (see page 191)
    - Site command (see page 253)
• The following areas contain information for the new IPv6 support:
  – Applications: FTP (see page 28)

An appendix with z/OS product accessibility information has been added.

Changed information:
• FTP Client sessions that use //DD: support (see page 58)

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Starting with z/OS V1R4, you may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

This document supports z/OS.e.

Summary of changes
for SC31-8780-01
z/OS Version 1 Release 2

This edition has been significantly changed for this release. The z/OS Version 1 Release 1 Communications Server: IP User’s Guide, SC31-8780-00, has been divided into two documents for z/OS V1R2. The two documents are:

• A new companion document, *z/OS Version 1 Release 2 Communications Server: IP System Administrator’s Commands*, SC31-8781-00, which contains the system administrator applications from *z/OS Version 1 Release 1 Communications Server: IP User’s Guide*, SC31-8780-00, and system administrator’s commands from *z/OS Version 1 Release 1 Communications Server: IP Configuration Reference*, SC31-8776-00.

**New information:**

• FTP Security Enhancements
  – IPSF Statistics
    - For PDS members
    - ISPFSStats parameter for FTP.DATA statement
    - ISPFSStats parameter for LOCSITE command
    - ISPFSStats parameter for SITE command
  – Protect Against Bounce Attacks
  – Restrict DIR Output
  – Socksify FTP Client
    - Configuring the FTP Client
    - SOCKSCONFIGFILE parameter for FTP.DATA statement
  – TLS Enablement for FTP
    - Commands
      - CLEAR command
      - PRIVATE command
      - PROTECT command
    - Parameters for FTP.DATA statements
    - Security considerations
    - Start options

• FTP Support Enhancements
  – Debug Enhancements
    - DEBUG command for FTP
    - DUMP command for FTP
    - DEBUG subcommand for SITE command
    - DUMP subcommand for SITE command
  – NETRCLEVEL parameter for FTP.DATA statement
  – READTAPEFormat parameter for LOCSITE
  – READTAPEFormat parameter for SITE
  – RFC Updates
    - FEATURE command
    - LANGUAGE command
    - UTF-8 parameter for FTP.DATA statement
  – Stream Mode Restart — SRESTART command
  – TLSTIMEOUT parameter to FTPDATA statement
  – User-level FTP Server Options
  – VERBOSE command
• SMTP Exit to Filter Unwanted Mail

Changed information
• PROXY command for FTP Security
  – TLS Enablement for FTP
  – User-level FTP Server Options
• Native ASCII Support for FTP
  – SBDataconn parameter for LOCSITE command
  – SBDataconn parameter for SITE command
• FTP Requests in Batch
• FTP Subcommand Code for LOCSTAT
• SBCS Translation Control and Data
• -s keyword for REXEC command
• -s keyword for orexec command

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes
for SC31-8780-00
z/OS Version 1 Release 1

This document contains information also presented in OS/390 V2R10 IBM Communications Server: IP User’s Guide.
Chapter 1. Getting started

Transmission Control Protocol/Internet Protocol (TCP/IP) is a set of
industry-standard protocols and applications that enable you to share data and
computing resources with other computers, both IBM and non-IBM. By using
TCP/IP commands at your workstation, you can perform tasks and communicate
easily with a variety of other systems and workstations. z/OS Communications
Server (z/OS CS) enables the user to interactively run TCP/IP applications (TCP/IP
commands) from both the Time Sharing Option (TSO) and the z/OS shell.

The following subjects are covered in this chapter:
- "Understanding TCP/IP"
- "How TCP/IP uses networks" on page 2
- "Understanding what you can do" on page 4
- "What you need to get started" on page 6
- "Testing commands with loopback" on page 6
- "Obtaining command help" on page 7

Understanding TCP/IP

TCP/IP is a set of protocols and applications that enable you to perform certain
computer functions in a similar manner independent of the types of computers or
networks being used. When you use TCP/IP, you are using a network of computers
to communicate with other users, share data with each other, and share the
processing resources of the computers connected to the TCP/IP network.

A computer network is a group of computer nodes electronically connected by some
communication medium. Each node has the hardware and the programs necessary
to communicate with other computer nodes across this communication medium.
The node can be a PC, workstation, microcomputer, departmental computer, or
large computer system. The size of the computer is not important. The ability to
communicate with other nodes is important.

Computer networks enable you to share the data and computing resources of many
computers. Applications, such as departmental file servers, rely on networking as a
way to share data and programs.

Many forms of communication media are available today. Each is designed to take
advantage of the environment in which it operates. Communication media consist of
a combination of the physical network used to connect the computer nodes and the
language, or protocol, they use to communicate with each other.

Physical network

A physical network consists of electrical wiring and components, such as modems,
bridges, controllers, access units, telephone lines, fiber optic cables, and coaxial
cables. These are used to connect the computer nodes. The physical network can
connect two nodes in a single room or thousands of nodes communicating across
large geographic areas. The most common networks in use today are Local Area
Networks (LANs) and Wide Area Networks (WANs). LANs cover a limited distance,
generally one or two floors or buildings, while WANs, using telecommunication
facilities, are used for longer distances.
Protocols

Network protocols are the rules that define how information is delivered between nodes. They describe the sequence and contents of the data exchanged between nodes on the network. Network protocols determine how a computer node functions during communication with another node, how data is encoded to reach its destination safely, and what path it should follow. Protocols coordinate the flow of messages and can specify which node a message is destined for in the network. A variety of protocols are used to take advantage of the characteristics of each of the physical network types. The most common protocols are Ethernet, 802.3, token ring, X.25, and System Network Architecture (SNA).

Network devices

Refer to the z/OS Communications Server: IP Configuration Reference for more information about network devices.

Addresses

A network address is a component of the communication network and is associated with both hardware and software. The address is the means by which the sending node selects the receiving node for data transfer. It is also used by the receiving node to recognize what data is destined for it. An address is a unique code assigned to every node on a network. But an address is formed differently for different protocols. The length, position, and method used to specify an address are unique for each protocol. A communication node using one protocol cannot recognize the address of another protocol.

How TCP/IP uses networks

TCP/IP consists of a layered structure of protocols ranging from hardware-dependent programs to high-level applications. Each TCP/IP layer provides services to the layer above it and uses the services of the layer below it. The lowest layer, which is next to the physical layer, is not part of TCP/IP. This layer consists of existing protocols, such as Ethernet and token ring. TCP/IP uses the services of this layer to transport data across dissimilar networks, much like a gateway.

Local and remote nodes

A physical network is used by the hosts that reside on that network. Each host is a node on that network. A node is an addressable location in a communication network that provides host processing services. The intercommunication of these nodes gives rise to the concept of local and remote nodes. A local node pertains to a device, file, or host accessed directly from your system. A remote node pertains to a device, file, or host accessed by your system through the network.

Client and server

A server is a computer or a program that contains data or provides services to be used by other computers on the network. Some of the common server types are file, print, and mail servers. They enable your computer to share the data, devices and resources of another computer. There are also servers that provide services to let you execute programs on a computer other than your own. This enables your computer to share the processing power of another computer. Servers are also known as daemons. Generally, a server runs continuously and can handle the requests of multiple clients simultaneously.
A client is a computer or a program that requests services or data from a server. A client could, for example, request that a file located at the server be sent across the network to the client.

**TCP/IP addresses**

An address enables data to be routed to the chosen destination. Each destination in your network, as well as any other TCP/IP network you have access to, can be uniquely identified by its assigned address (either a 32-bit IPv4 address in *dotted decimal* notation, or a 128-bit IPv6 address in *colon hexadecimal* notation).

- **IPv4 TCP/IP address**
  - An IPv4 TCP/IP address is a 32-bit number written in *dotted decimal* notation. This scheme is numeric and consists of four groups separated by a period (·). For example, 9.67.1.100 represents a single host on a single network. 193.5.86.9 represents another host on another network.

- **IPv6 TCP/IP address**
  - An IPv6 TCP/IP address is a 128-bit number written in *colon hexadecimal* notation. This scheme is hexadecimal and consists of eight 16-bit pieces of the address. For example, ::x::x:x::x represents a single host on a single network. Alternate notations described in RFC 2373 are acceptable. For example, FEDC:BA98:7654:3210:FEDC:BA98:7654:321 or :.1. To indicate IPv6 prefixing use a slash followed by the number of prefix bits. For example, use 12AB:0:0:CD30::/60 to indicate the prefix X'12AB00000000CD3'.

Most TCP/IP commands require you to include the address of the remote host where the server you wish to access resides. Each link (physical or virtual) on a host has an IP address.

**Network names**

An alternative to supplying a numeric address is to use the host name, rather than the address, in TCP/IP commands. (Each host may be assigned at least one name.) Your local host can resolve the name you supply in a command into the correct numeric address. The names are translated using either a translation file or an application known as a *name server*. Your ability to use *network names* depends on how your TCP/IP network has been designed and which features have been installed.

**Ports and port numbers**

The use of *ports* and their identifying numbers are an extension to the addressing scheme. Once the address is used to deliver data to the desired host on the network, the *port number* is used to identify the process for which the data is used. This enables one host to provide more than one service.

How you define the port number depends on your configuration. Some applications make use of standard, or well-known, port numbers. Two applications at the same address cannot use the same port number. If you are configuring your system with multiple instances of TCP/IP on the same system, however, they will have different addresses and therefore the same port number can be used for the same function on each stack.

TCP/IP assumes the well-known port number unless you explicitly specify otherwise when entering a TCP/IP command. A port number is entered as a decimal number on TCP/IP commands. For those cases when you are requesting the services of a user-developed server, you need to know the port number of that server.
Understanding what you can do

You can perform many functions from either TCP/IP environment: TSO or z/OS UNIX System Services (z/OS UNIX). See Appendix E, "How to read a syntax diagram" on page 405 for additional information about command syntax.

z/OS CS commands provide a set of basic functions that include:

- Logging on to other hosts
- Transferring data sets and files between hosts
- Sending and receiving mail
- Using other hosts
- Printing to or from other hosts

Logging on to other hosts

The Telnet protocol provides a standardized interface that enables terminal devices and terminal-oriented processes on hosts that support TCP/IP to communicate with each other. The TSO TELNET command runs the MVS telnet client that enables you to log on to a remote host from TSO as though you are directly attached to that host. This client supports the Telnet 3270 protocol. The MVS Telnet client does not run in the z/OS UNIX environment.

Connecting to the z/OS UNIX Telnet server from any client results in a session with the z/OS shell as if the user had entered UNIX System Services from TSO in line mode or character mode. Once the z/OS UNIX Telnet session has been established, you can enter any UNIX System Services command that can be issued from within the z/OS shell.

See Chapter 2, “Logging on to a host using TELNET” on page 9 for more information about TELNET.

Transferring data sets between hosts

When data is created or stored at one host but is processed by another host, some method for transferring the data between hosts is necessary. TCP/IP implements File Transfer Protocol (FTP), a protocol for transferring files between any two hosts. The FTP command invokes the z/OS FTP client. From the FTP client, you can copy data sets and files between your local host and any host that implements an FTP server.

The FTP command provides subcommands that enable you to change the local and remote directories, set the transmission character code, list remote files, delete remote files, and send and receive files between hosts. You can use z/OS FTP client to perform Structured Query Language (SQL) queries and to submit jobs to JES for batch processing, as well as file transfer. FTP includes optional security features such as Kerberos and TLS authentication, and server user ID and password verification.


Sending and receiving mail

The Simple Mail Transfer Protocol (SMTP) is a TCP/IP application that is used to transport electronic mail. Electronic mail enables you to send notes, messages, letters, or correspondence to others on the network. It is similar to sending a letter
through the post office. You compose the message just as you would an ordinary
letter, address the letter to one or more people and possibly carbon copy others.
You enclose copies of the letter in envelopes, address them to the recipients, and
give them to the delivery system. You expect the mail to be delivered to the correct
address available for pickup when the recipient is ready. And you want any
undeliverable mail returned to you. You can even keep a log of the mail you send
and receive. The following commands are available to let you send and receive
mail:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTPNOTE</td>
<td>Composes and sends mail from you to users on local or remote hosts. The TSO SMTPNOTE command helps you address the mail, set up a copy list, and enter the text of the message. The date, time, and your address are included automatically.</td>
</tr>
<tr>
<td>z/OS UNIX sendmail</td>
<td>Composes and sends mail from you to users on local or remote hosts. The z/OS UNIX sendmail command helps you address the mail, set up a copy list, and enter the text of the message. The date, time, and your address are included automatically.</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>Retrieves data that has been sent to you through the Job Entry Subsystem (JES). You use the RECEIVE command for your TCP/IP mail just as you do for other mail, messages, and data sets that are sent to you.</td>
</tr>
</tbody>
</table>

See Chapter 6, “Sending electronic mail using SMTP commands” on page 285 for more information about the SMTP mail facility.

See Chapter 7, “Sending electronic mail using z/OS UNIX sendmail” on page 319 for more information about the z/OS UNIX mail facility.

Remote command execution

Just as there are occasions when you want to transfer data to a host where it can be processed, there are also occasions when you want to process the data where it exists and send the processing results to another host. The data sets or files could be too large to transfer efficiently or all the data might be kept at one host for security reasons. The computing power necessary to perform some tasks could be more than your host is capable of or the only licensed copy of a required program might reside at some other host. TCP/IP provides a command that enables you to use the processing resources of other hosts. The TSO commands REXEC and RSH and the OE shell command orexec run client programs enable you to send any command that is valid in the remote host shell environment and receive the results at the local host. The remote host must be running a rexec server, an rsh server or both. A user ID and password provide security checking at the remote host. The command sent to the remote host must not require user interaction to complete. Refer to the z/OS Communications Server: IP System Administrator’s Commands for more information about these commands.

Printing to or from other hosts

You can print reports, documents, listings, and so on completely independent of where the job or process that created them was executed by routing the data sets to a remote host for printing. Four TSO commands are provided for remote printing. These TSO commands do not run in the z/OS shell environment and do not support printing of HFS files.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPQ</td>
<td>Enables you to query a printer queue on a remote printer. You can query a printer queue for a specific job, a specific user ID, or all the jobs in a remote printer queue.</td>
</tr>
<tr>
<td>LPR</td>
<td>Prints a data set on a remote printer. A variety of options enables you to specify how and where the data set is printed.</td>
</tr>
<tr>
<td>LPRM</td>
<td>Removes a job from a printer queue.</td>
</tr>
<tr>
<td>LPRSET</td>
<td>Specifies remote printer and remote host names when they are not specifically included in the Line Print commands.</td>
</tr>
</tbody>
</table>

What you need to get started

TCP/IP is a part of your z/OS system. To use it you need a TSO user ID and password. If you are already a TSO user, you can begin using TCP/IP.

In order to use z/OS CS for V1R4 applications, you must be authorized to use z/OS UNIX System Services. For information about z/OS UNIX, refer to the z/OS UNIX System Services User’s Guide.

Ensure that you have the following before proceeding:

User IDs and passwords
You should have a user ID and password for each host you intend to use that requires user authorization and authentication. This includes most hosts you use. Some hosts on a TCP/IP network use a user ID of anonymous and a password of guest to permit all interested parties access to data sets contained at that host, but that is the exception and not the rule.

Host names
TCP/IP commands require that you know the name or IP address of the remote host you want to use.

Authorizations for data and programs
Your ability to access data sets and programs on remote hosts depends on the data security system used by that host. You might require authorization by the Resource Access Control Facility (RACF) or other security programs before you can gain access to data sets, commands, or other resources on remote hosts.

Electronic mail addresses
To send mail electronically, you need the e-mail address of the users you wish to send mail to and they need to know your electronic mail address to send you mail.

Printer names
You need the printer name and name of the remote host to which it is attached to print using TCP/IP.

Testing commands with loopback
In order to test your local machine, an address is reserved that always refers to your local host rather than any other hosts on a network. For IPv4, this class A network address is 127.0.0.1. For IPv6, the reserved loopback address is ::1. You can also specify loopback as the host name. Not all commands and clients shipped with z/OS CS V1R4 support IPv6 addresses or hostnames that resolve to IPv6 addresses.
You can use the loopback address with any TCP/IP command that accepts IP addresses. When you issue a command with the loopback address, the command is sent out from your local host's client and continues until it reaches the IP layer on your local host. The command is then sent on to your local host's server.

**Note:** Any command or data that you send using the loopback address never actually goes out on any network.

You can also use a nonloopback local IP address for testing. It can be any local IP address assigned to a device, even though the device may not be active, but cannot be a multicast or broadcast address. The nonloopback address provides a faster response but may not reach the IP layer on your host.

The loopback address is commonly used as the first step in diagnosing network problems. The information you receive indicates the state of your system and checks to ensure that the client and server code for the function you are testing is operating properly. You should see the same response as for a normal, successful command. If the client or server code is not operating properly, the same message that would be returned for an unsuccessful command is returned.

### Obtaining command help

Commands typically support HELP or -? options that allow users to obtain online help. Additionally, end user commands supported in the z/OS UNIX System Services shell environment support man pages (manual pages). For example, typing `man ftp` will display the manual pages for the `ftp` command.
Chapter 2. Logging on to a host using TELNET

The TELNET protocol provides a standardized interface that enables terminal devices and terminal-oriented processes on hosts that support TCP/IP to communicate with each other.

The following subjects are covered in this chapter:
- "Using the TSO TELNET command"
- "Using the TELNET subcommands" on page 13
- "Using the TELNET function keys" on page 23
- "Suppressing carriage return and line feed" on page 23
- "Using TELNET 3270 DBCS transform mode" on page 24

Using the TSO TELNET command

When you use the TSO TELNET command you are running a Telnet client to connect to a remote host running a Telnet server. The data displayed on your terminal is managed by the Telnet client which communicates via TCP/IP with the Telnet server at the remote host. As a result, the operation of your terminal may differ from what you are used to seeing when you are directly logged on to TSO or to another MVS application. For example, the remote host may be running UNIX, VM or any other operating system that provides a telnet server. You need to use the terminal operation procedures of the remote host operating system while you have a TELNET session with that remote host.

TELNET management of your terminal for the remote host can also cause operational differences. For example, the function keys described in "Using the TELNET function keys" on page 23 can result in different actions.

When all of the display data will not fit on your screen, Linemode displays the HOLDING message in the lower right corner of your screen. If this message appears, press the CLEAR key to see the rest of the data.

If your TELNET session ends for any reason, the following message is displayed:
Subsession ended. <ENTER> to return to TSO.

If you invoke the services of the MVS telnet 3270 server from a telnet client that is not telnet 3270 capable, you will not be able to use applications in full-screen mode. Once in line-mode all nested telnet sessions will continue to be line mode. If you use TELNET in line mode to access an MVS or VM TELNET server, all subsequent nested TELNET requests are automatically connected in line mode as a start-stop TTY terminal, and transparent (full-screen) operations are not possible.

When you return to TSO, a message explaining why the TELNET session ended is displayed. The following is an example of what is displayed when you return to TSO:

TELNET terminated -- Foreign host is no longer responding

Note: The z/OS TELNET client does not support the Secure Sockets Layer (SSL) protocol.
TELNET command

Purpose
The TELNET command enables you to log on to a foreign host that supports TCP/IP.

Notes:
1. For information on how to read syntax diagrams, see Appendix E, “How to read a syntax diagram” on page 405.
2. The TSO TELNET command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

Format

```
> [Z(qOOOO] [Z(qOOOO
TELNET

foreign_host port_number
Help

{ Linemode DEBUG TRANslate data_set_name }
```

Parameters

- **foreign_host**
  Specifies the name or IP address of the local or remote host. If you do not specify the name or IP address of the host, you are prompted for the `foreign_host`. This must be an IPv4 address or a hostname that resolves to an IPv4 address.

- **port_number**
  Specifies the port number to which you want to connect on the host. The default is well-known port 23.

Help
Provides a description of the TELNET command, its subcommands, and how it operates.

Linemode
Uses the line mode and prevents operation in the transparent mode.

In line mode, the foreign host output is displayed on your screen one line at a time, without full-screen capabilities.

**Note:** You cannot use the TELNET command to log on to an MVS host from an existing MVS line mode TELNET session. In this situation, the error message TELNET requires a 327x-Type terminal is displayed.

In transparent mode, the foreign host full-screen capabilities are functional on your local terminal.

Transparent mode is the default.

DEBUG
Causes TELNET client-trace data, including the data transferred to and
received from the TELNET server, to be written to a data set defined by the DEBUGFIL DD statement in the user TSO LOGON procedure, or as specified by issuing the TSO ALLOC command.

The following is an example of the DEBUGFIL DD statement:

```
//DEBUGFIL DD DSN=USER28.TELNET.TRACE,DISP=OLD
```

The following is an example of the TSO ALLOC command:

```
ALLOC DDNAME(DEBUGFIL) DSNAME(USER28.TELNET.TRACE) OLD
```

**TRANslate data_set_name**

Specifies the name of a nonstandard translation table. If you specify this parameter, TELNET uses the translation table in the `user_id.data_set_name.TCPXLBIN` data set, rather than the standard translation tables `user_id.TELNET.TCPXLBIN` or `hlq.TELNET.TCPXLBIN`.

If `user_id.data_set_name.TCPXLBIN` does not exist, TELNET uses `hlq.data_set_name.TCPXLBIN`.

If `user_id.data_set_name.TCPXLBIN` and `hlq.data_set_name.TCPXLBIN` do not exist, or if they were incorrectly created, TELNET ends with an error message. A nonstandard translation table is used in line mode only.

**Examples**

- To log on to a host with an IP address of 1.1.2.3, enter:
  ```
  TELNET 1.1.2.3
  ```

  The following is displayed:

  ```
  System: READY
  User: TELNET 1.1.2.3
  System: MVS TCP/IP TELNET CS V1R2
  Connecting to 1.1.2.3, port TELNET (23)
  ***
  Using Transparent Mode...

  Notes on using TELNET when in Transparent Mode:
  - To enter TELNET Command, Hit PA1
  ***
  ```

- If your user ID is RON and the translation table RON.EXAMPLE.TCPXLBIN is required rather than the standard one, you should enter:
  ```
  TELNET 1.1.2.3 (TRANslate EXAMPLE
  ```

  - If the remote host is neither MVS nor VM, and you specify a nonstandard translation table, a linemode connection is automatically used.
  - If the remote host is an MVS or VM host and you specify a nonstandard translation table without the linemode parameter, the nonstandard translation table is ignored.
  - If the host is an MVS or VM host and both the linemode parameter and a nonstandard translation table are specified, the nonstandard translation table is used.

**Usage**

- The minimum abbreviation for each parameter is shown in uppercase letters.
- TELNET normally operates in transparent mode. In 3270 transparent mode, all full-screen capabilities of the remote host are functional at your local display station, but the **PA1** key is the only special-function key whose intended
function is passed to the application by Telnet. In line mode, the remote host output is displayed on your screen one line at a time, without full-screen capabilities.

- The TELNET command supports IBM 3270-type display stations. Examples of supported display stations are:
  - IBM 3178 Display Station
  - IBM 3179 Display Station
  - IBM 3180 Display Station
  - IBM 3191 Display Station
  - IBM 3192 Display Station
  - IBM 3193 Display Station
  - IBM 3194 Display Station
  - IBM 3275 Display Station Model 2
  - IBM 3276 Control Unit Display Station Models 2, 3, and 4
  - IBM 3277 Display Station Model 2
  - IBM 3278 Display Station Models 2, 3, 4, and 5
  - IBM 3279 Color Display Station Models 2 and 3
Using the TELNET subcommands

You must be in the TELNET environment to use the TELNET subcommands.

To invoke a TELNET subcommand while you are logged on to the foreign host, press the PA1 key (transparent mode) or the designated PF key (line mode). After you press the PA1 or PF key, you will be prompted to enter a TELNET subcommand. You can enter TELNET subcommands in uppercase or lowercase. Table 1 lists the TELNET subcommands.

Table 1. TELNET subcommands

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AO</td>
<td>Stops the display of information</td>
<td>14</td>
</tr>
<tr>
<td>AYt</td>
<td>Queries the existence of the connection</td>
<td>15</td>
</tr>
<tr>
<td>Brk</td>
<td>Sends a Break or Attn keystroke</td>
<td>16</td>
</tr>
<tr>
<td>Help or ?</td>
<td>Displays help information</td>
<td>17</td>
</tr>
<tr>
<td>Ip</td>
<td>Interrupts the current process</td>
<td>18</td>
</tr>
<tr>
<td>Pa1</td>
<td>Sends a PA1 keystroke in transparent mode</td>
<td>19</td>
</tr>
<tr>
<td>Quit</td>
<td>Disconnects from the foreign host</td>
<td>20</td>
</tr>
<tr>
<td>Synch</td>
<td>Clears the data path</td>
<td>21</td>
</tr>
</tbody>
</table>

Note: The minimum abbreviation for each subcommand is shown in uppercase letters.
AO—Terminate output display

**Purpose**
Use the AO (Abort Output) subcommand to stop the display of output.

**Format**

```
AO
```

**Parameters**
There are no parameters for this subcommand.

**Usage**
The AO subcommand is used to clear any output that has already been produced, but has not been displayed on your terminal.
AYT—Query the connection

Purpose
Use the AYT (Are You There) subcommand to query the existence of the connection.

Format

\[ \text{AYT} \]

Parameters
There are no parameters for this subcommand.

Usage
- You can use the AYT subcommand to check for the existence of a TELNET connection. For example, if you feel that a command is taking longer than it should to complete, issue the AYT subcommand to test whether the connection is still active.
- If the connection exists and you are operating in transparent mode, the terminal makes a sound. If you are operating in line mode, you receive a message from the TELNET server.
BRK—Send a Break or Attention keystroke to a host

Purpose
Use the BRK subcommand to send a Break or Attention keystroke to the remote session.

Format

```
> [Z(qOOOO-[ZU3OOOO
```

Parameters
There are no parameters for this subcommand.

Usage
You can use the BRK subcommand to end a command without terminating the TELNET session.
HELP or ?—Display help information

Purpose
Use the HELP (or ?) subcommand to access the help facility.

Format

Usage

• After your TELNET connection is established, your display station screen is controlled by the foreign host, so the help you see is different for line mode and transparent mode. The help available in transparent mode is abbreviated because TELNET does not have control of the screen.

• When you invoke the HELP or ? subcommand in line mode, TELNET displays the help information one line after another, as in the following example:

Once connected, follow the log in and usage procedures of the remote host.
To invoke one of several TELNET commands, hit a PF key (PF4-12, PF16-24), and then enter any of the following commands:
Help or ? -- Receive (this) assistance
AYT -- Are You There?
AO -- Abort Output
BRK -- Break
IP -- Interrupt Process
SYNCH -- Clear data path, except for TELNET commands
Quit -- Quit the TELNET session

The following PF settings are in force:
PF1 or 13 -- Retrieve previous input line
PF2 or 14 -- Scroll halfway up
PF3 or 15 -- Turn off display of user-line; designed to be used before entering password

For control characters, enter \c or `c where c is:
"0": 0D, "a" - "z" or "A" - "Z": 0x01-0x1A
"2" - "6": 0x1B-0x1F
"{": 0x5B, "}": 0x5D, ":": 0x7F

• When you invoke the HELP or ? subcommand in transparent mode, TELNET overwrites one line of the current screen with the help information, as in the following example:

Valid TELNET cmds: AO, AYT, BRK, IP, PA1, QUIT, SYNCH.
IP—Interrupt the process

Purpose
Use the IP subcommand to interrupt the current process running on the remote host.

Format

```plaintext
Ip
```

Parameters
There are no parameters for this subcommand.

Usage
You can use the IP subcommand if you want to stop a process that is in a loop, or when you want to stop a process that you inadvertently started.
PA1—Send the PA1 keystroke to a host

**Purpose**
Use the PA1 subcommand to send a PA1 keystroke to the remote session in transparent mode.

**Format**
```
> [Z(qOOOO>
Pa1
```

**Parameters**
There are no parameters for this subcommand.

**Usage**
- The PA1 subcommand operates only in transparent mode. This subcommand replaces the PA1 attention key on the remote host.
- When there are nested TELNET sessions, use the PA1 key to enter a TELNET subcommand in the first open TELNET session. To enter a TELNET subcommand in the second open TELNET session, send a PA1 subcommand from the first session.
- You would normally interrupt a PING command by pressing PA1. However, in a transparent mode TELNET session, this key is used to invoke a TELNET subcommand. You would issue a PA1 subcommand to interrupt the PING command instead.
QUIT—End the TELNET session

Purpose
Use the QUIT subcommand to end the TELNET session.

Format
]

Parameters
There are no parameters for this subcommand.

Usage
• You should use the QUIT subcommand carefully because it can create an MVS error condition. If you do not reconnect within a timeout period, your TELNET session is canceled.
• If you are logged on to an application on a remote host, and that application is defined as disconnectable to VTAM® and TCP/IP, you can use the QUIT subcommand to disconnect from the remote host without logging off the application.
• When you want to end a logon session with the host, use the logoff procedure of the host.

For more information about defining applications to VTAM and TCP/IP, refer to the Z/OS Communications Server: IP Configuration Reference.
SYNCH—Clear the data path

**Purpose**
Use the SYNCH subcommand to clear the data path.

**Format**

```
  Synch
```

**Parameters**
There are no parameters for this subcommand.

**Usage**
The SYNCH subcommand clears the data path to the host, except for any TELNET subcommands in the data path. This subcommand enables you to ensure that commands issued when the TELNET server is inactive are not executed when the TELNET server becomes active.
` and `—Send ASCII control characters to a host in line mode

**Purpose**
Use the ` character and the grave accent (´) in line mode to indicate a control character.

**Format**
```
control_character
```

**Parameters**
*control_character*
Indicates the ASCII control character that you want to send to the host. The purpose of each control character is specific to the remote host.

**Examples**
To send `\x02`, use either: `p or `p.

**Usage**
- If you want to use ` or ` without indicating a control character, you must enter these characters twice.
- The ASCII control characters are shown in Table 2.

<table>
<thead>
<tr>
<th>Character input</th>
<th>ASCII output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>A – </code>Z</td>
<td>01 – 1A (Ctrl-a – Ctrl-z)</td>
</tr>
<tr>
<td>`{</td>
<td>5B (left square bracket - [)</td>
</tr>
<tr>
<td>`}</td>
<td>5D (right square bracket - ])</td>
</tr>
<tr>
<td><code>2 – </code>6</td>
<td>1B – 1F</td>
</tr>
<tr>
<td>`#</td>
<td>7F (DEL)</td>
</tr>
</tbody>
</table>

Table 2. ASCII control characters
Using the TELNET function keys

This section describes the functions that are assigned to PF keys when you invoke TELNET in transparent mode and line mode.

Transparent mode function keys

In transparent mode, the only function key available is the PA1 attention key. It is used to invoke a TELNET subcommand. If there is more than one nested TELNET session, the PA1 key is used to invoke a TELNET subcommand for the first TELNET session.

See “PA1—Send the PA1 keystroke to a host” on page 19 for information about how to send the PA1 keystroke to the foreign host session.

Line mode function keys

Table 3 describes the function keys that are available in line mode.

Table 3. TELNET function keys in line mode

<table>
<thead>
<tr>
<th>Function key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF4 – PF12</td>
<td>Enables you to invoke a TELNET subcommand. After pressing one of these function keys, enter a subcommand or enter Help to get a list of valid subcommands.</td>
</tr>
<tr>
<td>PF16 – PF24</td>
<td></td>
</tr>
<tr>
<td>PF1 , PF13</td>
<td>Retrieves the previous input line, except when the line was entered in hidden mode for security reasons.</td>
</tr>
<tr>
<td>PF2 , PF14</td>
<td>Scrolls halfway up the screen.</td>
</tr>
<tr>
<td>PF3 , PF15</td>
<td>Turns off input line display so data is not echoed to the screen. For example, use either of these keys before entering your password to keep it from being displayed.</td>
</tr>
</tbody>
</table>

Suppressing carriage return and line feed

It is useful if the command environment of the foreign host responds when you enter a single character, without the need for a carriage return and line feed after that character. This function is also useful when your cursor is at the end of the input field, but you want to continue the line without introducing a carriage return.

Figure 1 and Figure 2 on page 24 show the output of a BSD UNIX program called MORE™. This program displays one line or one page at a time. A carriage return character (CR) causes it to display one line, while a blank character causes it to display one page. If you are executing this program from an MVS host, use the grave accent (´) character to suppress the CR that is normally sent when you press Enter.
Using TELNET 3270 DBCS transform mode

When 3270 DBCS transform mode is configured for the MVS TELNET server, all new line-mode sessions to the server are introduced with a panel where you can select transform mode or line mode.

TELNET 3270 DBCS transform mode is used to provide 3270 DBCS emulation, while the 3270 processing is done only at the host end of the connection. This enables full-screen access from non-3270 terminals.
TELNET 3270 DBCS transform mode supports terminals of the VT100/VT220 family of terminals, including VT100 and VT282.

To log on to the server using 3270 DBCS transform mode, specify the LINEMODE option on the TELNET client command line. The following banner is displayed if transform mode is available:

IBM TCP/IP TELNET SERVER DBCS SERVICE START AT HH.MM.SS ON MM/DD/YY
KEY-IN YOUR TERMINAL TYPE & CONVERSION TYPE:
Terminal and conversion type

Format
When the DBCS banner appears, enter the required terminal and conversion type.

- VT282
- VT100
- TTY
- SJISKANJI
- JIS78KJ
- JIS83KJ
- BIG5
- EUCKANJI
- DECKANJI
- HANGEUL
- KSC5601
- SCHINESE
- TCHINESE

Parameters

VT100
VT100 terminal type, full-screen mode—no DBCS support

VT282
VT282 terminal type, full-screen mode—with DBCS support

TTY
Line mode—no DBCS support. This option bypasses operation in transform mode.

SJISKANJI
Shift JIS Kanji DBCS conversion

JIS78KJ
JIS Kanji 1978 DBCS conversion

JIS83KJ
JIS Kanji 1983 DBCS conversion

BIG5
Big-5 DBCS conversion

EUCKANJI
Extended UNIX code Kanji DBCS conversion

DECKANJI
DEC Kanji DBCS conversion

HANGEUL
Hangeul DBCS conversion

KSC5601
Korean Standard code KSC-5601 DBCS conversion

SCHINESE
Simplified Chinese DBCS conversion

TCHINESE
Traditional Chinese (5550) DBCS conversion
Usage
- Do not enter any spaces between the terminal type and the slash character (/), or between the slash character (/) and the conversion type. For example, to specify a VT282 terminal with Shift JIS Kanji DBCS conversion, enter the following:
  VT282/SJISKANJI
- If the conversion type is not specified, it defaults to the CODEKIND specified in the TNBCSTM configuration data set. If neither terminal type nor conversion type is specified, the terminal type defaults to VT282 and the conversion type to the CODEKIND specified in the TNDBCSTM configuration data set.
- TELNET 3270 with DBCS transform mode supports a screen size of 24 by 80. Unpredictable results may occur when using a larger screen size.
- The maximum number of concurrent TELNET 3270 DBCS Transform connections is 250.

Context
For more information about using translation tables, refer to the z/OS Communications Server: IP Configuration Reference.

For information about character sets, see "Character set cross reference table" on page 395.

For information about the TELNET extensions for terminals other than the 3270 family, see Appendix C, “TELNET extensions” on page 395.
Chapter 3. File Transfer Protocol (FTP)

The FTP command runs the FTP client program that enables you to transfer data sets and files between your local host and another host running an FTP server. Using the FTP command and its subcommands, you can sequentially access multiple hosts without leaving the FTP client.

This chapter describes:

- "Using the FTP command"
- "Getting help" on page 37
- "Establishing and exiting a connection" on page 38
- "Obtaining status and system information" on page 40
- "Working with directories on the remote host" on page 41
- "Working with directories on the local host" on page 45
- "Security considerations" on page 46

Using the FTP command

Before transferring files between your local host and a remote host, or using any other FTP functions, you must use the FTP command to enter the FTP environment. The FTP command can be used from either the TSO environment or the z/OS UNIX shell.
FTP command—Enter the FTP environment

Purpose
Use the FTP command to enter the FTP environment.

Note: In a z/OS UNIX environment, you receive an error message if you use this form of the FTP command:

```
ftp 1.1.2.3 (trace
```

Instead, use the standard UNIX flag (for example, -d) or precede the left parenthesis with an escape character, such as the backslash (\):

```
ftp 1.1.2.3 \(trace
```

Format

```
ftp
```

```
-a NEVER GSSAPI TLS NEVER GSSAPI TLS
-r NEVER GSSAPI TLS
-d
-e
-g
-i
-n
-p tcpip
-t data_set_name
-v
-w nn
-x

( Exit TRACe TImeout nn
TCP tcpip
```

```
TRANslate data_set_name NOOPTMSs
```

Parameters

-a

NEVER
FTP will not attempt authentication upon initial connection.

GSSAPI
FTP will attempt auto-authentication upon initial connection. FTP attempts to authenticate to the FTP server by sending the AUTH command specifying GSSAPI as the authentication type. Once the authentication type is accepted, the authentication protocol will proceed by issuing ADAT commands.

TLS
FTP will attempt auto-authentication upon initial connection. FTP attempts
to authenticate to the FTP server by sending the AUTH command specifying TLS as the authentication type.

-d Starts the generation of tracing output. Equivalent to TRACE.

-e Terminates FTP for certain FTP errors with a nonzero MVS return code. Equivalent to EXIT.

-g Turns off metacharacter expansion (globbing). Equivalent to the GLOB subcommand.

-i Turns off interactive prompting for MDELETE, MGET, and MPUT subcommands. Equivalent to the PROMPT subcommand.

-n Inhibits automatic login. If you have defined a NETRC data set, it will not be used to log in to this session.

-p tcpip
Indicates the name of the TCP on the local host to which the FTP client should connect. This parameter is ignored if your system is not configured for multiple instances of TCP/IP. This is equivalent to TCP tcpip.

-t data_set_name
Specifies the name of a nonstandard translation table. Equivalent to TRANSLATE data_set_name.

-r The option -r is the same as -a except that the AUTH command must be accepted by the server. If it is not, then the client ends the session.

NEVER
FTP will not attempt authentication upon initial connection. This option overrides a value in the FTP.DATA file that would cause authentication.

GSSAPI
FTP will attempt auto-authentication upon initial connection. FTP attempts to authenticate to the FTP server by sending the AUTH command specifying GSSAPI as the authentication type. Once the authentication type is accepted, the authentication protocol will proceed by issuing ADAT commands. If the authentication type is not accepted the client will terminate the connection.

TLS
FTP will attempt auto-authentication upon initial connection. FTP attempts to authenticate to the FTP server by sending the AUTH command specifying TLS as the authentication type.

-v Enables verbose mode. This parameter gives you extra information (such as message IDs) when running in z/OS UNIX.

Note: If you are running the FTP client in TSO, you can have message numbers prepended to client messages by using the profile option MSGID.

-w nn
Specifies the number of seconds to be used for the TIMEOUT parameters. Equivalent to TIMEOUT nn.

-x Client attempts to negotiate encryption (data and command protection level of private) immediately after a successful authentication negotiation.
foreign_host
Specifies the name of the host to which you are connecting. Specify the host by its host name or its IP address. The host can be a remote host or your local host.

You are prompted for a host name if you do not specify a foreign_host with the FTP command. If you specify foreign_host incorrectly, or if the host is not accessible, you enter the FTP environment without connecting to a host. You should then use either the OPEN subcommand to attempt another connection with a host, or the QUIT subcommand (or Ctrl-C, in z/OS UNIX) to exit the FTP environment.

port_number
Specifies the port number of the FTP server on the remote host. The default is well-known port 21. The maximum port number that can be specified is 65534. This parameter should not be used unless you are sure there is a server listening on a port other than the well-known port 21 at the destination.

Exit
Terminates FTP, for certain FTP errors, with a nonzero MVS return code. See Table 14 on page 98 for examples of these return codes.

Exit=nn
Terminates FTP with a nonzero return code of your choice when an FTP error occurs. Valid values are in the range 0–255.

TRACe
Starts the generation of tracing output. TRACe is used in debugging.

Timeout nn
Specifies the number of seconds (nn) to be used for the following TImeout parameters:
- MyopenTime
- DconnTime
- CconnTime
- InactTime
- DataCtTime

See “Changing local site defaults using FTP.DATA” on page 63 for a description of each of these timers and their default values. FTP.DATA can be used to customize each timer individually.

If you specify a value outside the range 15–86400, the Timeout parameters are not changed.

Note: If the value is not a number, all Timeout values are set to 0.

TCP tcpip
Indicates the name of the TCP on the local host to which the FTP client should connect. This parameter is ignored if your system is not configured for multiple instances of TCP/IP.

Note: You must specify this value as a parameter, not as a value in the FTP.DATA data set. You may choose to specify this value with the TCPIPJOBNAME statement in the resolver configuration file.

TRANslate data_set_name
Specifies the data set name of a nonstandard translation table. If you specify this parameter, FTP uses the translation table in the
**user_id.data_set_name.TCPXLBIN** data set, rather than the standard translation table provided with TCP/IP (**hlq.STANDARD.TCPXLBIN**). The **hlq.STANDARD.TCPXLBIN** data set is never used if you specify the TRANSLATE parameter.

If **user_id.data_set_name.TCPXLBIN** does not exist, FTP uses **hlq.data_set_name.TCPXLBIN**. If neither data set exists, or if they were incorrectly created, FTP ends with an error message.

Since the TRANslate parameter also dictates the search order for DBCS translation tables, you may wish to use a customized DBCS translation table but not require a modified SBCS translation table. If this is the case, copy **hlq.STANDARD.TCPXLBIN** into the nonstandard TCPXLBIN translation table data set to ensure FTP will start.

**Notes:**

1. Use the CTRLConn and SBDdataconn statements in your local FTP.DATA data set to specify different SBCS tables for the control and data connections, or use the LOCSITE SBDdataconn subcommand to change the SBCS translation for the data connection. For information on specifying these statements, see “Using different SBCS translation tables for the control and data connections” on page 85.

2. If you require the use of a customized DBCS translation table, but cannot or do not wish to use the TRANSLATE parameter, you can name the data set such that it will be found in the client search order (e.g. **userid.FTP.TCPdbBIN**). Refer to the **z/OS Communications Server: IP Configuration Reference** for information about the DBCS translation table search order. FTP does not terminate because it fails to find a nonstandard DBCS translation table data set.

**NOOPTMSSs**

Use the NOOPTMSSs parameter to prevent TCP from building optimal size segments (OPTMSs). The default is for TCP to build optimal size segments.

**Note:** The z/OS TCP/IP stack always operates as if NOOPTMSSs were specified, regardless of what you entered. However, if any stack in use does support the SO_OPTMSS option, you can specify NOOPTMSSs to prevent FTP from using it.

The option can be specified in the following ways:

- //STEP EXEC PGM=FTP,PARM='... (EXIT NOOPTMSS TRACEx'
- By way of TSO: FTP ... (EXIT NOOPTMSS TRACEx
- INPUT DD: .open remote.server.ip.address (EXIT NOOPTMSS

**Usage**

- When starting FTP in a TSO environment that includes support for the REXX programming language, you receive the following message:
CSV003I Requested module IRXSTK not found

This is a normal informational message when starting FTP in a TSO environment.

- If entered from z/OS UNIX, the FTP flags (-d, -e, -g, -i, -n, -p, -t, -v, and -w) must be entered in lowercase. These options can be entered in lowercase or uppercase from TSO.

- GSSAPI authentication is supported only for IPv4 connections. The client will fail the negotiation when the connection is IPv6.
Context

• See "Open subcommand—Connect to the FTP server" on page 217 and "QUIT subcommand—Leave the FTP environment" on page 228 for more information about the OPEN and QUIT subcommands.

• See "FTP EXIT return codes" on page 97 for more information about the EXIT return codes.

• See "Changing local site defaults using FTP.DATA" on page 63 for information about the FTP.DATA data set.

• Refer to the z/OS Communications Server: IP Configuration Reference for information about the TCPIP.DATA data set or loading and customizing DBCS translation tables.
Logging in to FTP

If you correctly specify a foreign host with the FTP command, you are prompted to identify yourself. The following is a sample of the information that is displayed after you successfully invoke the FTP command with foreign_host correctly specified.

```
IBM FTP CS V1R4
FTP: using TCPCS
220 Connection will close if idle for more than 5 minutes.
NAME (9.67.113.37:USER10):
>>> USER USER10
331 Send password please.
PASSWORD:
>>> PASS
230 USER10 is logged on. Working directory is "/tmp".
Command
```

After successfully identifying yourself, you are prompted for a password if the foreign host requires a password. If you enter the password correctly, you are connected to the foreign host.

Notes:
1. You can use the data set NETRC to automatically log in to a remote host. For information about using NETRC, see "NETRC data set" on page 37.
2. If you have enabled UTF-8 encoding of the control connection, the login sequence is different. See "UTF-8 enabled control connection" on page 36 for more information.
3. During the FTP login process, if you receive a series of 230-type replies there may be errors in the user-level configuration files. These replies contain CD and SITE command errors that may appear. These errors describe the failures of CD and SITE commands listed in a user-level configuration file. Contact the system programmer for assistance.

Refer to "z/OS Communications Server: IP and SNA Codes" for additional information about 230-type replies.

For the procedure to enter the FTP environment using the FTP command, see "Establishing and exiting a connection" on page 38 for more information.

Allocating FTP input and output data sets

When you invoke the FTP command from TSO, a check is made to see whether a data set is allocated to INPUT. If a data set is allocated, subcommands are read from that data set rather than from your terminal. Similarly, a check is also made to see whether a data set is allocated to OUTPUT. If so, all FTP prompts and replies are written to that data set rather than to your terminal.

The record length and block size of the output data set can be any size. If the logical record length of the output data set is less than 100 bytes, some messages could be truncated or wrapped around to the next line.

If you create INPUT and OUTPUT data sets, use the following guidelines:

- Specify the INPUT data set:
  - Record format=FB.
  - Logical record length=2080.
The logical record length of the input data set may be any value in the range 80–2080.
- Block size is a multiple of logical record length.
- Specify the OUTPUT data set:
  - Record format=FB.
  - Logical record length=160.
  - Block size is a multiple of 160.

**UTF-8 enabled control connection**

You can specify EXTENSIONS UTF8 in the client's FTP.DATA data set to enable the FTP client to use and accept UTF-8 encoding of the control connection. See Table 13 on page 64 and [z/OS Communications Server: IP Configuration Reference](#) for information about the EXTENSIONS statement. If you code EXTENSIONS UTF8 in the client's FTP.DATA data set, the login sequence is different. Following is an example of logging in to a UTF-8 enabled FTP server when the client has enabled UTF-8 encoding of the control connection.

```
IBM FTP CS V1R4
FTP: using TCPCS
220 Connection will close if idle for more than 5 minutes.
>>> FEAT
211- Extensions supported
  UTF8  LANG en*
211 End
>>> LANG en
200 - Language is en-US (United States English)
NAME (9.67.113.37:USER10):

>>> USER USER10
331 Send password please.
PASSWORD:

>>> PASS
230 USER10 is logged on. Working directory is "/tmp".
Command:
```

The difference is that the client issues the FEAT command during login to negotiate use of UTF-8 on the control connection, as specified in RFC 2640 (see Appendix F, "Related protocol specifications (RFCS)" on page 409). In this example, the FEAT reply indicates the server supports RFC 2640 (UTF8 and LANG keywords), so the client issues LANG to commence UTF-8 encoding of the control connection.

Here is an example of a UTF-8 enabled client logging in to a server which does not support UTF-8 encoding:

```
IBM FTP CS V1R4
FTP: using TCPCS
220 Connection will close if idle for more than 5 minutes.
>>> FEAT
211- Extensions supported
  SIZE
  MDTM
  REST STREAM
211 End
NAME (9.67.113.37:USER10):
```
The client issued the FEAT command during login because EXTENSIONS UTF8 is coded in FTP.DATA. Since the server FEAT reply did not indicate the server supports RFC 2640, no LANG command was issued by the client. The client will not send UTF-8 encoded data to this server.

NETRC data set

The user_id.NETRC data set (/$HOME/.netrc, in z/OS UNIX) provides you with an alternative to specifying your user_id and password as FTP parameters when you want to FTP to a remote host. The following example shows you how to specify the user_id.NETRC data set:

```
MACHINE mvs1.tcp.raleigh.ibm.com LOGIN user28 PASSWORD user28
MACHINE 9.67.112.25 LOGIN user28
```

The keywords MACHINE, LOGIN, and PASSWORD do not have to be uppercase. The variables user_ID and password may be case-sensitive, depending on the remote host. (For example, UNIX or AIX® hosts require user_ID and password to be specified case-sensitive.)

To invoke the user_id.NETRC data set and automatically log on to the remote host named MVS1, enter the FTP command as shown in the following example:

```
User: ftp mvs1
System: IBM FTP CS V1R2
FTP: using TCPCS
220-FTPD1 IBM FTP CS V1R2 at MVSVIC04, 12:00:51 on 2001-04-12.
220 Connection will close if idle for more than 5 minutes.
>>>USER user28
331 Send password please.
>>>PASS ********
230 USER28 is logged on. Working directory is "/u/user28".
Command:
```

In order to bypass definitions in a user_id.NETRC data set for an ftp session, specify the -n parameter. For information about using the user_id.NETRC data set in a batch file, see "Submitting FTP requests in batch" on page 91.

Getting help

The FTP Help subcommands are listed in Table 4.

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Provides an introduction to using FTP.</td>
<td>172</td>
</tr>
<tr>
<td>HELP</td>
<td>Displays help information for FTP.</td>
<td>172</td>
</tr>
</tbody>
</table>
Table 4. FTP subcommands for getting help (continued)

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>man</td>
<td>UNIX Shell command provides help information about the z/OS UNIX ftp client.</td>
<td>7</td>
</tr>
</tbody>
</table>

Establishing and exiting a connection

You normally establish a connection to a foreign host when you invoke the FTP command with a `foreign_host` specified. If you are not successful in specifying a foreign host, or if you need to connect to a different foreign host, use the subcommands listed in Table 5.

Table 5. FTP subcommands for establishing and exiting a connection

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACount</td>
<td>Sends host-dependent account information.</td>
<td>131</td>
</tr>
<tr>
<td>Close</td>
<td>Disconnects from the foreign host.</td>
<td>146</td>
</tr>
<tr>
<td>Open</td>
<td>Opens a connection to a foreign host.</td>
<td>217</td>
</tr>
<tr>
<td>PAss</td>
<td>Supplies a password to the foreign host.</td>
<td>218</td>
</tr>
<tr>
<td>QUIT</td>
<td>Leaves the FTP command environment.</td>
<td>228</td>
</tr>
<tr>
<td>User</td>
<td>Identifies you to a foreign host.</td>
<td>281</td>
</tr>
</tbody>
</table>

Example of establishing and exiting a connection

This example shows how a single FTP session can be used to connect to the following multiple foreign hosts:

1. MVS Host: fec9:c2d4::9:67:115:12 port 21
2. VM Host: 192.9.2.4
Initial Working Directory Considerations at the z/OS FTP Server

When you first log in to a z/OS FTP server, the initial or default working directory at the server is determined by the following:

- The value specified on the STARTDIRECTORY statement in the FTP.DATA file of the server
- The user ID you used when you connected to the server
- The prefix defined in the profile for the user ID

**Note:** To use FTP, your user ID must have an OMVS segment defined (or defaulted).

If STARTDIRECTORY HFS is defined at the server, the initial working directory is the home directory for the user ID. An example of an initial working directory for USER1 is the following:

```
/u/user1
```

If STARTDIRECTORY MVS is defined at the server and no prefix is defined for the user ID, the initial working directory is the user ID followed by a period. An example of an initial working directory for USER1 is the following:

```
USER1.
```
If your TSO user ID is defined through Resource Access Control Facility (RACF®) and a PREFIX is defined for the user ID, the PREFIX value is used as the initial working directory.

The PREFIX of a TSO user ID can be set or changed by using the TSO PROFILE command as follows:

1. Log in to TSO on the MVS system of the FTP server.
2. Set your new prefix using the TSO PROFILE command:
   
   ```
   TSO PROFILE Prefix(prefix)
   ```

   where `prefix` is any TSO prefix that you choose.

**Notes:**

a. You must enter both the opening and closing parentheses.

b. At this point, the TSO prefix is defined for your current TSO session but is not known to RACF or the FTP server until you log off and log on.

3. Log off to save the new default working directory name.

The TSO prefix should now be your default working directory whenever you log on to an FTP session on that FTP server. To make sure you set up the default working directory correctly, use the following steps.

1. Establish an FTP session to the FTP server.
2. Issue a PWD command. This should show the TSO prefix as your new default working directory. For information on using the PWD command, see "PWD subcommand—Display the current working directory" on page 227.

**Notes:**

1. To use the TSO PREFIX as your default working directory, you must have installed RACF Version 1.9 and you must define your TSO user IDs through RACF.

2. When you log in to an FTP server from a z/OS UNIX shell, the default local working directory is the directory from where the ftp client was started.

## Obtaining status and system information

To retrieve and display status information about the local host and remote host, use the subcommands listed in the following table:

### Table 6. FTP subcommands for obtaining status and system information

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Passes a z/OS UNIX System Services command to the local z/OS shell. This command must be issued while using FTP in the z/OS shell.</td>
<td>130</td>
</tr>
<tr>
<td>DEBug</td>
<td>Sets general trace options.</td>
<td>149</td>
</tr>
<tr>
<td>DUMP</td>
<td>Sets extended trace options.</td>
<td>159</td>
</tr>
<tr>
<td>FEATure</td>
<td>Asks server which features or extensions it supports.</td>
<td>164</td>
</tr>
<tr>
<td>LOCSTat</td>
<td>Displays FTP status information for the local host.</td>
<td>201</td>
</tr>
<tr>
<td>NOop</td>
<td>Checks whether the foreign host is still responding.</td>
<td>216</td>
</tr>
<tr>
<td>STAtus</td>
<td>Displays status information for the foreign host.</td>
<td>267</td>
</tr>
<tr>
<td>SYSystem</td>
<td>Displays the name of the foreign host's operating system.</td>
<td>272</td>
</tr>
</tbody>
</table>
Table 6. FTP subcommands for obtaining status and system information (continued)

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO</td>
<td>Passes a TSO command to the local host TSO</td>
<td>275</td>
</tr>
</tbody>
</table>

Working with directories on the remote host

To obtain directory information on the remote host, use the subcommands listed in Table 7.

Table 7. FTP subcommands for working with directories on the remote host

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>Changes the working directory.</td>
<td>140</td>
</tr>
<tr>
<td>CDUP</td>
<td>Changes to the parent of the current working directory.</td>
<td>143</td>
</tr>
<tr>
<td>CWd</td>
<td>Changes the working directory (synonymous with CD).</td>
<td>140</td>
</tr>
<tr>
<td>Dir</td>
<td>Lists the directory entries for files on the foreign host.</td>
<td>155</td>
</tr>
<tr>
<td>LS</td>
<td>Lists the names of files on the foreign host.</td>
<td>203</td>
</tr>
<tr>
<td>MKdir</td>
<td>Creates a directory on the foreign host.</td>
<td>211</td>
</tr>
<tr>
<td>PWd</td>
<td>Displays the name of the active working directory on the foreign host.</td>
<td>227</td>
</tr>
<tr>
<td>RMDir</td>
<td>Removes a directory on the foreign host.</td>
<td>234</td>
</tr>
</tbody>
</table>

Examples of the CD subcommand

This example shows how to change and choose remote working directories and how the z/OS FTP server enables you to switch between the MVS and Hierarchical File System (HFS) environments. For more information on how to change the directory levels, see the information on the CD, CDUP, and LCD subcommands later in this chapter.
Examples showing the differences between DIR and LS output for HFS directories

The examples in this section use the following Internet addresses:

MVSXA2: 9.67.113.25
MVSXA3: 9.67.113.24

The current host is MVSXA2 (9.67.113.25). An FTP command is issued from 9.67.113.25 to 9.67.113.24.
DIR provides detailed information about the data sets under the remote working directory, while LS shows the data set names only.

Examples showing the differences between DIR and LS output with DIRECTORYMode and DATASETmode for MVS

This section gives examples of issuing a DIR and LS command in both DIRECTORYMode and DATASETmode.

DIR provides detailed information about the data sets under the remote working directory, while LS shows the data set names only.
User: cd 'tcpv3'

System:
>>>CWD 'tcpv3'
257 "TCPV3." is working directory name prefix.

Command:

User: site directorymode

System:
>>>SITE directorymode
200 Site command was accepted

Command:

User: dir

System:
>>>PORT 1,1,2,2,4,39
200 Port request OK.
>>>LIST 125 List started OK.

Volume Unit
Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
Pseudo Directory ETC
Pseudo Directory FTP
Pseudo Directory HOSTS
Pseudo Directory NSMAIN
Pseudo Directory PROFILE
Pseudo Directory STANDARD
Pseudo Directory TCPIP
Pseudo Directory TCPIPL62
Pseudo Directory TELNET
250 List completed successfully.

Command:

User: site datasetmode

System:
>>>SITE datasetmode
200 Site command was accepted

Command:

User: dir

System:
>>>PORT 1,1,2,2,4,40
200 Port request OK.
>>>LIST
125 List started OK.

Volume Unit
Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
APCSPL 3380D 07/16/97 1 1 FB 80 8800 PS ETC.RPC
APCSPL 3380D 08/03/97 1 1 FB 80 3200 PS ETC.SERVICES
APCSPL 3380D 08/03/97 1 1 FB 80 3120 PS FTP.DATA
APCSPL 3380D 08/02/97 1 1 F 158 158 PS HOSTS.ADDRINFO
APCSPL 3380D 08/03/97 1 1 FB 80 3120 PS HOSTS.LOCAL
APCSPL 3380D 07/30/97 1 1 F 56 56 PS HOSTS.SITEINFO
APCSPL 3380D 07/15/97 1 1 FB 80 8800 PS NSMAIN.CACHE
APCSPL 3380D 07/28/97 1 1 FB 80 8800 PS NSMAIN.DATA
APCSPL 3380D 08/03/97 1 2 FB 80 3200 PS PROFILE.TCPIP
APCSPL 3380D 07/26/97 1 2 FB 80 3200 PS PROFILE.TCPIPL62
APCSPL 3380D 08/03/97 1 1 VB 5124 6160 PS STANDARD.TCPKJBIN
APCSPL 3380D 08/03/97 1 15 F 256 256 PS STANDARD.TCPIXBIN
APCSPL 3380D 08/03/97 1 1 FB 80 3120 PS TCPIP.DATA
APCSPL 3380D 06/29/97 1 2 FB 80 3200 PS TCPIPL62.CONFIG
APCSPL 3380D 07/29/97 1 15 F 256 256 PS TELNET.TCPXLBIN
250 List completed successfully.
Working with directories on the local host

To work with directories on the local host, use the subcommands listed in Table 8.

Table 8. FTP subcommands for working with directories on the local host

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCd</td>
<td>Changes the current directory on the local host.</td>
<td>179</td>
</tr>
<tr>
<td>Lmkdir</td>
<td>Creates a directory on the local host.</td>
<td>181</td>
</tr>
<tr>
<td>LPwd</td>
<td>Displays the name of the active working directory on the local host.</td>
<td>202</td>
</tr>
</tbody>
</table>
Your default working directory on the local host is set according to the environment in which the FTP client is invoked: $HOME in z/OS UNIX, your MVS user ID in TSO.

The following examples show how to choose local working directories.

```
User: ftp 1.1.2.3
System: IBM FTP CS V1R2
       FTP: using TCPCS
       Connecting to 1.1.2.3, port 21
       220-FTPD1 IBM FTP CS V1R2 at MVS164, 20:12:38 on 1997-12-02.
       220 Connection will close if idle for more than 5 minutes.
       NAME (<host>:tsouserid):
User: mvsuser
System: >>>USER mvsuser
       331 Send password please.
       Password:
       >>>PASS ********
       230 MVSUSER is logged on. Working directory is "/u/mvsuser".

Command:
```

```
User: lpwd
System: Local directory is MVSUSER.
       Command:
User: lcd tcpip
System: Local directory name set to MVSUSER.TCPIP.
       Command:
User: lpwd
System: Local directory is MVSUSER.TCPIP.
       Command:
User: lcd 'ftp.test'
System: Local directory name set to FTP.TEST.
       Command:
User: lpwd
System: Local directory is FTP.TEST.
       Command:
User: lcd ..
System: Local directory name set to FTP.
       Command:
User: lpwd
System: Local directory is FTP.
       Command:
```

Security considerations

The following describes security issues to consider when using FTP.

Using security mechanisms

File data transferred between an FTP client and server can be secured with respect to encryption, authentication, and data integrity.
Authentication established using a security mechanism may also be used to make the authorization decision. The FTP security interaction begins with a client telling the server what security mechanism it wants to use with the AUTH command. The server either accepts this mechanism, rejects this mechanism, or, in the case of a server which does not implement the security extensions, rejects the command completely. The server’s reply indicates if the client must respond with additional data for the security mechanism to interpret.

Once a security association is established, authentication (which is part of this association) may be used in addition to the standard user ID/password exchange for authorizing a user to connect to the server. A user ID specified by the USER command is always required to specify the identity to be used on the server.

Transport Layer Security (TLS) is an upwardly-compatible successor to Secure Sockets Layer (SSL). SSL is a protocol that performs secure and encrypted TCP transmission. The FTP client supports either SSL or TLS protected sessions, including client authentication. Note that the negotiation of SSL versus TLS is performed by the sockets-layer TLS code and is transparent to FTP.

Many TLS/SSL applications work by having a client connect to one TCP port for unprotected sessions and a separate TCP port for protected ones. FTP supports this mode for compatibility with the original SSL design. However, FTP also provides a more general solution for FTP security, where the client connects to the FTP server on the regular, non-encrypted port and negotiates authentication and encryption options.

The FTP client assumes that port 990 is a protected port. An AUTH command is not needed and the client completes the exchange of additional data with the server immediately after a successful connection.

There are optional FTP commands and statements for negotiating session security. The following is a list of the configuration parameters that determine whether the client uses a security mechanism to protect the session:

- See “FTP command—Enter the FTP environment” on page 30 for a description of the start parameters.
- See “Changing local site defaults using FTP.DATA” on page 63 and the z/OS Communications Server: IP Configuration Reference for a description of the FTP.DATA statements.

**Using a SOCKS server**

You can configure the FTP client to access FTP servers through a SOCKS server. When entering FTP through a SOCKS server, the FTP client establishes a connection with the SOCKS server and then the SOCKS server establishes a
connection to the FTP server. All data and commands for the FTP session are relayed through the SOCKS server. A SOCKS server can be configured to log all FTP connections and permit or deny access to certain FTP servers. See "Configuring the FTP client for SOCKS server" on page 76 for more information.
Chapter 4. Transferring data using the File Transfer Protocol (FTP)

The FTP command enables you to transfer data sets between your local host and any host that supports TCP/IP and FTP. Using the FTP command and its subcommands, you can sequentially access multiple hosts without leaving the FTP environment.

This chapter describes information about:

- “Preparing the environment”
- “Transferring data” on page 50
- “Changing local site defaults using FTP.DATA” on page 63
- “Sample FTP.DATA data set (FTCDATA)” on page 77
- “Using different SBCS translation tables for the control and data connections” on page 85
- “Using different MBCS translation tables for the data connection” on page 85
- “Specifying values for new data sets” on page 86
- “Generation data group support” on page 88
- “Submitting FTP requests in batch” on page 91
- “Using the EXEC interface” on page 94
- “FTP EXIT return codes” on page 97
- “Restarting a failed data transfer” on page 101
- “Interfacing with JES” on page 101
- “Performing DB2 SQL queries with FTP” on page 117
- “FTP with DBCS support” on page 123

Preparing the environment

You can use the subcommands listed in Table 9 to prepare the environment before working with data.

Table 9. FTP subcommands for preparing the environment

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AScii</td>
<td>Sets the transfer type to ASCII.</td>
<td>134</td>
</tr>
<tr>
<td>BINary</td>
<td>Sets the transfer type to IMAGE.</td>
<td>137</td>
</tr>
<tr>
<td>BLock</td>
<td>Sets the data transfer mode to block mode.</td>
<td>138</td>
</tr>
<tr>
<td>COMpress</td>
<td>Sets the data transfer mode to compressed mode.</td>
<td>147</td>
</tr>
<tr>
<td>EBcdic</td>
<td>Sets the transfer type to EBCDIC.</td>
<td>161</td>
</tr>
<tr>
<td>File</td>
<td>Sets the file structure to file.</td>
<td>165</td>
</tr>
<tr>
<td>GLob</td>
<td>Toggles globbing (the expansion of metacharacters in file names) for the MDELETE, MGET, and MPUT subcommands.</td>
<td>168</td>
</tr>
<tr>
<td>LOCSite</td>
<td>Specifies information that is used by the local host to provide service specific to that host system.</td>
<td>183</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>Sets language for server replies.</td>
<td>178</td>
</tr>
<tr>
<td>MOde</td>
<td>Specifies the mode or data format of the transfer.</td>
<td>213</td>
</tr>
</tbody>
</table>
Table 9. FTP subcommands for preparing the environment (continued)

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROMpt</td>
<td>Toggles interactive prompting for MDELETE, MGET, and MPUT commands.</td>
<td>220</td>
</tr>
<tr>
<td>QUOte</td>
<td>Sends an uninterpreted string of data.</td>
<td>229</td>
</tr>
<tr>
<td>REcord</td>
<td>Sets the file structure to record.</td>
<td>231</td>
</tr>
<tr>
<td>SENDPort</td>
<td>Enables or disables automatic transmission of the FTP server PORT command.</td>
<td>238</td>
</tr>
<tr>
<td>SENDSite</td>
<td>Enables or disables automatic transmission of the SITE subcommand.</td>
<td>239</td>
</tr>
<tr>
<td>Site</td>
<td>Sends information to the foreign host using site-specific commands.</td>
<td>240</td>
</tr>
<tr>
<td>STREam</td>
<td>Sets the data transfer mode to stream mode.</td>
<td>269</td>
</tr>
<tr>
<td>STRucture</td>
<td>Sets the file transfer structure.</td>
<td>270</td>
</tr>
<tr>
<td>SUNique</td>
<td>Changes the storage methods.</td>
<td>271</td>
</tr>
<tr>
<td>TType</td>
<td>Specifies the transfer type.</td>
<td>276</td>
</tr>
</tbody>
</table>

Transferring data

You can use the subcommands listed in Table 10 to work with and transfer data.

Table 10. FTP subcommands for transferring data

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APpend</td>
<td>Appends a data set on your local host to a file on the foreign host.</td>
<td>132</td>
</tr>
<tr>
<td>DELEte</td>
<td>Deletes a single file on the foreign host.</td>
<td>153</td>
</tr>
<tr>
<td>DELimit</td>
<td>Displays the delimiter character between the file_name and file_type.</td>
<td>154</td>
</tr>
<tr>
<td>Get</td>
<td>Copies a file from the foreign host to your local host.</td>
<td>166</td>
</tr>
<tr>
<td>MDelete</td>
<td>Deletes multiple files on the foreign host.</td>
<td>205</td>
</tr>
<tr>
<td>MGet</td>
<td>Copies multiple files from the foreign host to your local host.</td>
<td>207</td>
</tr>
<tr>
<td>MPut</td>
<td>Copies multiple files on your local host to the foreign host.</td>
<td>214</td>
</tr>
<tr>
<td>PUT</td>
<td>Copies a file on your local host to the foreign host.</td>
<td>225</td>
</tr>
<tr>
<td>REName</td>
<td>Renames a file on the foreign host.</td>
<td>232</td>
</tr>
<tr>
<td>REStart</td>
<td>Restarts a checkpointed data transfer.</td>
<td>233</td>
</tr>
<tr>
<td>SRESTART</td>
<td>Restarts a stream mode data transfer.</td>
<td>265</td>
</tr>
</tbody>
</table>

How to transfer data

FTP supports only the data transfer of a data set or file structured as a continuous sequence of data bytes. This ensures that the correct record format is preserved across MVS hosts. Information could be lost or altered during transmission if you use an incorrect transfer.

Table 11 shows how to set the transmission attributes for different host systems. For example, VM or MVS host systems use EBCDIC for internal character representation. A text file of ASCII data type contains displayable characters; a carriage return (X'0D') and line feed (X'0A') are used to delimit a line. A text file of
EBCDIC data type contains displayable characters; the new-line character (X'15') is used to delimit a line. A binary file contains a contiguous stream of bits with no line delimiters.

Table 11. Recommended methods for data transfer

<table>
<thead>
<tr>
<th>Transfer between host types</th>
<th>Data transfer type</th>
<th>Data transfer mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBCDIC and EBCDIC — DBCS text data</td>
<td>IBMKANJI (EBCDIC)</td>
<td>Stream</td>
</tr>
<tr>
<td>EBCDIC and EBCDIC — text data</td>
<td>EBCDIC</td>
<td>Stream</td>
</tr>
<tr>
<td>EBCDIC and EBCDIC — DBCS binary data</td>
<td>IBMKANJI (EBCDIC)</td>
<td>Block</td>
</tr>
<tr>
<td>EBCDIC and EBCDIC — binary data</td>
<td>EBCDIC</td>
<td>Block</td>
</tr>
<tr>
<td>EBCDIC and ASCII — DBCS text data</td>
<td>SJISKANJI,</td>
<td>Stream</td>
</tr>
<tr>
<td></td>
<td>EUCKANJI,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIS78KJ,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIS83KJ,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HANGEUL,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KSC5601,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCHINESE,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BIG5,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCHINESE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(ASCII)</td>
<td></td>
</tr>
<tr>
<td>ASCII and EBCDIC — MBCS data</td>
<td>ASCII</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII and EBCDIC — text data</td>
<td>ASCII</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII and EBCDIC — DBCS binary data</td>
<td>Image (binary)</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII and EBCDIC — binary data</td>
<td>Image (binary)</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII-to-EBCDIC-to-ASCII — all data</td>
<td>Image (binary)</td>
<td>Stream</td>
</tr>
</tbody>
</table>

**Note:** The EBCDIC host is used for storage only. Data remains encoded in ASCII; therefore, the data cannot be used on the EBCDIC host.

For more information about the DBCS data type keywords and examples, see “FTP with DBCS support” on page 123.

For information about setting data transfer type, see “Type subcommand—Set the data transfer type” on page 276. For information about setting data transfer mode, see “MOde subcommand—Set the data transfer mode” on page 213.

**Examples of GET and MGET subcommands**

GET and MGET enable you to obtain files from a remote host and send them to the local host. In this example, FTP subcommands are issued from MVSXA2 to MVSVIC03. See Table 10 on page 50 for other subcommands useful for working with and transferring data.

The following members exist in the data set USER121.FTP.EXAMPLE on MVSVIC03:

- FILE1
- FILE2
- FILE3
- FILE4
- FILE5
The following is displayed when entering the FTP environment:

User: ftp 9.67.113.24 621
System:
  IBM FTP CS V1R2
  FTP: using TCPCS
  Connecting to 9.67.113.24, port 621
  220-FTPD1 IBM FTP CS V1R2 at MVS164, 20:12:38 on 1997-12-02.
  220 Connection will not timeout.
  NAME <host>:tsouserid):
User: user121
System:
  >>>USER user121
  331 Send password please.
  Password:
  >>>PASS ********
  230 USER121 is logged on. Working directory is "/u/user121".
  Command:
User: get 'user121.ftp.example(file1)' 'user121.ftp.example(file1)'
System:
  'USER121.FTP.EXAMPLE(FILE1)' IS AN non-EXISTENT PARTITIONED DATASET.
  USE LMKDIR TO CREATE IT. LOCAL FILE NOT FOUND
  COMMAND:
User: lmkdir 'user121.ftp.example'
System:
  USER121.FTP.EXAMPLE CREATED.
  COMMAND:
User: get 'user121.ftp.example(file1)' 'user121.ftp.example(file1)'
System:
  >>>PORT 9,67,112,25,4,9
  200 Port request OK.
  >>>RETR 'USER121.ftp.example(file1)'
  125 Sending data set USER121.FTP.EXAMPLE(FILE1) FIXrecfm 128
  250 Transfer completed successfully.
  3464 bytes transferred in 0.754 seconds. Transfer rate 4.59 Kbytes/sec.
  Command:
User: get 'user121.ftp.example(file2)' 'user121.ftp.example(file2)'
System:
  >>>PORT 9,67,112,25,4,34
  200 Port request OK.
  >>>RETR 'USER121.ftp.example(file2)'
  125 Sending data set USER121.FTP.EXAMPLE(FILE2) FIXrecfm 128
  250 Transfer completed successfully.
  3464 bytes transferred in 1.483 seconds. Transfer rate 2.34 Kbytes/sec.
  Command:
User: get 'user121.ftp.example(file2)' 'user121.ftp.example(file2)'
System: Data set 'USER121.FTP.EXAMPLE(FILE2)' was not replaced.
Local file already exists
To replace it, use command with the (REPLACE option
Command:
User: get 'user121.ftp.example(file2)' 'user121.ftp.example(file2)' (replace
System: >>>PORT 9,67,112,25,4,35
200 Port request OK.
>>>RETR 'user121.ftp.example(file2)'
125 Sending data set USER121.FTP.EXAMPLE(FILE2)
250 Transfer completed successfully.
3464 bytes transferred in 0.767 seconds. Transfer rate 0.50 Kbytes/sec.
Command:
User: lpwd
System: Local directory is USER121
COMMAND:
User: mget 'user121.ftp.example(file3)' 'user121.ftp.example(file4)'
System: >>>PORT 9,67,112,25,4,10
200 Port request OK.
>>>NLST 'user121.ftp.example(file3)'
125 List started OK.
250 List completed successfully.
>>>PORT 9,67,112,25,4,11
200 Port request OK.
>>>NLST 'user121.ftp.example(file4)'
125 List started OK.
250 List completed successfully.
>>>PORT 9,67,112,25,4,12
200 Port request OK.
>>>RETR 'USER121.FTP.EXAMPLE(FILE3)'
125 Sending data set USER121.FTP.EXAMPLE(FILE3)
250 Transfer completed successfully.
3993 bytes transferred in 0.745 seconds. Transfer rate 0.51 Kbytes/sec.
>>>PORT 9,67,112,25,4,13
200 Port request OK.
>>>RETR 'USER121.FTP.EXAMPLE(FILE4)'
125 Sending data set USER121.FTP.EXAMPLE(FILE4)
250 Transfer completed successfully.
7367 bytes transferred in 0.818 seconds. Transfer rate 9.01 Kbytes/sec.
Command:
User: pwd

System: Local directory is USER121.

Command:

User: cd 'user121.ftp.example'

System: >>>CWD 'user121.ftp.example'
250 "USER121.FTP.EXAMPLE" partitioned data set is working directory.

Command:

User: pwd

System: >>>PWD
257 "USER121.FTP.EXAMPLE" partitioned data set is working directory.

Command:

User: mget file3 file4

System: >>>PORT 9,67,112,25,4,20
200 Port request OK.
>>>NLST file3
125 List started OK.
250 List completed successfully.
>>>PORT 9,67,112,25,4,21
200 Port request OK.
>>>NLST file4
125 List started OK.
250 List completed successfully.
>>>PORT 9,67,112,25,4,22
200 Port request OK.
>>>RETR FILE3
125 Sending data set USER121.FTP.EXAMPLE(FILE3)
250 Transfer completed successfully.
3993 bytes transferred in 0.549 seconds. Transfer rate 0.46 Kbytes/sec.
>>>PORT 9,67,112,25,4,23
200 Port request OK.
>>>RETR FILE4
125 Sending data set USER121.FTP.EXAMPLE(FILE4)
250 Transfer completed successfully.
7367 bytes transferred in 0.936 seconds. Transfer rate 0.23 Kbytes/sec.

Command:

User: quit

System: >>>QUIT
221 Quit command received. Goodbye.
READY

User: ftp 9.67.113.24 621

System: IBM FTP CS/390 V2R10 1997 314 01:11 UTC
Connecting to 9.67.113.24, port 621
220-FTPD1 IBM FTP CS/390 V2R10 AT MVS164, 20:12:38 ON 1997-12-02.
220 Connection will not timeout.
NAME (<host>:touserid):

User: user121

System: >>>USER user121
331 Send password please.
Password:
>>>PASS ********
230 USER121 is logged on. Working directory is "/u/user121".

Command:
User: get '/u/user121/ftp.example/file1' 'user121.ftp.example(file1)'

System:
>>>PORT 9,67,112,25,4,24
200 Port request OK.
>>>RETR '/u/user121/ftp.example/file1'
125 Sending data set /u/user121/ftp.example/file1
250 Transfer completed successfully.
3464 bytes transferred in 1.391 seconds. Transfer rate 2.49 Kbytes/sec.

Command:
User: lcd 'user121.ftp.example'
System: Local directory name set to partitioned data set USER121.FTP.EXAMPLE.

Command:
User: lpwd
System: Local directory is partitioned data set USER121.FTP.EXAMPLE.

Command:
User: cd '/u/user121/ftp.example'
System: >>>CWD '/u/user121.ftp.example'
250 HFS directory /u/user121/ftp.example is the current working directory

Command:
User: pwd
System: >>>PWD
257 "/u/user121.ftp.example" is the HFS working directory.

Command:
User: get file1
System: >>>PORT 9,67,112,25,4,26
200 Port request OK.
>>>RETR file1
125 Sending data set /u/user121/ftp.example/file1
250 Transfer completed successfully.
3464 bytes transferred in 1.059 seconds. Transfer rate 3.27 kbytes/sec.

Command:
User: mget '/u/user121/ftp.example/file4' '/u/user121/ftp.example/file5'
System:
>>>PORT 9,67,112,25,4,33
200 Port request OK.
>>>NLST '/u/user121/ftp.example/file4'
125 List started OK
250 List completed successfully.
>>>PORT 9,67,112,25,4,34
200 Port request OK.
>>>NLST '/u/user121/ftp.example/file5'
125 List started OK
250 List completed successfully.
>>>PORT 9,67,112,25,4,35
200 Port request OK.
>>>RETR /u/user121/ftp.example/file4
125 Sending data set /u/user121/ftp.example/file4
250 Transfer completed successfully.
7367 bytes transferred in 1.324 seconds. Transfer rate 5.56 kbytes/sec.
200 Port request OK.
>>>RETR /u/user121/ftp.example/file5
125 Sending data set /u/user121/ftp.example/file5
250 Transfer completed successfully.
3464 bytes transferred in 0.951 seconds. Transfer rate 3.64 kbytes/sec.

Command:
The data set USER121.FTP.EXAMPLE on MVSXA2 now contains the following members:
FILE1
FILE2
FILE3
FILE4
Notes:
1. You do not have a choice of names for the local file as a result of the MGET command.
2. The MGET command is not applicable for generation data groups (GDGs).

Examples of PUT and MPUT subcommands

PUT and MPUT enable you to send files from a local host to a remote host. In this example, FTP subcommands are issued from MVSXA2 to MVSVIC03. The data set USER121.FTP.EXAMPLE on MVSXA2 contains the following members:

APPEND01
XA2FILE1
XA2FILE2
XA2FILE3

The data set USER121.FTP.EXAMPLE on MVSVIC03 contains the following members:

XA3FILE1
XA3FILE2
XA3FILE3

The following is displayed when entering the FTP environment:

User: ftp 1.1.2.3
System: IBM FTP CS V1R2
FTP: using TCPCS
Connecting to 1.1.2.3, port 21
220-FTPD1 IBM FTP CS V1R2 at MVS164, 20:12:38 on 1997-12-02.
220 Connection will close if idle for more than 5 minutes.
NAME (<host>:tsouserid):
User: user121
System: >>>USER user121
331 Send password please.
Password:
>>>PASS ********
230 user121 is logged on. Working directory is "/u/user121"

Command:

User: put 'user121.ftp.example(xa2file1)' 'user121.ftp.example(f1from2)'
System: >>>SITE FIXrecfm 128 Lrec=128 Recfm=FB BlockSize=6144
200 Site command was accepted
>>>PORT 1,1,2,4,4,48
200 Port request OK.
>>>STOR 'user121.ftp.example(f1from2)'
125 Storing data set USER121.FTP.EXAMPLE(F1FROM2)
250 Transfer completed successfully.
Transfer rate 0.35 Kbytes/sec.
Command:

User: put 'user121.ftp.example(xa2file1)' 'user121.ftp.example(f1from2)'
System: >>>SITE FIXrecfm 128 Lrec=128 Recfm=FB BlockSize=6144
200 Site command was accepted
>>>PORT 1,1,2,4,4,49
200 Port request OK.
>>>STOR 'user121.ftp.example(f1from2)'
125 Storing data set USER121.FTP.EXAMPLE(F1FROM2)
250 Transfer completed successfully.
390 bytes transferred in 0.680 seconds.
Transfer rate 0.57 Kbytes/sec.
The data set USER121.FTP.EXAMPLE on MVSVIC03 now contains the following members:

- F1FROM2
- F1FROM21
- XA3FILE1
- XA3FILE2
- XA3FILE3

MVSVIC03 now also has the following data sets:

USER121.FTP.EXAMPLE.XA2FILE2
**Note:** The MPUT command is not applicable for generation data groups (GDGs).

### DD name support

The //DD: token prefixed before a 1–8 character local file name on a client file access command indicates that the token which follows is actually a DDNAME, rather than a local file name. This DDNAME must be allocated by the user (for example, in the JCL that started the FTP client). The server file name must be explicitly specified when a DDNAME is being used to access a local file for a put command.

Sometimes the client requires DCB information before it opens a data set. Among the situations where this is true are:

- Reading and writing spanned records (RECFM=VS or VBS)
- Reading and writing records that contain ASA control characters
- Reading and writing variable-length records while preserving the RDW
- Reading and writing fixed-length records while preserving trailing blanks

When a data set is allocated via a DDname and the DCB information is needed before open, the FTP client must be able to find the DCB information on the DD statement that was used to allocate the data set.

DCB attributes for a DDname allocation are acquired using the attributes or data set name specified in the DD statement's DCB parameter. Refer to the [z/OS MVS JCL Reference](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.7/s.gif?/info/zos/mvs/jcl) for restrictions on using backward references in the DCB parameter.

If the DD refers to a cataloged DASD data set, any DCB attributes not specified will be retrieved from the DSCB. DCB attributes on the DD statement will override those found in the DSCB, except that LRECL=0 and BLKSIZE=0 will not override a different value in the DSCB.

If the DD refers to a tape data set that is to be opened for input (PUT //DD:) then the record format specified on the DD statement will be used instead of the READTAPEFormat setting. If no record format (RECFM) is specified on the DD statement, the READTAPEFormat setting (if any) will be used.

Once the data set is opened by FTP, its attributes are set using the data returned in the DCB by open.

Following is a sample job that shows usage of the //DD: token. In the sample job there are two data sets that use the local file specification with the //DD: token. One is a data set that is created as a new GDG data set in STEP01 (see the OUTSET DD statement). Note that STEP02 (the FTP step) uses a backward reference with the DD02 DD statement to locate the data set. Since the referenced DD statement contains explicit DCB attributes, FTP can access the attributes prior to opening the data set. The second data set is an old data set that existed before the job was executed.

```plaintext
//USER33J JOB MSGLEVEL=1,MSGCLASS=H,USER=USER33,PASSWORD=**pw**
//STEP01 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//OUTSET DD DSNAME=USER33.MYGDG(+1),DISP=(NEW,CATLG,CATLG),
//VOLUME=SER=CPDLB1,SPACE=(TRK,(5,5)),UNIT=SYSDA,
```
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSIN DD *
< create statements >
//STEP02 EXEC PGM=FTP,REGION=2048K,PARM='(TCP TCPCS TRACE'
//STEPLIB DD DSN=USER33.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*  
//DD01 DD DSNAME=USER33.TEST.S.A,DISP=OLD
//DD02 DD DSNAME=*.STEP01.OUTSET,DISP=SHR
//OUTPUT DD SYSOUT=*  
//INPUT DD *
9.67.113.57 6321
USER33 **pw**
/put //DD:DD02 data
get data //DD:DD01
quit
/*
Following are short descriptions of the numbered items in the example.

1. DD statement that allocates a new generation of a GDG data set
2. DD statement for an existing data set
3. Backward reference for the new data set in STEP01
4. Put subcommand using the //DD: token for the new data set created in STEP01
5. Get subcommand using the //DD: token for the existing data set

The FTP output for the above job is the following. Note that only a select few of the FTP trace statements are shown.

EZA1736I FTP (TCP TCPCS
EZA1450I IBM FTP CS VIR2 2000 090 19:22 UTC
EZA1466I FTP: using TCPCS
EZA1456I Connect to ?
EZA1736I 9.67.113.57 6321
EZA1554I Connecting to: 9.67.113.57 port: 6321.
220-FTPDJG1 IBM FTP CS VIR2 at MVS164, 14:58:36 on 2000-04-01.
220 Connection will not timeout.
EZA1459I NAME (9.67.113.57:USER33):
EZA1701I >>> USER USER33
331 Send password please.
EZA1701I >>> PASS
230 USER33 is logged on. Working directory is "/u/user33".
EZA1460I Command:
1. EZA1736I put //DD:DD02 data
2. MF0680 seq_open_file: DName DD02 has filename USER33.MYGDG.G0087V00
EZA1701I >>> PORT 9,67,113,57,6,158
200 Port request OK.
EZA1701I >>> STOR data
125 Storing data set /u/user33/data
250 Transfer completed successfully.
EZA1617I 820 bytes transferred in 0.020 seconds. Transfer rate 41.00 Kbytes/sec.
EZA1460I Command:
3. EZA1736I get data //DD:DD01
4. MF0680 seq_open_file: DName DD01 has filename USER33.TEST.S.A
EZA1701I >>> PORT 9,67,113,57,6,159
200 Port request OK.
EZA1701I >>> RETR data
125 Sending data set /u/user33/data
250 Transfer completed successfully.
EZA1617I 820 bytes transferred in 0.030 seconds. Transfer rate 27.33 Kbytes/sec.
EZA1460I Command:
Put subcommand using \( /DD: \) token for the local file

Trace statement showing that the local data set name is the new GDG data set created in STEP01

Get subcommand using \( /DD: \) token for the local file

Trace statement showing that the local data set name is the existing data set.

Load Module Transfer (LMTR)

As long as your FTP client and FTP server are both at the IBM Communications Server for OS/390 V2R10 level or higher, you can use FTP to transfer MVS load modules between load libraries on different hosts or the same host. MVS load modules transferred, using IBM Communications Server for OS/390 V2R10 level or higher of support, will be executable on the target system. A load module may be specified by its real name or by one of its alias names, and in either case all aliases will be transferred with each load module. Load module transfer is also supported for proxy transfer, in which case all three hosts (client, primary server, and secondary server) must be IBM Communications Server for OS/390 V2R10 or higher.

Load module transfer processing (at z/OS V1R2 Communications Server level or higher) makes use of the IEBCOPY system utility, which must be available on both the origin and destination hosts.

The following FTP file transfer commands will properly transfer MVS load modules:

- get
- mget
- put
- mput

Because of the special requirements of MVS load modules, there are some additional restrictions:

- The current working directory on both the client and the server must be the source or destination load library. A load library is a PDS or PDSE with RECFM=U.
- Only member names may be specified. No fully qualified names may be specified.
- File rename is not supported on load module transfer.
- Load modules may only be transferred between the same types of libraries. For example, PDS to PDSE transfer is not allowed.
- If load modules are being sent to or from the z/OS FTP client, the client must be started from one of the following environments:
  - TSO terminal session
  - TSO REXX
  - TSO batch
  - TSO background
  - Unix System services terminal session
A load module loading from a temporary data set will always be a REPLACE operation overwriting existing members. LMTR will not be performed in STOU mode (the user has toggled SUNIQUE on).

There is no prompting on mput and mget commands. All files that match the mask provided will be transferred.

In most cases where load module processing cannot be performed, including failure to abide by the restrictions given above, FTP will complete the file transfer using normal processing. Any load modules transferred with normal processing will not be executable on the target system.

For the examples shown below, the following assumptions are made:

- The contents of load library USER.LINKLIB are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Alias-of</th>
<th>Size</th>
<th>TTR</th>
<th>AC</th>
<th>AM</th>
<th>RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZACDOPN</td>
<td>0000D268</td>
<td>00160F</td>
<td>01</td>
<td>31</td>
<td>ANY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZAFTPLC</td>
<td>000E3758</td>
<td>00001B</td>
<td>01</td>
<td>31</td>
<td>ANY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTP</td>
<td>EZAFTPLC</td>
<td>000E3758</td>
<td>00001B</td>
<td>01</td>
<td>31</td>
<td>ANY</td>
<td></td>
</tr>
<tr>
<td>OPING</td>
<td>EZACDOPN</td>
<td>0000D268</td>
<td>00160F</td>
<td>01</td>
<td>31</td>
<td>ANY</td>
<td></td>
</tr>
</tbody>
</table>

- USER1.TESTLIB is a PDS with RECFM=U.

The following example is a sample session involving a load module transfer with debug/trace on. (For clarity, user input is shown offset to the left and notes are contained within // characters.)

```
220 Connection will not timeout.
NAME (9.67.43.61:USER1):
user1
>>> USER user1
331 Send password please.
PASSWORD:

>>> PASS
230 USER1 is logged on. Working directory is "USER1.".
Command:
cd 'user.linklib'
>>> CWD 'user.linklib'
250-The working directory may be a load library //1//
250-The working directory "USER.LINKLIB" is a partitioned data set //1//
Command:
lcd 'user1.testlib'
Local directory might be a load library //1//
Local directory name set to partitioned data set USER1.TESTLIB
Command:
get oping
>>> XLMT PDS 0 oping //2//
250 PDS 53864 - send next command for load module transfer //2//
>>> PORT 9,67,43,65,4,41
200 Port request OK.
>>> RETR oping
125-Transferring load module //2//
125 DCR 32760 32760 //2//
```
In this next example, the user attempts to transfer a load module and rename the load module. As this is not supported, load module processing will exit and normal processing will take over. The transferred load module will not be executable on the target system.

```
cd 'user.linklib'
>>> CWD 'user.linklib'
250-The working directory may be a load library
250-The working directory "USER.LINKLIB" is a partitioned data set
Command:
lcd 'user1.testlib'
Local directory might be a load library
Local directory name set to partitioned data set USER1.TESTLIB
Command:
get oping ping
Load module transfer does not support load module rename //5//
>>> PORT 9,67,43,66,4,41
200 Port request OK.
>>> RETR oping
125 Sending data set USER.LINKLIB(OPING)
250 Transfer completed successfully.
61984 bytes transferred in 0.190 seconds. Transfer rate 326.23 Kbytes/sec.
Command:
```
2. There are additional flows between the client and server for load module transfer.

3. The IEBCOPY system utility is invoked by FTP as part of load module transfer processing. Note that this output is only seen when running with debug or trace on. Otherwise the IEBCOPY output is placed in the syslog.

4. The actual load module and all its aliases will be transferred, even though (in this case only) the alias was specified by the user.

5. When load module transfer processing cannot be performed, the user is warned and the transfer may be completed using normal processing. The data set 'USER.TESTLIB(PING)' will not be executable on the client system.

### Changing local site defaults using FTP.DATA

The default values for the local site parameters are hardcoded in the FTP client module. You can change these default values by creating an FTP.DATA configuration data set.

**Note:** Unless otherwise indicated, FTP.DATA data set refers to both the /etc/ftp.data HFS file and the MVS data set FTP.DATA.

The FTP.DATA configuration data set is optional. As shown in Table 12, the FTP client uses one of the following search orders to obtain the local site parameter values:

**Table 12. FTP client search orders**

<table>
<thead>
<tr>
<th>TSO shell</th>
<th>z/OS UNIX shell</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SYSFTPD DD statement</td>
<td>1. $HOME/ftp.data</td>
</tr>
<tr>
<td>2. tso_prefix.FTP.DATA</td>
<td>2. userid.FTP.DATA</td>
</tr>
<tr>
<td>3. userid.FTP.DATA</td>
<td>3. /etc/ftp.data</td>
</tr>
<tr>
<td>4. /etc/ftp.data</td>
<td>4. SYS1.TCPPARMS(FTPDATA) data set</td>
</tr>
<tr>
<td>5. SYS1.TCPPARMS(FTPDATA) data set</td>
<td>5. tcpip.hlq.FTP.DATA file</td>
</tr>
<tr>
<td>6. tcpip.hlq.FTP.DATA file</td>
<td></td>
</tr>
</tbody>
</table>

Table 13 on page 64 describes the statements you can code in the FTP.DATA data set. It is not necessary to include all statements in the FTP.DATA data set, only those statements whose default values are to be changed. The hard-coded default is used for any statement not included in the FTP.DATA data set.

You can change several of the FTP local site parameters during the FTP session by using the LOCSITE subcommand. See "LOCSITE subcommand—Specify site information to the local host" on page 183 for more information about using the LOCSITE subcommand to change the local site parameters.

You can also use the FTP.DATA data set to override the default site parameters for the FTP server. Refer to the z/OS Communications Server: IP Configuration Reference for more information about overriding FTP server default site parameters.
Table 13. FTP.DATA data set statements for the FTP client

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAtrans</td>
<td>true</td>
<td>Permits the client to translate characters in the first column of ASA files being transferred as print control characters, if recognized as such.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Treats ASA file transfers as regular file transfers; that is, the ASA characters in column 1 are treated as part of the data and are not converted to print control character sequences. This is the hard-coded default.</td>
</tr>
<tr>
<td>AUTOMount</td>
<td>true</td>
<td>Permits automatic mounting of volumes for data sets on volumes that are not mounted. This is the hard-coded default.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Prevents automatic mounting of volumes for data sets on volumes that are not mounted.</td>
</tr>
<tr>
<td>AUTOREcall</td>
<td>true</td>
<td>Permits automatic recall of migrated data sets. This is the hard-coded default.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Prevents automatic recall of migrated data sets.</td>
</tr>
<tr>
<td>AUTOTAPEMount</td>
<td>true</td>
<td>The default in the FTP server.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>The default in the FTP client.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>Tape mounts issued in the client, while running in an interactive or batch TSO environment, are also governed by the user TSO MOUNT authority.</td>
</tr>
<tr>
<td>BLKsize</td>
<td>size</td>
<td>Specifies the block size of a newly allocated data set. If BLKsize is not specified at all, the hard-coded default is 6233. Valid range is 0-32760, where BLKsize 0 is a special case that indicates that the operating system is to determine the block size for new data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifying a size value is optional. Specify BLKsize with no size value to enable the block size from a model DCB data set or SMS dataclass to be used. See also the DATAClass and DCbdsn parameters in this table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLKsize is functionally equivalent to BLOCKSize. The BLOCKSize parameter is obsolete, but it is accepted to provide compatibility with previous releases.</td>
</tr>
<tr>
<td>BLOCKSize</td>
<td>size</td>
<td>Specifies the block size of a newly allocated data set. The hard-coded default is 6233. Valid range is 0–32760.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLOCKSize is functionally equivalent to BLKsize. The BLOCKSize parameter is obsolete but it is accepted to provide compatibility with previous releases.</td>
</tr>
<tr>
<td>BUfno</td>
<td>number</td>
<td>Specifies the number of access method buffers to be used when data is read from or written to a data set. Valid values are in the range 1–255. The hard-coded default is 5.</td>
</tr>
<tr>
<td>CCONNTIME</td>
<td>seconds</td>
<td>Defines the amount of time to wait after attempting to close a control connection before terminating it and reporting an error. The default is 30. The valid range for CCONNTIME is 0 (CCONNTIME not used) or 15-86400.</td>
</tr>
</tbody>
</table>
Table 13. FTP.DATA data set statements for the FTP client  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCTRANS</td>
<td>dsname</td>
<td>Specifies the SBCS translation table to be used for the control connection. FTP uses the translation table in the user_id.dsname.TCPXLBIN data set. If that data set does not exist, FTP uses the hlq.dsname.TCPXLBIN data set. FTP resolves the translation table to be used for the control connection in the following priority:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Data set specified in FTP subcommand with TRANSLATE option</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a. $HOME/ dsname.TCPXLBIN (z/OS UNIX environment only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b. user_id.dsname.TCPXLBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>c. hlq.dsname.TCPXLBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Data set specified in CTRLConn parameter in FTP.DATA data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Data set specified in CCTRANS parameter in FTP.DATA data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Default search order</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a. user_id.FTP.TCPXLBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b. hlq.FTP.TCPXLBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>c. user_id.STANDARD.TCPXLBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>d. hlq.STANDARD.TCPXLBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. 7-bit table build</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. FTP internal tables</td>
</tr>
<tr>
<td>CHKPTPrefix</td>
<td>value</td>
<td>Used to determine the hlq for the checkpoint file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Home</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Equal to 'tso_prefix.FTP.CHECKPOINT'. In the z/OS UNIX shell, it is 'current_path.FTP.CHECKPOINT'. This is the default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Userid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uses userid, 'userid.FTP.CHECKPOINT' for TSO, z/OS UNIX shell, and batch.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Local</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uses the current path as the hlq for TSO, z/OS UNIX shell, and batch. This is the directory set by the lcd command. If the directory is a PDS, then the checkpoint data set name will be 'path.pdsname(CHKPOINT)'.</td>
</tr>
<tr>
<td>CHKPTInt</td>
<td>interval</td>
<td>Specifies the checkpoint interval for the sending site in a file transfer request. This value is used to determine when checkpoint marker blocks are to be transmitted. The actual frequency of marker blocks is a function of the checkpoint interval value and the size of the buffers being used. If the checkpoint interval is 0, no checkpointing occurs and no marker blocks are transmitted. The default value is 0. <strong>Note:</strong> Use this parameter only if you are sending to a server that supports checkpointing.</td>
</tr>
</tbody>
</table>
### Table 13. FTP.DATA data set statements for the FTP client (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIPHERSUITE</td>
<td>SSL_NULL_MD5</td>
<td>The client and server specify the list of encryption types that they support. The client and server negotiate which of the available ciphers is used for the data encryption by specifying the desired ciphers in order of preference. The actual cipher used is the best match between what the server supports and what the client requests. If the server does not support any of the ciphers the client requests, the TLS handshake will fail and the connection will be closed if authentication is required by either client or server. Otherwise, the connection will continue without the use of TLS security.</td>
</tr>
<tr>
<td></td>
<td>SSL_NULL_SHA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSL_RC4_MD5_EX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSL_RC4_MD5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSL_RC4_SHA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSL_RC2_MD5_EX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSL_DES_SHA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSL_3DES_SHA</td>
<td></td>
</tr>
<tr>
<td>CLIENTERRCODES</td>
<td>true</td>
<td>Convert FTP return codes into a set of codes defined in &quot;FTP client error codes&quot; on page 100.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Convert an FTP return code. The converted code is the remainder of dividing the displayed return code by 4096 and should be in the range 0-4095. This is the default. If FTP is called from a REXX exec, the return code will be the 5 digit xxyyy format, where xx is the subcommand code and yyy is the reply from the server.</td>
</tr>
<tr>
<td>CONDdisp</td>
<td>CATLG</td>
<td>Specifies that a data set is kept and cataloged when an FTP file transfer ends prematurely. This is the hard-coded default.</td>
</tr>
<tr>
<td></td>
<td>DELETE</td>
<td>Specifies that a data set is deleted when a file transfer ends prematurely.</td>
</tr>
<tr>
<td>CTRLConn</td>
<td>7BIT</td>
<td>Indicates 7-bit ASCII is to be used on the control connection. This is the default if no TCPXLBIN data set is found.</td>
</tr>
<tr>
<td></td>
<td>iconv_ascii</td>
<td>Indicates a name recognized by iconv to indicate the ASCII code page to be used for the control connection.</td>
</tr>
<tr>
<td></td>
<td>FTP_STANDARD_TABLE</td>
<td>Indicates that the FTP internal tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), are to be used on the control connection.</td>
</tr>
<tr>
<td>DATAClass</td>
<td>Class</td>
<td>Specifies the data class for SMS created files as defined by your organization for the FTP client. To cancel the specification, use LOCSITE with the DATAC= parameter. See &quot;Specifying values for new data sets&quot; on page 86 for more information about specifying attributes when allocating new data sets.</td>
</tr>
<tr>
<td>DATACTTIME</td>
<td>seconds</td>
<td>Defines the amount time that the client waits after attempting to send or receive data before terminating the connection and reporting an error to the user. The default is 120. The valid range for DATACTTIME is 0 (DATACTTIME not used) or 15-86400.</td>
</tr>
<tr>
<td>DATETIMEOUT</td>
<td>seconds</td>
<td>Defines the amount of time that the server waits after attempting to send or receive data before terminating the connection and reporting an error to the user. The default is 300. The valid range for DATETIMEOUT is 0 (DATETIMEOUT not used) to 86400. If coded with value greater than 86400, the timer is set to 86400.</td>
</tr>
<tr>
<td>DB2</td>
<td>name</td>
<td>Specifies the name of the DB2\textsuperscript{®} subsystem. The default name is DB2.</td>
</tr>
</tbody>
</table>
### Table 13. FTP.DATA data set statements for the FTP client (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2PLAN</td>
<td>planname</td>
<td>Specifies the DB2 plan name to be opened by the FTP client. This name should match the name specified in the BIND job for the EZAFTPMQ DBRM used by FTP. If DB2PLAN is not present in FTP.DATA, the default plan name of EZAFTPMQ is used by the FTP client.</td>
</tr>
<tr>
<td>DCbdsn</td>
<td>name</td>
<td>Specifies the name of the MVS data set to be used as a model for allocation of new data sets. HFS file names are not allowed. There is no hard-coded default. <strong>Note:</strong> See “Generation data group support” on page 88 for information about the relationship between DCbdsns and GDGs in the CFTP and z/OS UNIX-FTP servers.</td>
</tr>
<tr>
<td>DCONNTIME</td>
<td>seconds</td>
<td>Defines the amount of time to wait after attempting to close a data transfer before terminating the connection and reporting an error. The default is 120. The valid range for DCONNTIME is 0 (DCONNTIME not used) or 15-86400.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>parameter</td>
<td>Enables or disables general internal tracing. <strong>Note:</strong> Only one parameter per DEBUG statement can be defined. The ? parameter cannot be used in FTP.DATA. See “DEBug subcommand—Set general trace options” on page 149 for information about parameter entries.</td>
</tr>
<tr>
<td>Directory</td>
<td>size</td>
<td>Specifies the number of directory blocks to be allocated for the directory of a PDS. The hard-coded default is 27.</td>
</tr>
<tr>
<td>DIRECTORYMode</td>
<td>true</td>
<td>Specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory. In directory mode, this qualifier is the only one used by the MPUT, MGET, LS, and DIR subcommands.</td>
</tr>
<tr>
<td>DUMP</td>
<td>parameter</td>
<td>Enables or disables extended internal tracing. <strong>Note:</strong> Only one parameter per DUMP statement can be defined. The ? parameter cannot be used in FTP.DATA. See “DUMP subcommand—Set extended trace options” on page 159 for information about parameter entries.</td>
</tr>
<tr>
<td>ENCODING</td>
<td>SBCS</td>
<td>Single Byte encoding. Codepages are specified via SBDATACONN statement. This is the default value.</td>
</tr>
<tr>
<td></td>
<td>MBCS</td>
<td>Multibyte encoding. Codepages are specified by the MBDATACONN statement. Indicate the type of data encoding on the network.</td>
</tr>
<tr>
<td>EXTENSIONS UTF8</td>
<td></td>
<td>Indicates the client supports UTF-8 encoded pathnames on the control connection. Initially, the client will use 7-bit ASCII on the control connection. During login, the client will query the server to determine whether UTF-8 is supported by the server. If the server does support UTF-8 encoding on the control connection, the client will issue a LANG command specifying the server’s current reply language.</td>
</tr>
<tr>
<td>FILEtype</td>
<td>type</td>
<td>Specifies the file type of the data set. The hard-coded default is SEQ. The description of each file type is:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td></td>
<td>SEQ</td>
<td>Sequential or partitioned data sets</td>
</tr>
<tr>
<td></td>
<td>SQL</td>
<td>SQL query function</td>
</tr>
</tbody>
</table>
# Table 13. FTP.DATA data set statements for the FTP client (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTPKEEPALIVE</td>
<td>seconds</td>
<td>Defines the control connection keepalive timer value in seconds. This is the number of seconds before a keepalive packet will be sent out on the FTP control connection. The default is 0, indicating that the TCP stack wide keepalive timer will be used for the control connection. The valid range for FTPKEEPALIVE is 0 (FTPKEEPALIVE not used) or 60–86400.</td>
</tr>
</tbody>
</table>
| FWFriendly      | true | Specifies that the FTP client is firewall-friendly. This means that data connections will be set up from the FTP client to the FTP server.  
false | Specifies that the FTP client is not firewall-friendly. This means that data connections will be set up from the FTP server to the FTP client. This is the default. |
| Note:           |       | When the connection to the server is IPv6, data connections are set up from client to server regardless of the FWFRIENDLY setting.               |
| INACTTIME       | seconds | Specifies the amount of time to wait for an expected response from the server, on either the control or the data connection, before closing the session. Data transfer times that exceed this value will not cause session termination unless the time between data packet arrivals exceeds this value. The hard-coded default is 120. The valid range for INACTTIME is 0 (INACTTIME not used) or 15–86 400. |
| ISPFStats       | true | FTP creates the statistics.  
false | FTP does not create statistics if the target PDS member does not already exist or does exist but does not have statistics. FTP updates the statistics and sends the message indicating the behavior, if the target PDS member already exists and has statistics. |
| KEYRING         | keyring | keyring is the name of the keyring. If the parameter starts with a slash (/), it is the name of an hfs file. Otherwise, it is the name of a RACF resource that defines the keyring. |
| LRecl           | length | Specifies the record length of a newly allocated data set. Valid values are 0–32756. The default is 256.  
Note:          | You can also use a value of x to indicate the maximum possible value. |
| MBDATACONN      | file_system_codepage, network_transfer_codepage | Specifies the names of the codepages that are recognized by iconv. |
| MGmtclass       | class | Specifies the SMS management class to be assigned to newly allocated data sets.  
To cancel the specification, use LOCSITE subcommand with the MG= parameter.  
See [Specifying values for new data sets](#) for more information about specifying attributes when allocating new data sets. |
<p>| MIGratevol      | valid | Indicates the volume ID (valid) for migrated data sets using non-IBM storage management systems. If you do not specify MIGratevol, the default valid is MIGRAT. |
| MYOPENTIME      | seconds | Defines the amount of time to wait for a session to open before terminating the attempt and reporting an error. The default is 60. The valid range for MYOPENTIME is 0 (MYOPENTIME not used) or 15-86 400. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETRCLEVEL</td>
<td>1</td>
<td>An IP address will not be converted to a host name. To match a passed IP address, the same IP address must be in the NETRC file. A DD card must point to the NETRC file for batch. FTP will not look for ‘userid.NETRC’ if there is no DD NETRC in the job. Note that this selection causes FTP to treat the NETRC file and data similar to the way release 320 and previous releases did. This is the DEFAULT.</td>
</tr>
<tr>
<td>NETRCLEVEL</td>
<td>2</td>
<td>All IP addresses will be converted to a host name before they are compared and if there is no DD NETRC an attempt will be made to read ‘userid.NETRC’.</td>
</tr>
<tr>
<td>PRImary</td>
<td>value</td>
<td>Specifies the amount of direct access storage for primary allocation of new data sets. The hard-coded default is 1.</td>
</tr>
<tr>
<td>QUOtesoverride</td>
<td>true</td>
<td>Indicates that single quotation marks (‘‘), at the beginning of, or surrounding, a file name are interpreted to mean that the file name contained inside the single quotation marks should override the current working directory instead of being appended to it. Any single quotation marks inside the file name are treated as part of the file name. This is the hard-coded default.</td>
</tr>
<tr>
<td>QUOtesoverride</td>
<td>false</td>
<td>Specifies that a single quote at the beginning of a file name, as well as all other single quotation marks contained in the file name, is treated as part of the actual file name. The entire file name, including the leading single quote, is appended to the current working directory.</td>
</tr>
<tr>
<td>RDW</td>
<td>true</td>
<td>Specifies that RDWs are treated as if they were part of the record and not discarded during FTP transmission of variable format data sets. This applies to transfers in stream mode only. <strong>Note:</strong> RDW information is stored in binary. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.</td>
</tr>
<tr>
<td>RDW</td>
<td>false</td>
<td>Specifies that RDWs are discarded during FTP transmission of variable format data sets. This applies to transfers in stream mode only. This is the hard-coded default.</td>
</tr>
<tr>
<td>RECfm</td>
<td>format</td>
<td>Specifies the record format of a data set. Valid record formats are: F, FA, FB, FBA, FB, FBSA, FBSM, FM, FS, FSA, FSM, U, UA, UM, V, VA, VB, VBA, VBM, VBS, VBSA, VBSM, VM, VS, VSA, and VSM. The default is VB. The characters used to specify these record formats have the following meanings: <strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>Fixed record length</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>Variable record length</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>Undefined record length</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>Blocked records</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>Spanned records (if variable) / Standard (if fixed)</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>Records contain ISO/ANSI control characters</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>Records contain machine code control characters</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RESTGet</td>
<td>true</td>
<td>The checkpoint data set will be opened for a GET request. This is the default.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>The checkpoint data set will not be opened for a GET request.</td>
</tr>
<tr>
<td>RETpd</td>
<td>days</td>
<td>Specifies the number of days that a newly allocated data set should be retained. The default is 0, which means no retention period. If you do not specify a retention period, no retention period is assigned to newly allocated data sets. The maximum is 9999.</td>
</tr>
<tr>
<td>SBDatasync</td>
<td>dsname</td>
<td>Specifies the fully qualified data set name or HFS file name that contains the EBCDIC-to-ASCII and ASCII-to-EBCDIC translate tables that are generated by the CONVXLAT utility.</td>
</tr>
<tr>
<td></td>
<td>file_system_cp, network_transfer_cp</td>
<td>Specifies the names of the code pages that are recognized by iconv.</td>
</tr>
<tr>
<td></td>
<td>FTP_STANDARD_TABLE</td>
<td>Indicates that the FTP internal tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), are to be used on the data connection.</td>
</tr>
<tr>
<td>SBSUB</td>
<td>true</td>
<td>Specifies that substitution is allowed for data bytes that cannot be translated. SBSUBCHAR defines the substitution value for untranslatable characters.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Specifies that substitution is not allowed for data bytes that cannot be translated. This causes data transfer failure if a character cannot be translated during the transfer. This is the default.</td>
</tr>
<tr>
<td>SBSUBCHAR</td>
<td>SPACE</td>
<td>When the target codeset is ASCII and SBSUB is true, replace untranslatable characters with X’20’ during SBCS data transfers. When the target codeset is EBCDIC and SBSUB is true, replace untranslatable characters with X’40’ during SBCS data transfers. This is the default.</td>
</tr>
<tr>
<td></td>
<td>nn</td>
<td>When SBSUB is true, replace untranslatable characters with nn during SBCS data transfers, where nn is a hexadecimal value from 00 to FF.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SBTRANS</td>
<td>dsn_qual</td>
<td>Specifies the SBCS translation table to be used for the data connection. This table is used for SBCS and DBCS data transfers. FTP uses the translation table in the user_id.dsn_qual_TCPXLBIN data set. If the user_id.dsn_qual_TCPXLBIN data set does not exist, FTP uses the hlq.dsn_qual_TCPXLBIN data set. FTP resolves the translation table to be used for the data connection in the following priority: 1. Data set specified in FTP subcommand with TRANSLATE option   a. $HOME/dsn_qual_TCPXLBIN (z/OS UNIX environment only)   b. user_id.dsn_qual_TCPXLBIN   c. hlq.dsn_qual_TCPXLBIN 2. Data set specified in SBDataconn parameter in FTP.DATA data set 3. Data set specified in SBTRANS parameter in FTP.DATA data set 4. Default search order   a. user_id.FTP_TCPXLBIN   b. hlq.FTP_TCPXLBIN   c. user_id.STANDARD_TCPXLBIN   d. hlq.STANDARD_TCPXLBIN 5. 7-bit table build 6. FTP internal tables</td>
</tr>
<tr>
<td>SECondary</td>
<td>value</td>
<td>Specifies the amount of direct access storage for secondary allocation of new data sets. The hard-coded default is 1.</td>
</tr>
<tr>
<td>SECURE_CTRLCONN</td>
<td>CLEAR</td>
<td>The control connection is not required to be integrity protected or encrypted. The client will allow the cprotect clear and ccc subcommands. This is the default.</td>
</tr>
<tr>
<td>SECURE_CTRLCONN</td>
<td>PRIVATE</td>
<td>The control connection is required to be integrity protected and encrypted. The client must issue a valid AUTH command before attempting to log on to the FTP server. The client will not allow the cprotect clear or cprotect safe subcommands.</td>
</tr>
<tr>
<td>SECURE_CTRLCONN</td>
<td>SAFE</td>
<td>The control connection is required to be integrity protected but not required to be encrypted. The client must issue a valid AUTH command before attempting to log on to the FTP server. The client will not allow the cprotect clear and ccc subcommands.</td>
</tr>
</tbody>
</table>
Table 13. FTP.DATA data set statements for the FTP client (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECURE_DATACONN</td>
<td>NEVER</td>
<td>The data connection is required to be neither integrity protected nor encrypted. The client will only allow the protect clear or clear subcommands.</td>
</tr>
<tr>
<td></td>
<td>CLEAR</td>
<td>The data connection is not required to be integrity protected or encrypted. The client will allow the protect, clear, safe, and private subcommands. CLEAR is the default.</td>
</tr>
<tr>
<td></td>
<td>PRIVATE</td>
<td>The data connection is required to be integrity protected and is required to be encrypted. The client must issue a valid AUTH command before attempting to log on to the FTP server. The client will only allow the protect private or private subcommands.</td>
</tr>
<tr>
<td></td>
<td>SAFE</td>
<td>The data connection is required to be integrity protected but not required to be encrypted. The client must issue a valid AUTH command before attempting to log on to the FTP server. The client will not allow the protect clear or clear subcommands.</td>
</tr>
<tr>
<td>SECURE_FTP</td>
<td>level</td>
<td>Specifies whether authentication is required. The valid values for level are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ALLOWED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies that authentication is allowed but not required. This is the default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>REQUIRED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies that authentication is required. For the client, the AUTH command is issued and is required to be accepted by the server.</td>
</tr>
<tr>
<td>SECURE_MECHANISM</td>
<td>TLS</td>
<td>Specifies that TLS is the security mechanism that is used by the client when it sends an AUTH command.</td>
</tr>
<tr>
<td></td>
<td>GSSAPI</td>
<td>Specifies that GSSAPI is the security mechanism that is used by the client when it sends an AUTH command. <strong>Note:</strong> Security mechanism GSSAPI is supported for IPv4 connections only.</td>
</tr>
<tr>
<td>SECURE_PBSZ</td>
<td>buffer_size</td>
<td>Specifies the maximum size of the encoded data blocks sent during file transfer. The client will initially issue the PBSZ command specifying buffer_size. If the PBSZ command is rejected, the client will reissue the PBSZ command with a smaller value until it is accepted by the server. The valid range for buffer_size is 512–32 768. The default value is 16 384.</td>
</tr>
<tr>
<td>SOCKSCONFIGFILE</td>
<td>file-path</td>
<td>Provide the name of an MVS data set or HFS file containing SOCKS configuration data. <strong>Note:</strong> When the server host is an IPv6 node, the SOCKSCONFIGFILE is ignored.</td>
</tr>
<tr>
<td>SPACETYPE</td>
<td>type</td>
<td>Specifies whether newly allocated data sets are allocated in blocks, cylinders, or tracks. Valid values for type are BLOCK, CYLINDER, or TRACK. The hard-coded default is TRACK.</td>
</tr>
<tr>
<td>SPRead</td>
<td>true</td>
<td>Specifies that the output is in spreadsheet format when the file type is SQL.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Specifies that the output is in report format rather than spreadsheet format when the file type is SQL. This is the hard-coded default.</td>
</tr>
</tbody>
</table>
Table 13. FTP.DATA data set statements for the FTP client (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLCol</td>
<td>type</td>
<td>Specifies what the column headings of the SQL output file are. The following list describes each type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Names</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Labels</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Any</strong></td>
</tr>
<tr>
<td>STORclass</td>
<td>class</td>
<td>Specifies the storage class for the SMS-created files as defined by your organization for the FTP client. To cancel the specification, use LOCSITE with the STO= parameter. See “Specifying values for new data sets” on page 86 for more information about specifying attributes when allocating new data sets.</td>
</tr>
<tr>
<td>TLSTIMEOUT</td>
<td>value</td>
<td>Provides a timeout value for TLS handshake processing. This timeout is the maximum time between full TLS handshakes. If this time period has not been reached since the last full handshake, a partial handshake occurs when a data connection is protected by TLS.</td>
</tr>
<tr>
<td>TRAILingblanks</td>
<td>true</td>
<td>Specifies that the FTP client preserves any trailing blanks that are in a fixed format data set when the data is sent to a foreign host (the trailing blanks are transmitted with the rest of the data).</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Specifies that the FTP client strips any trailing blanks that are in a fixed format data set when the data is sent to a foreign host (trailing blanks are not transmitted to the foreign host). This is the default.</td>
</tr>
<tr>
<td>TRUNcating</td>
<td>true</td>
<td>Indicates that if records are truncated, FTP does not set an error and file transfer continues. This is the default value.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Indicates that, if a record is detected that is longer than LRECL, an error will be set and the file transfer will fail. <strong>Note:</strong> If WRAPRECORD is set, the transfer continues.</td>
</tr>
<tr>
<td>UCOUNT</td>
<td>unit_count</td>
<td>Specifies number of devices to allocate. Valid range is 1–59. When specified without a value, the FTP server does not specify a unit count when allocating data sets.</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>Parallel mount request. The setting for UCOUNT may be overridden by the SMS storage class, if it is specified. Therefore, it is recommended that UCOUNT not be coded if an SMS storage class is in use.</td>
</tr>
<tr>
<td>UCSHOSTCS</td>
<td>code_set</td>
<td>Specifies the EBCDIC code set to be used when converting to or from Unicode. If you do not specify a <code>code_set</code>, the current code set is used.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UCSSUB</td>
<td>true</td>
<td>In Unicode-to-EBCDIC conversion, the EBCDIC substitution character is used to replace any Unicode character that cannot successfully be converted. Data transfer continues.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>In Unicode-to-EBCDIC conversion, the data transfer is terminated if any Unicode character cannot be converted into the EBCDIC code set. This is the hard-coded default.</td>
</tr>
<tr>
<td>UCSTRUNC</td>
<td>true</td>
<td>In Unicode-to-EBCDIC conversion, truncation of EBCDIC data is allowed. The data transfer continues even if EBCDIC data is truncated.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>In Unicode-to-EBCDIC conversion, truncation of EBCDIC data is not allowed. The data transfer is aborted if the logical record length of the receiving data set is too small to contain the data after conversion to EBCDIC. This is the hard-coded default.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>If data contains any double-byte characters, there is no guarantee that truncation will honor character boundaries or that records will end in shift-in state.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>The setting of the CONDdisp statement determines what happens to the target data set if the transfer is aborted.</td>
</tr>
<tr>
<td>UMask</td>
<td>octal_umask</td>
<td>Defines the file mode creation mask that specifies the permission bits that must be OFF when a new file is created. The default is 27.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>You cannot use FTP to create HFS files with execute permissions. If you require execute permissions, use the LOCSITE CHMOD command after the file has been created.</td>
</tr>
<tr>
<td>UNITNAME</td>
<td>type</td>
<td>Specifies the unit type (for example, 3380) for allocation of new data sets on direct access devices. If UNITNAME is not specified, the unit type used for allocation is the system default.</td>
</tr>
<tr>
<td>VCOUNT</td>
<td>volume_count</td>
<td>Specifies the number of volumes an allocated data set may span. Valid ranges span from 1–255. When specified without a value, the FTP server uses a volume count of 59 when allocating data sets.</td>
</tr>
<tr>
<td>VOLume</td>
<td>volume_serial</td>
<td>Specifies the volume serial number for allocation of new data sets.</td>
</tr>
<tr>
<td></td>
<td>volume_serial_list</td>
<td>A list of volume serial numbers for allocation.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>If volume_serial_list or volume_serial is not specified, the volume serial number used for allocation is set back to the system default.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>The setting for VOLume may be overridden by the SMS storage class, if one is specified. Therefore, VOLume should not be coded if an SMS storage class is in use.</td>
</tr>
<tr>
<td>WRAPrecord</td>
<td>true</td>
<td>Indicates that data is wrapped to the next record if there is no new line character.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Indicates that data is truncated if there is no new line character. This is the default.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>If TRUNCATE is also FALSE, and a record is detected that is longer than the LRECL of the new file is found, an error will be set and the file transfer will fail.</td>
</tr>
</tbody>
</table>
Table 13. FTP.DATA data set statements for the FTP client (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRTAPEFastio</td>
<td>true</td>
<td>Allow write to tape of ASCII stream data to use BSAM I/O routines.</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>Write to tape of ASCII stream data must use the LE Run-Time Library fwrite function.</td>
</tr>
</tbody>
</table>

Statistics for PDS members

ISPFStats can be set to either TRUE or FALSE in a client FTP.DATA or can be set using the LOCSITE subcommand. If ISPFStats is set to TRUE, FTP creates and maintains statistics for partitioned data set members. The following explains the effect ISPFStats has on PDS member statistics when you are issuing GET and MGET subcommands.

Note: ISPFStats is ignored for sequential data sets. Also, the record format must be either variable or fixed, and the record length must be less than 256. Transferring PDS member to PDS member in block mode or in compress mode differs in behavior from transferring in stream mode. If the user wants to preserve the statistics of the PDS member that already has the statistics and have the same statistics copied over to targeted PDS member, transferring in block mode or in compress mode is required.

- **Effect of ISPFStats setting when issuing GET or MGET when the file does not already exist**

  Whenever a PDS member is being transferred, FTP checks the setting of ISPFStats. If the member does not already exist, FTP will follow what the ISPFStats is set to. For example:
  - If ISPFStats is TRUE, FTP creates statistics for PDS members.
  - If ISPFStats is FALSE, FTP does not create statistics.

- **Effect of ISPFStats setting when issuing GET or MGET when the target PDS member already exists**

  Whenever a member is being transferred, FTP checks the setting of ISPFStats. If the targeted PDS member already exists, FTP will consider whether the target member has statistics and the setting of ISPFStats. For example:
  - If ISPFStats is TRUE and the existing member has statistics, FTP updates the statistics.
  - If ISPFStats is TRUE and the existing member does not have statistics, FTP creates the statistics.
  - If ISPFStats is FALSE and the existing member has statistics, FTP updates the statistics and sends a message indicating the behavior.
  - If ISPFStats is FALSE and the existing member does not have statistics, FTP does not create statistics.

Setting user-level options using FTPS.RC

The default values for the site parameters are coded in the server FTP.DATA. As a client user, you can change the site parameters by creating an FTPS.RC configuration data set. This file can have CD and SITE subcommands.

The FTPS.RC configuration data set is optional. The FTP server uses the following search order to find the data set:

1. tso_prefix.FTPS.RC
2. userid.FTPS.RC
3. $HOME/ftps.rc

The following are the syntax rules for the FTPS.RC file:

- Only SITE and CWD commands are allowed for this file. (CD will be also accepted and will be treated as CWD.)
- Each command must be contained in one line.
- SITE command can have multiple parameters.
- Comments are allowed, but the entire line will be commented out. In other words, comments cannot exist on the same line as the actual command line.
- Comments must begin with a semicolon.
- Comments can be entered between two or more command lines.

**Note:** Test your configuration file to ensure it is processed correctly by naming the file (following the preceding naming convention) and putting the file into the correct directory. A 230-type reply of Processing FTPS.RC configuration file.. will be displayed during your login process. The 230-type replies indicate any errors encountered or if the file was successfully opened and processed. Contact system programmer if you cannot fix any errors reported or if the file cannot be opened or created. Refer to [z/OS Communications Server: IP and SNA Codes](#) for additional information about 230-type replies.

The following are examples of an FTPS.RC file with no errors and with errors.

- **No errors (correctly coded).**

```plaintext
; This is a sample configuration file for FTPS.RC
; You may enter comments by starting a line with semicolon (;). The entire line will be ignored.
; The server FTP.DATA file must have the statement DEBUGONSITE TRUE
; for the SITE DEBUG= subcommand to be accepted by the server.
SITE debug=all
; You may also insert comments between command lines.
; You may code either 'CD' or 'CWD'.
CWD /user2
CD tmp
```

- **Errors within the file.**

```plaintext
; This is a sample configuration file for FTPS.RC
; You may enter comments by starting a line with semicolon (;). The entire line will be ignored.
; The line below has an error: errormount is an unrecognized parameter.
SITE errormount
; PUT subcommand is not accepted in this file.
PUT 'user2.tmp.banner' 'user3.tmp.banner'
```

**Configuring the FTP client for SOCKS server**

The FTP client uses a SOCKS configuration data set or file to determine whether to access a given FTP server directly or through a SOCKS server. The name of the data set or file is specified by coding the SOCKSCONFIGFILE statement in the client's FTP.DATA.

The FTP client references the SOCKSCONFIGFILE only when the FTP server is known to it by an IPv4 IP address or by a DNS name that resolves to an IPv4 IP address. The FTP client always connects directly to FTP servers known to it by IPv6 addresses or DNS names that resolve to IPv6 addresses.
Refer to the **z/OS Communications Server: IP Configuration Reference** for details regarding the contents of the SOCKS configuration file.

**Sample FTP.DATA data set (FTCDATA)**

The following is a sample of the contents of the FTP.DATA data set in the FTCDATA member of the *hlq.SEZAINST* data set.

```plaintext
;***********************************************************************
; * Name of File: SEZAINST(FTCDATA) *
; * Descriptive Name: FTP.DATA (for FTP Client) *
; * SMP/E Distribution Name: EZAFTPAC *
; * Copyright: Licensed Materials - Property of IBM *
; "Restricted Materials of IBM"
; 5694-A01
; (C) Copyright IBM Corp. 1977, 2002
; US Government Users Restricted Rights - Use, duplication or disclosure restricted by
; GSA ADP Schedule Contract with IBM Corp.
; Status: CSV1R4

This FTP.DATA file is used to specify default file and disk parameters used by the FTP client.

Note: For an example of an FTP.DATA file for the FTP server, see the FTPSDATA example.

Syntax Rules for the FTP.DATA Configuration File:

(a) All characters to the right of and including a ; will be treated as a comment.

(b) Blanks and <end-of-line> are used to delimit tokens.

(c) The format for each statement is:

   parameter value

The FTP.DATA options are grouped into the following groups in this sample FTP client FTP.DATA configuration data set:

1. Basic configuration options (timers, conditional options, etc.)
2. Defaults for MVS data set creation
3. Code page conversion options
4. DB2 (SQL) interface options
5. Security options
6. Debug (trace) options

For options that have a pre-selected set of values, a (D) indicates the default value for the option.

Options that can be changed via LOCSITE subcommands are identified with an (S).
```

Chapter 4. Transferring data using the File Transfer Protocol (FTP)  77
1. Basic FTP client configuration options -
  Timeout values, conversion options,
  and conditional processing options

ASATRANS FALSE ; (S) Conversion of ASA print
  control characters
  TRUE = Use C conversion
  FALSE = Do not convert (D)

AUTOMOUNT TRUE ; (S) Automatic mount of unmounted
  DASD volumes
  TRUE = Mount volumes (D)
  FALSE = Do not mount volumes

AUTORECALL TRUE ; (S) Automatic recall of
  migrated data sets
  TRUE = Recall them (D)
  FALSE = Do not recall them

AUTOTAPEMOUNT TRUE ; Automatic mount of unmounted
  tape volumes
  TRUE = Mount volumes (D)
  FALSE = Do not mount volumes

BUFNO 5 ; (S) Specify number of access
  method buffers
  Valid range is from 1 through
  35 - default value is 5

CCONNTIME 30 ; Timeout value for successful
  close of control connection.
  Default value is 30 seconds.
  Valid range is 15 through 720.

CHKPTINT 0 ; (S) Specify the checkpoint interval
  in number of records.
  NB: checkpointing only works
  with datatype EBCDIC and block
  or compressed transfer mode.
  0 = no checkpoints (D)

CONDDISP CATLG ; (S) Disposition of a new data set
  when transfer ends prematurely
  CATLG = Keep and catalog (D)
  DELETE = Delete data set

DATACTTIME 120 ; Timeout for send/receive data
  operations.
  Default value is 120 seconds.
  Valid range is 15 through 720.

DCONNTIME 120 ; Timeout value for successful
  close of data connection.
  Default value is 120 seconds.
  Valid range is 15 through 720.

DIRECTORYMODE FALSE ; (S) Specifies how to view the MVS
  data set structure:
  FALSE = All qualifiers below
  (D) LCWD are treated as
  entries in the directory
; TRUE = Qualifiers immediately below the LCWD are treated as entries in the directory

;EXTENSIONS UTF8 ; Enable RFC 2640 support.
; Default is disabled.
; Control connection starts as 7bit ASCII and switches to UTF-8 encoding when LANG command processed successfully. CCTRANS and CTRLCONN are ignored.

FILETYPE SEQ ; (S) Client mode of operation
; SEQ = transfer data sets or files (D)
; SQL = submit queries to DB2

INACTTIME 300 ; The time in seconds to wait for an expected response from the server.
; Default value is 300 seconds.
; Valid range is 15 through 720.

ISPFSTATS FALSE ; TRUE = create/update PDS statistics
; FALSE = does not create/update PDS statistics

MIGRATEVOL MIGRAT ; (S) Migration volume VOLSER to identify migrated data sets under control of non-HSM storage management products.
; Default value is MIGRAT.

MYOPENTIME 60 ; Connection timeout value in seconds.
; Default value is 60 seconds.
; Valid range is 15 through 720.

QUOTESOVERRIDE TRUE ; (S) How to treat quotes at the beginning or surrounding file names.
; TRUE = Override current working directory (D)
; FALSE = Treat quotes as part of file name

RDW FALSE ; (S) Specify whether Record Descriptor Words (RDWs) are discarded or retained.
; TRUE = Retain RDWs and transfer as part of data
; FALSE = Discard RDWs when transferring data (D)

;SOCKSCONFIGFILE /etc/socks.conf ; file path for SOCKS configuration file. The SOCKS configuration file specifies which FTP servers should be accessed via SOCKS

TRAILINGBLANKS FALSE ; (S) How to handle trailing blanks in fixed format data sets during text transfers.
; TRUE = Retain trailing blanks (include in transfer)
; FALSE = Strip off trailing

Chapter 4. Transferring data using the File Transfer Protocol (FTP) 79
UMASK 027 ; (S) Octal UMASK to restrict setting of permission bits when creating new HFS files. Default value is 027.

WRAPRECORD FALSE ; (S) Specify what to do if no new-line is encountered before reaching the MVS data set record length limit as defined by LRECL when transferring data to MVS.

TRUE = Wrap data to new record
FALSE = Truncate data (D)

---------------------------------------------------------------------

2. Default MVS data set creation attributes
---------------------------------------------------------------------

BLKSIZE 6233 ; (S) New data set allocation block size

; DATACLASS SMSDATA ; (S) SMS data class name
; There is no default

; MGMTCLASS SMSMGNT ; (S) SMS mgmtclass name
; There is no default

; STORCLASS SMSSTOR ; (S) SMS storclass name
; There is no default

; DCBDSN MODEL.DCB ; (S) New data set allocation model DCB name - must be a fully qualified data set name
; There is no default

DIRECTORY 27 ; (S) Number of directory blocks in new PDS/PDSE data sets.
; Default value is 27.
; Range is from 1 to 16777215.

LRECL 256 ; (S) New data set allocation LRECL.
; Default value is 256.
; Valid range 0 through 32760.

PRIMARY 1 ; (S) New data set allocation primary space units according to the value of SPACETYPE.
; Default value is 1.
; Valid range 1 through 16777215.

RECFM VB ; (S) New data set allocation record format.
; Default value is VB.
; Value may be specified as certain combinations of:
; A - ASA print control
; B - Blocked
; F - Fixed length records
; M - Machine print control
; S - Spanned (V) or Standard (F)
; U - Undefined record length
; V - Variable length records

RETPD ; (S) New data set retention period in days.
SECONDARY 1 ; (S) New data set allocation
; secondary space units according
to the value of SPACETYPE.
; Default value is 1.
; Valid range 1 through 16777215.

SPACETYPE TRACK ; (S) New data set allocation
; space type.
; TRACK (D)
; BLOCK
; CYLINDER

UCOUNT ; (S) Sets the unit count for an
; allocation.
; If this option is not specified
; or is specified with a value of
; blank, the unit count attribute
; is not used on an allocation (D)
; Valid range is 1 through 59 or
; the character P for parallel
; mount requests

;UNITNAME SYSDA ; (S) New data set allocation unit
; name.
; There is no default.

VCOUNT 59 ; (S) Volume count for an
; allocation.
; Valid range is 1 through 255.
; Default value is 59.

;VOLUME WRKLB1,WRKLB2 ; (S) Volume serial number(s) to
; use for allocating a data set.
; Specify either a single VOLSER
; or a list of VOLSERs
; separated with commas

; 3. Text code page conversion options
;
; 4. Transferring data using the File Transfer Protocol (FTP)
; ENCODING SBCS ; (S) Specifies whether multi-byte or single-byte data conversion is to be performed on ASCII data transfers.
; MBCS = Use multi-byte
; SBCS = Use single-byte (D)

; MBDATACONN (IBM-1388,IBM-5488) ; (S) Specifies the conversion table names for the data connection when ENCODING has a value of MBCS. The names are the file system code page name and the network transfer code page name.

; SBDATACONN (IBM-1047,IBM-850) ; (S) Specifies the file system/network transfer code pages for data connection.
; Either a fully-qualified MVS data set name or HFS file name built with the CONVXLAT utility -
; HLQ.MY.TRANS.DATASET
; /u/user1/my.trans.file
; Or a file system code page name followed by a network transfer code page name followed by a network transfer code page name according to iconv supported code pages -
; for example
; (IBM-1047,IBM-850)
; If the SYSFTSX DD-name is present it will override SBDATACONN.
; If neither SYSFTSX nor SBDATACONN are present, std. search order for a default translation table data set will be used.

; SBSUB FALSE ; Specifies whether untranslatable data bytes should be replaced with SBSUBCHAR when detected during SBCS data transfer.
; TRUE = Replace each untranslatable byte with SBSUBCHAR.
; FALSE = Terminate transfer (D) when untranslatable bytes are detected

; SBSUBCHAR nn ; Specifies the substitution character for SBCS data transfer when SBSUB is TRUE.
; nn = hexadecimal value from 0x'00' to 0x'FF'.
; SPACE = x'40' when target code set is EBCDIC, and
; x'20' when target code set is ASCII. (D)

; SBTRANS dsn_qual ; Data connection translate table data set qualifier.
; Used to search for
; a) userid.dsn_qual.TCPXLBIN
; b) hlq.dsn_qual.TCPXLBIN
; If SBDATACONN is specified, that value overrides SBTRANS

; UCHOSTCS code_set ; (S) Specify the EBCDIC code set to be used for data conversion
to or from Unicode.
; If UCSHOSTCS is not specified,
; the current EBCDIC code page
; for the data connection is used.

UCSSUB FALSE ; (S) Specify whether Unicode-to-EBCDIC
; conversion should use the EBCDIC
; substitution character or
; cause the data transfer to be
; terminated if a Unicode
; character cannot be converted to
; a character in the target
; EBCDIC code set
; TRUE = Use substitution char
; FALSE = Terminate transfer (D)

UCSTRUNC FALSE ; (S) Specify whether the transfer
; of Unicode data should be
; aborted if truncation
; occurs at the MVS host
; TRUE = Truncation allowed
; FALSE = Terminate transfer (D)

; 4. DB2 (SQL) interface options

DB2 DB2 ; (S) DB2 subsystem name
; The default name is DB2

DB2PLAN EZAFTPMQ ; DB2 plan name for FTP client
; The default name is EZAFTPMQ

SPREAD FALSE ; (S) SQL spreadsheet output format
; TRUE = Spreadsheet format
; FALSE = Not spreadsheet
; format (D)

SQLCOL NAMES ; (S) SQL output headings
; NAMES = Use column names (D)
; LABELS = Use column labels
; ANY = Use label if defined,
; else use name

; 5. Security options

;SECURE_MECHANISM GSSAPI ; Name of the security mechanism
; that the client uses when it
; sends an AUTH command to the
; server.
; GSSAPI = Kerberos support
; TLS = TLS

;SECURE_FTP ALLOWED ; Authentication indicator
; ALLOWED (D)
; REQUIRED

;SECURE_CTRLCONN CLEAR ; Minimum level of security for
; the control connection
; CLEAR (D)
; SAFE
; SECURE_DATACONN CLEAR ; Minimum level of security for the data connection
; NEVER ; CLEAR (D)
; SAFE ; PRIVATE

;SECURE_PBSZ 16384 ; Kerberos maximum size of the encoded data blocks
; Default value is 16384
; Valid range is 512 through 32768

; Name of a ciphersuite that can be passed to the partner during the TLS handshake. None, some, or all of the following may be specified. The number to the far right is the cipherspec id that corresponds to the ciphersuite's name.
;CIPHERSUITE SSL_NULL_MD5 ; 01
;CIPHERSUITE SSL_NULL_SHA ; 02
;CIPHERSUITE SSL_RC4_MD5_EX ; 03
;CIPHERSUITE SSL_RC4_MD5 ; 04
;CIPHERSUITE SSL_RC4_SHA ; 05
;CIPHERSUITE SSL_RC2_MD5_EX ; 06
;CIPHERSUITE SSL_DES_SHA ; 09
;CIPHERSUITE SSL_3DES_SHA ; 0A

;KEYRING name ; Name of the keyring for TLS
; It can be the name of an HFS file (name starts with /) or a resource name in the security product (e.g., RACF)

;TLSTIMEOUT 100 ; Maximum time limit between full TLS handshakes to protect data connections
; Default value is 100 seconds.
; Valid range is 0 through 86400

; ---------------------------------------------------------------------
; 6. Debug (trace) options
; ---------------------------------------------------------------------

;DEBUG TIME ; time stamp client trace entries
;DEBUG ALL ; activate all traces
;DEBUG BAS ; active basic traces (marked with *)
;DEBUG FLO ; function flow
;DEBUG CMD ; * command trace
;DEBUG PAR ; parser details
;DEBUG INT ; * program initialization and termination
;DEBUG ACC ; access control (logging in)
;DEBUG SEC ; security processing
;DEBUG UTL ; utility functions
;DEBUG FSC(1) ; * file services
;DEBUG SOC(1) ; * socket services
;DEBUG SQL ; special SQL processing
Using different SBCS translation tables for the control and data connections

If the SBCS table you need for SBCS or DBCS data transfer does not support the standard encodings for the portable character set, you should establish different translation tables for the control and data connections. This avoids unrecognized ASCII conversion of FTP commands.

You can specify different conversions for the data connection by using any of the following:

- FTP.DATA statements: SBDATACONN, SBTRANS, UCSHOSTCS.
- Use LOCSITE SBDataconn or LOCSITE XLATE statements to set the code page.
- Specify the TRANSLATE parameter as an FTP client start option.

You can change the control connection using CTRLConn statements in FTP.DATA, using the LOCSITE CTRLCONN subcommand, or by specifying the TRANSLATE parameter as an FTP client start option.

If you code EXTENSIONS UTF8 in the client’s FTP.DATA, the control connection will use 7-bit ASCII for commands and, when negotiated with the server, UTF-8 encoding of pathnames. The client can override the EXTENSIONS UTF8 statement by use of either the ftp start parameter, or by issuing SITE and LOCSITE commands. However, the client will not be able to resume UTF-8 encoding on the control connection until you start the client again.

Using different MBCS translation tables for the data connection

You can specify that you want to use the Chinese standard GB18030 provided by the codepage IBM-5488 for data conversion on your data connections. To use the codepage IBM-5488, you must specify that multibyte encoding is to be used. You can do this in one of the following ways:

- Code the following statement in the FTP.DATA file:
  ENCODING MBCS

- Issue a subcommand:
  LOCSITE ENCODING=MBCS

You can then specify which codepage the IBM-5488 encoded data is be converted to or from in the file system by doing one of the following:

- Code one of the following statements in the FTP.DATA file:
  MBDATAConn (IBM-1388,IBM-5488)

  or
  MBDATAConn (UTF-8,IBM-5488)

- Issue one of the following subcommands:
  LOCSITE MBDATAConn=(IBM-1388,IBM-5488)

  or
  LOCSITE MBDATAConn=(UTF-8,IBM-5488)

Note that these steps control the client’s end of the data connection. To request the same conversions at the server’s end of the data connection, the same statements must be added to the server’s FTP.DATA file or use the SITE subcommand.
Specifying values for new data sets

When allocating new data sets, there are several methods you can use to specify the data set attributes. You can individually use the storage attribute parameters with the SITE and LOCSITE commands or the FILE FTP DATA data set. Or, if your system administrator has used the Storage Management Subsystem to group together default attributes into named classes, you can specify those class names on the DATACLASS, STORCLASS, and MGmtCLASS parameters.

Dynamic allocation

FTP enables you to dynamically allocate a new physical-sequential data set or a partitioned data set (PDS) for the purpose of transferring data to be written to that data set. The following optional allocation variables can be used by the client to override and turn off the hard-coded defaults that affect the allocation of the data set.

<table>
<thead>
<tr>
<th>Variable</th>
<th>FTP.DATA parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocation units</td>
<td>SPACETYPE</td>
</tr>
<tr>
<td>blocksize</td>
<td>BLKSIZE</td>
</tr>
<tr>
<td>data class</td>
<td>DATACLASS</td>
</tr>
<tr>
<td>directory blocks</td>
<td>DIRECTORY</td>
</tr>
<tr>
<td>logical record length</td>
<td>LRECL</td>
</tr>
<tr>
<td>management class</td>
<td>MGMTCLASS</td>
</tr>
<tr>
<td>model DCB values</td>
<td>DCBDSN</td>
</tr>
<tr>
<td>primary space</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>record format</td>
<td>RECFM</td>
</tr>
<tr>
<td>retention period</td>
<td>RETPD</td>
</tr>
<tr>
<td>secondary space</td>
<td>SECONDARY</td>
</tr>
<tr>
<td>storage class</td>
<td>STORCLASS</td>
</tr>
<tr>
<td>unit</td>
<td>UNITNAME</td>
</tr>
<tr>
<td>volume count</td>
<td>VCOUNT</td>
</tr>
<tr>
<td>unit count</td>
<td>UCOUNT</td>
</tr>
<tr>
<td>volume serial number</td>
<td>VOLUME</td>
</tr>
</tbody>
</table>

Some of these allocation variables might provide duplicate information. For example, the model DCB may have a record format (RECFM) that differs from the record format specified by a data class and from the one explicitly specified by the client. FTP passes all variables that are specified to dynamic allocation and lets it determine which of the specifications takes precedence. The following list describes the exceptions to that policy:

- If neither the primary nor secondary space quantity is specified, the allocation units value is not sent.
- If the data set organization is physical-sequential, directory blocks specification is not sent.
- Otherwise, all variables are sent to dynamic allocation where the order of precedence is:
  1. Any FTP.DATA, SITE, or LOCSITE parameters explicitly specified or defaulted
2. Any attributes picked up from the model DCB and not otherwise explicitly specified
3. Any attributes picked up from the data class and not previously derived from 1 or 2
4. Any allocation defaults

Storage Management Subsystem (SMS)

An FTP client can specify one or more of the following Storage Management Subsystem (SMS) classes to manage characteristics that are associated with or assigned to data sets.

- Data class is an SMS construct that determines data set allocation attributes used by SMS for creation of data sets. The fields listed are available attributes that serve as a template for allocation. Each is *optional* and is overridden by any explicit specification of FTP allocation variables or by a model DCB (DCBDSN).

<table>
<thead>
<tr>
<th>Variable</th>
<th>FTP.DATA parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>directory blocks</td>
<td>DIRECTORY</td>
</tr>
<tr>
<td>logical record length</td>
<td>LRECL</td>
</tr>
<tr>
<td>primary space</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>record format</td>
<td>RECFM</td>
</tr>
<tr>
<td>retention period</td>
<td>RETPD</td>
</tr>
<tr>
<td>secondary space</td>
<td>SECondary</td>
</tr>
</tbody>
</table>

**Note:** If either primary or secondary space is explicitly specified, the primary and secondary values from data class are not used.

- Management class is an SMS construct that determines Data Facility Hierarchical Storage Manager (DFHSM) action for data set retention, migration, backup, and release of allocated but unused space. Management class replaces and expands attributes that otherwise would be specified. That is, management class may override any other specification of retention period.

- Storage class is a list of storage performance and availability services requests for an SMS-managed data set that SMS attempts to honor when selecting a volume or volumes for the data set. It may conflict with an explicit specification of volume and unit. If storage class is used, volume and unit parameters should be unspecified.

Steps for using a DCBDSN model to create a new data set

Perform the following steps to use a DCBDSN model to create a data set.

1. Issue the following command:
   
   ```
   SITE DCBDSN=data_set_name
   
   where data_set_name is the name of the data set to be used as a model to set the values of the logical record length (LRecl), the block size (BLKsize), the retention period (RETpd), and the record format (RECFm) of a new data set.
   ```

2. Issue the following command to enable the LRecl, BLKSize, and RECFm of the model to be used:

   ```
   SITE LRECL BLKSIZE RETPD RECFM
   ```
3. Issue the following command to create the new data set with the values specified by the DCBDSN model:

```
PUT data_set_name
```

where `data_set_name` is the name of the new data set.

**Note:** If you are using a non-MVS client that does not support the SITE command, you may be able to send the SITE command to the MVS server by using the QUOTE command. For example:

```
QUOTE SITE DCBDSN=data_set_name
```

---

**Generation data group support**

Generation data groups (GDGs) enable you to store multiple data sets, called generation data sets (GDSs), as versions of the GDG. You cannot use FTP to create a new GDG, but you can use it to create a new version (that is, a new GDS) or to transfer an existing version of an existing GDG.

The relationship between DCBDSN and GDGs is governed by MVS allocation rules rather than FTP usage rules. Therefore, when creating a new GDG [put `sys1.proclib(jes2)` user77.mygdg(+1)], at least one of the following must be true:

- A valid MODEL or PATTERN DSCB (for FTP, DCBDSN) specification must be coded in the FTP.DATA file when the CFTP server is started.
- A valid SITE DCbdsn=dataset_name must be issued before a PUT command is issued.
- A data set having the same name as the GDG base must reside on the volume as the user catalog that contains the GDG definition. In this case, neither a SITE DCbdsn nor a DCBDSN argument in the FTP.DATA data file is required. Allocation detects that a GDG is being created and looks in the VTOC of the volume containing the USERCATALOG for a data set (uncataloged) that has the same name as the GDG BASE (see the sample GDG JCL on page 89).

**Notes:**

1. A model or pattern DSCB that is the same name as the GDG BASE cannot exist on an SMS managed volume. This is an SMS restriction and is documented in the DFP manuals pertaining to using data sets (generation data sets or generation data groups).
2. Allocation does not generally have any requirements about the characteristics of a MODEL DSCB (cannot be VSAM, must be on DASD). Most facilities create one model DSCB for the entire system and everyone uses that model. The system-wide model usually has no logical record length (LRecl), block size (BLKsize), record format (RECfm), data set organization (DSORG) or retention period (RETpd) associated with it.
3. The existing z/OS UNIX-FTP server requires the MODEL DSCB to have a valid DSORG of physical sequential organization (PS). Otherwise the SITE command for the DCBDSN is ignored, and a message is issued indicating the DCBDSN was ignored.

The following restrictions apply:

- DCBDSN=USER.MYGDG(0)/ USER.MYGDG(-n), not supported
- DCBDSN=SYS1.PROCLIB(JES2), specifying a member of a PDS is not valid
• DCBDSN=SYS1.PROCLIB, valid

• The data set referenced on the DCBDSN, a DSORG of PS needed (FTP requirement)

Note: If there are explicit values associated with LRecl, BLKsize, RECfm, or the SMS management equivalent parameters, these explicit parameters override the values associated with the model DSCB specified on the DCBDSN.

The following is a sample Job Control Language (JCL) to create a model and the GDG BASE:

```
USER77.MYGDG -MODEL/PATTERN
VOL=SER=CPDLB1 -Volume having USERCATALOG, where USER77 is defined
(NAME(USER77.MYGDG) -GDG BASE definition
//USER77X JOB MSGLEVEL=(1,1),MSGCLASS=D,NOTIFY=USER77
//GDGA EXEC PGM=IDCAMS
//*
//GDGMOD DD DSN=USER77.MYGDG,
// VOL=SER=CPDLB1,
// UNIT=SYSALLDA,
// SPACE=(TRK,(0)),
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=6800,DSORG=PS),
// DISP=(,KEEP)
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
DEFINE GENERATIONDATAGROUP -
(NAME(USER77.MYGDG) -
EMPTY -
NOSCRATCH -
LIMIT(255) )
```

**GDG examples**

Before you specify a (+nnn) value to create a new GDS, issue the following command:

```
SITE DCBDSN=model
```

This subcommand specifies an MVS data set to be used as a model. The model must have a DSORG of PS. The other DCB characteristics of the data set are not checked.

**Notes:**

1. Failure to have a valid DCBDSN before trying to create a new GDS may cause FTP or ALLOCATION to fail or to return unpredictable results.
2. If you issue a SITE DCbdsn LRecl BLKsize command before the creation of a new data set, the LRecl and BLKsize parameters on the SITE command override the LRecl and BLKsize parameters on the DCbdsn command.

For more information about GDGs, refer to *MVS/DFP Version 3 Release 3: Using Data Sets*.

The following are sample FTP commands that access a GDG called JIMKEO.GDG.

**Notes:**

1. In the following examples, gdg (0), gdg (-1), and gdg (+1) specify which copy of the GDG you are using. 0 indicates the latest version, -1 indicates the previous version, and +1 indicates that a new version is created.
2. GDGALL is not supported by the C-FTP server. GDGALL processing occurs when the base name for the GDG is specified without a relative index value.
3. The MPUT and MGET commands are not applicable for GDGs.

The following example illustrates a PUT to the latest existing GDS. The working directory at the server is JIMKEO:

```
Command:
put my.gdg gdg(0)
>>>SITE FIXrecfm 150
200-Blocksize must be a multiple of lrecl for FB data sets. Blocksize set to 6150.
200 Site command was accepted
>>>PORT 129,34,128,245,126,229
200 Port request OK.
>>>STOR gdg(0)
125 Storing data set JIMKEO.GDG.G0055V00
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 3.24 Kbytes/sec.
```

The following example illustrates a GET of the latest GDG:

```
Command:
get gdg(0) my.gdg2
>>>PORT 129,34,128,245,126,233
200 Port request OK.
>>>RETR gdg(0)
125 Sending data set JIMKEO.GDG.G0055V00 FIXrecfm 150
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 3.04 Kbytes/sec.
```

The following example illustrates a PUT to a new GDS [After the STOR is complete, this new version is referenced by (0)].

```
Command:
put my.gdg gdg(+1)
>>>SITE FIXrecfm 150
200 Site command was accepted
>>>PORT 129,34,128,245,126,234
200 Port request OK.
>>>STOR gdg(+1)
125 Storing data set JIMKEO.GDG.G0056V00
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 1.16 Kbytes/sec.
```

The following example illustrates a GET of the previous GDS into the local file called my.gdg3:

```
Command:
get gdg(-1) my.gdg3
>>>PORT 129,34,128,245,126,239
200 Port request OK.
>>>RETR gdg(-1)
125 Sending data set JIMKEO.GDG.G0055V00 FIXrecfm 150
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 2.77 Kbytes/sec.
```

The following example illustrates a GET that replaces the contents of my.gdg3 with the most recent GDS:
Command:
get gdg(0) my.gdg3 (replace
>>>PORT 129,34,128,245,126,243
200 Port request OK.
>>>RETR gdg(0)
125 Sending data set JIMKEO.GDG.G0056V00 FIX recfm 150
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 3.36 Kbytes/sec.

The following example illustrates changing the working directory:

Command:
cd gdg
>>>CWD gdg
257 "JIMKEO.GDG." is working directory name prefix.

The following example shows the files created:

Command:
dir
>>>MODE s
200 Data transfer mode is Stream.
>>>PORT 129,34,128,245,127,12
200 Port request OK.
>>>LIST
125 List started OK.
Volume Unit Referred Ext Used Recfm Lrecl Blksz Dsorg Dsname
STRG73 3380K 04/30/92 1 5 FB 150 32700 PS G0003V00
STRG65 3380K 04/30/92 1 5 FB 150 32700 PS G0006V00
STRG61 3380K 04/30/92 1 5 FB 150 32700 PS G0010V00
STRG47 3380K 04/30/92 1 5 FB 150 32700 PS G0015V00
STRG47 3380K 04/30/92 1 5 FB 150 32700 PS G0021V00
STRG66 3380K 04/30/92 1 5 FB 150 32700 PS G0028V00
STRG47 3380K 04/30/92 1 5 FB 150 32700 PS G0036V00
STRG59 3380K 04/30/92 1 5 FB 150 32700 PS G0055V00
250 List completed successfully.
>>>MODE b
200 Data transfer mode is Block.
Command:

---

Submitting FTP requests in batch

FTP is usually run interactively by starting and entering commands from your terminal. You can also run FTP as a batch job, but you must supply the JCL file. You can use batch when you know what functions you want to perform, when you want a hardcopy of the results, or when you want to perform an FTP function many times.

Notes:

1. FTP can be run in batch either by specifying data sets for input and output as in Figure 3 on page 93 or without referring to data sets for input and output as in Figure 5 on page 94.

2. The file containing the FTP commands cannot have sequenced line numbers in it. If each line does not end in a semicolon, the FTP command file must be saved unnumbered.

3. Comments can be added to the command input file through the REXXX program that stacks the commands. The support includes stand-alone comment records and comments appended to the end-of-line.
The syntax for adding stand-alone comments is to use a semicolon (;) as the first nonspace character on a line. For example,

; This is a stand-alone comment record

You can add a command line comment by appending the end of the command line with a space-semicolon followed by the comment. For example,

USER userx ; This is an appended comment

When a user ID or password is expected (including passwords required for read or write access to files or disks), the entire line has to be blank. This example shows a blank password between user id and a comment line.

USER userx

; The line above is a blank password

This example shows a blank user ID between a comment line and password.

; This below is a blank userid

mypasswd

4. A blank followed by a plus sign ( + ) at the end of an FTP subcommand line is used as a continuation indicator for all FTP subcommands except the QUOTE subcommand. When the continuation indicator is encountered at the end of an FTP subcommand line, the next line is appended to the subcommand. For example,

PUT SOURCE.DS.NAME +
DEST.DS.NAME

is interpreted as:

PUT SOURCE.DS.NAME DEST.DS.NAME

5. To have the FTP client perform DB/2 queries in a batch job, the DSNLOAD library must be in the link list or appear on a STEPLIB DD statement for the job.

6. When connecting to a server such as a UNIX server where user IDs, passwords, directory names, and file names are case-sensitive, the data in the FTP batch job must be in the correct case.

7. Use the EXIT parameter if you want FTP to display an error return code and then exit when certain errors are detected. See "FTP EXIT return codes" on page 97 for more information.

Because z/OS UNIX batch condition codes range from 0 to 4095 (X'FFF'), the step condition code displayed by the JCL will not match the exit return code that FTP displays in its message. What is displayed is the remainder of dividing the FTP exit return code by 4096. When viewed in hex, this is the same as the last three hex digits of the exit return code.

For example, a PUT that fails and returns the message:

"550 Open failed for ..."

would get an exit return code of 27550. If you divide 27550 by 4096, the remainder of 2974 would be the JCL STEP condition code.

It is easier to visualize this in hex, where decimal 27550 becomes hex 6B9E. The last three hex digits, B9E, is the decimal number 2974.

Figure 3 on page 93 shows an example of the JCL required to submit a batch job by referring to data sets for input and output.
Notes:

1. REGION=2048K is a minimum requirement. The requirement could increase depending on the block size of the data set being transmitted.

2. The first JCL statement is a standard job statement. The next JCL statement is an EXEC statement. It has PGM=FTP (a region parameter) because FTP might use more storage than your default region size, and a PARM field.

3. For PARM=, you can specify any parameter that is valid when invoking FTP from your terminal. See "Using the FTP command on page 29" for more information. These parameters are only supported on the PARM= field of the EXEC card.

As shown in Figure 3, to run FTP in batch, you must have the following three DD statements:

- **SYSPRINT DD**: Alternative name for the OUTPUT DD statement.
- **OUTPUT DD**: Specifies the data set where FTP is to place the client messages and server replies generated during the FTP session.
- **INPUT DD**: Specifies the data set where the FTP subcommands to be performed are located.

**Note**: The data set specified on the OUTPUT DD statement should have an LRecl of 160 with any block size that is a multiple of the LRecl. The data set specified on the INPUT DD statement should have an LRecl of 80 with any block size that is a multiple of the LRecl.

You can use the user_id.NETRC data set, as defined by the NETRC DD statement in Figure 3, to identify the user ID and password for a batch-processed remote login. You can also specify the user ID and password in the INPUT DD data set.

Figure 4 shows the records in an INPUT DD data set that contains the FTP commands to be executed.

```plaintext
HOSTNAME
USERID PASSWD
DIR
PUT MYFILE.LISTING
QUIT
```

**Figure 4. Contents of an INPUT DD data set**

The first line of Figure 4 contains the name of the host that you want FTP to use. The second line contains the user ID followed by its password. The next three lines contain the FTP commands that you want FTP to perform. In this example, FTP is
doing a directory listing of the server to which you are connecting. The example
then instructs FTP to send a file to the server. The last line ends the connection.

Any client messages and server replies to the commands you execute appear in
the OUTPUT DD data set. The SYSPRINT can contain some additional messages
that relate to the execution of your FTP session.

Note: If you do not want your password to be copied to the output file, specify your
user ID and password on separate input lines. See Figure 5 for an example.

Submitting requests without input and output data sets

Figure 5 shows an easier way to submit a batch job, because you can avoid
referring to data sets for input and output.

```
//USERIDX JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID,MSGCLASS=H,TIME=9
//FTP EXEC PGM=FTP,REGION=4096K
//INPUT DD *
nodeid
userid
password
CD
DIR
GET hostfile.name locfile.name
QUIT
```

Figure 5. JCL to run FTP in batch without using data sets

Figure 6 shows step 1 creating a new GDS in batch and FTP getting the data set.

```
//USERIDX JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID,MSGCLASS=H,TIME=9
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*  
//SYSIN DD DUMMY  
//SYSUT1 DD DSN=USER31.SOURCE.DATA,DISP=SHR  
//SYSUT2 DD DSN=MYGDG(+1),DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,(1,1)),DCB=(MODEL)
//FTP EXEC PGM=FTP,REGION=4096K
//INPUT DD *
nodeid
userid
password
CD /u/joe
DIR
GET MYGDG(0) A.DATA.SET
QUIT  
//SYSPRINT DD SYSOUT=
```

Figure 6. Job to create a new GDS in batch

Note: All the GDG allocation in batch must be complete before the start of FTP.

Using the EXEC interface

The FTP EXEC interface enables you to execute FTP commands from an EXEC
rather than interactively from a terminal. The FTP subcommands to be performed
can be in a file (MVS data set or HFS file), or you can code them directly in the
EXEC.
By default, the FTP session dialog is printed on the terminal. If you want the dialog sent to a data set rather than the terminal do one of the following:

**TSO**
Specify an OUTPUT data set as part of the ALLOC statement.

**z/OS UNIX System Services**
Redirect the output to an HFS file when invoking the FTP command.

The following examples are written in REXX. Refer to the [z/OS TSO/E REXX Reference](#) and [z/OS Using REXX and z/OS UNIX System Services](#) for more information about the REXX language.

### Issuing FTP subcommands from a file

[Figure 7](#) is an example of an EXEC that issues FTP subcommands from a data set. In this example, the FTPIN1 data set is used for the FTP subcommands, and FTPOUT1 is used to store the FTP session dialog. This example must be invoked from TSO.

```rexx
/*REXX*/
"ALLOC DA(FTPIN1) DD(INPUT) SHR REU" /* Input will be from FTPIN1 */
if rc ¬= 0 then do
  say 'Error in ALLOC INPUT, rc = ' rc
  exit
end
"ALLOC DA(FTPOUT1) DD(OUTPUT) SHR REU" /* Output goes to FTPOUT1 */
if rc ¬= 0 then do
  say 'Error in ALLOC OUTPUT, rc = ' rc
  exit
end
"FTP YKTVSH" /* FTP to the YKTVSH host */
"FREE DD(INPUT)"
"FREE DD(OUTPUT)"
EXIT
```

**Figure 7. How to issue the FTP subcommands from a data set**

[Figure 8 on page 96](#) is an example of an EXEC that issues FTP subcommands from an HFS file. In this example, the name of the input file and output file are passed as arguments on the EXEC. For example,

```bash
EXAMPLE1 /u/user117/ftpin1 /u/user117/ftput1
```

where EXAMPLE1 is the name of the EXEC. This example must be invoked from the z/OS UNIX shell.

---

Chapter 4. Transferring data using the File Transfer Protocol (FTP) 95
The following is an example of the input file (either the input data set FTPIN1 or the HFS file /u/user117/ftpin1).

krasik mvsftp
cd examples
put t.info t1.info
get t1.info t2.info (r quit

Where:
krasik is the user ID
mvsftp is the password
t.info is the file to be transferred

Issuing FTP subcommands directly from the EXEC interface

Issuing FTP subcommands directly from the EXEC interface [Figure 9 on page 97] is an example of how to issue FTP subcommands directly from a REXX EXEC. The REXX stack is used to hold the FTP subcommands. This example runs in both the TSO and z/OS UNIX environments.

Note: To use FTP in a z/OS UNIX environment, TSO users must be authorized users or have a default z/OS UNIX user ID.
FTP EXIT return codes

FTP EXIT return codes are displayed (in message EZA1735I) when the (EXIT parameter is used on the FTP command and an error is detected. The return codes are composed of a subcommand code and a reply code. FTP EXIT return codes have the following format:

yyxxx

Where:

 yy     Represents the subcommand code, which is a number in the range 1–99. Each subcommand has an EXIT_IF_ERROR flag that determines whether FTP is exited when an error occurs if you specified EXIT on the FTP command. See Table 14 on page 98 for a description of the possible FTP subcommand codes.

 xxx    Represents the reply code that is sent from the server. The reply code is a 3-digit number. For a complete list of FTPD reply codes see the z/OS Communications Server: IP and SNA Codes.

For example, the FTP EXIT return code 16550 indicates the following:

Notes:

1. If data set DUMMY does not exist, FTP exits with a return code.
2. When using the REXX stack, ADDRESS LINKMVS must be used to invoke the FTP command in the z/OS UNIX environment.

Figure 9. How FTP subcommands can be issued from an EXEC
The GET command failed.

550 The reply code from the FTP server. The requested action was not taken; the file was not found or could not be accessed.

The FTP EXIT return code 4532 indicates the following:

4 The APPEND command failed.

532 The reply code from the FTP server. Need an account for storing files.

Note that 00 and 000 are valid values for $yy$ and $xxx$. This means that the error occurred at a time when no FTP subcommand was being processed $yy=00$ or at a time when no REPLY had been received from the server for the current process. Message EZA1735I Error code is the internal FTP error code. Use this value and Table 15 on page 100 to find the cause of the error.

FTP subcommand codes

Table 14 lists the valid FTP subcommand codes. The EXIT_IF_ERROR column specifies whether an error causes FTP to end if you specified the EXIT parameter on the FTP command.

<table>
<thead>
<tr>
<th>Code number</th>
<th>Subcommand</th>
<th>EXIT_IF_ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>No subcommand selected</td>
<td>Determined by internal FTP CLIENT ERROR CODE</td>
</tr>
<tr>
<td>1</td>
<td>AMBIGUOUS</td>
<td>true</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>ACCOUNT</td>
<td>true</td>
</tr>
<tr>
<td>4</td>
<td>APPEND</td>
<td>true</td>
</tr>
<tr>
<td>5</td>
<td>ASCII</td>
<td>true</td>
</tr>
<tr>
<td>6</td>
<td>BINARY</td>
<td>true</td>
</tr>
<tr>
<td>7</td>
<td>CD</td>
<td>true</td>
</tr>
<tr>
<td>8</td>
<td>CLOSE</td>
<td>true</td>
</tr>
<tr>
<td>9</td>
<td>TSO</td>
<td>true</td>
</tr>
<tr>
<td>10</td>
<td>OPEN</td>
<td>true</td>
</tr>
<tr>
<td>11</td>
<td>DEBUG</td>
<td>false</td>
</tr>
<tr>
<td>12</td>
<td>DELIMIT</td>
<td>false</td>
</tr>
<tr>
<td>13</td>
<td>DELETE</td>
<td>true</td>
</tr>
<tr>
<td>14</td>
<td>DIR</td>
<td>true</td>
</tr>
<tr>
<td>15</td>
<td>EBCDIC</td>
<td>true</td>
</tr>
<tr>
<td>16</td>
<td>GET</td>
<td>true</td>
</tr>
<tr>
<td>17</td>
<td>HELP</td>
<td>false</td>
</tr>
<tr>
<td>18</td>
<td>LOCSTAT</td>
<td>false</td>
</tr>
<tr>
<td>19</td>
<td>USER</td>
<td>true</td>
</tr>
<tr>
<td>20</td>
<td>LS</td>
<td>true</td>
</tr>
<tr>
<td>21</td>
<td>MDELETE</td>
<td>true</td>
</tr>
<tr>
<td>22</td>
<td>MGET</td>
<td>true</td>
</tr>
<tr>
<td>Code number</td>
<td>Subcommand</td>
<td>EXIT_IF_ERROR</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>---------------</td>
</tr>
<tr>
<td>23</td>
<td>MODE</td>
<td>true</td>
</tr>
<tr>
<td>24</td>
<td>MPUT</td>
<td>true</td>
</tr>
<tr>
<td>25</td>
<td>NOOP</td>
<td>true</td>
</tr>
<tr>
<td>26</td>
<td>PASS</td>
<td>true</td>
</tr>
<tr>
<td>27</td>
<td>PUT</td>
<td>true</td>
</tr>
<tr>
<td>28</td>
<td>PWD</td>
<td>true</td>
</tr>
<tr>
<td>29</td>
<td>QUIT</td>
<td>true</td>
</tr>
<tr>
<td>30</td>
<td>QUOTE</td>
<td>true</td>
</tr>
<tr>
<td>31</td>
<td>RENAME</td>
<td>true</td>
</tr>
<tr>
<td>32</td>
<td>SENDPORT</td>
<td>true</td>
</tr>
<tr>
<td>33</td>
<td>SENDSITE</td>
<td>false</td>
</tr>
<tr>
<td>34</td>
<td>SITE</td>
<td>false</td>
</tr>
<tr>
<td>35</td>
<td>STATUS</td>
<td>true</td>
</tr>
<tr>
<td>36</td>
<td>STRUCTURE</td>
<td>true</td>
</tr>
<tr>
<td>37</td>
<td>SUNIQUE</td>
<td>true</td>
</tr>
<tr>
<td>38</td>
<td>SYSTEM</td>
<td>true</td>
</tr>
<tr>
<td>39</td>
<td>TRACE</td>
<td>false</td>
</tr>
<tr>
<td>40</td>
<td>TYPE</td>
<td>true</td>
</tr>
<tr>
<td>41</td>
<td>LCD</td>
<td>true</td>
</tr>
<tr>
<td>42</td>
<td>LOCSITE</td>
<td>true (see Note on page 100)</td>
</tr>
<tr>
<td>43</td>
<td>LPWD</td>
<td>true</td>
</tr>
<tr>
<td>44</td>
<td>MKDIR</td>
<td>true</td>
</tr>
<tr>
<td>45</td>
<td>LMKDIR</td>
<td>true</td>
</tr>
<tr>
<td>46</td>
<td>EUCKANJI</td>
<td>true</td>
</tr>
<tr>
<td>47</td>
<td>IBMKANJI</td>
<td>true</td>
</tr>
<tr>
<td>48</td>
<td>JIS78KJ</td>
<td>true</td>
</tr>
<tr>
<td>49</td>
<td>JIS83KJ</td>
<td>true</td>
</tr>
<tr>
<td>50</td>
<td>SJISKANJI</td>
<td>true</td>
</tr>
<tr>
<td>51</td>
<td>CDUP</td>
<td>true</td>
</tr>
<tr>
<td>52</td>
<td>RMDIR</td>
<td>true</td>
</tr>
<tr>
<td>53</td>
<td>HANGEUL</td>
<td>true</td>
</tr>
<tr>
<td>54</td>
<td>KSC5601</td>
<td>true</td>
</tr>
<tr>
<td>55</td>
<td>TCHINESE</td>
<td>true</td>
</tr>
<tr>
<td>56</td>
<td>RESTART</td>
<td>false</td>
</tr>
<tr>
<td>57</td>
<td>BIG5</td>
<td>true</td>
</tr>
<tr>
<td>58</td>
<td>BLOCK</td>
<td>true</td>
</tr>
<tr>
<td>59</td>
<td>COMPRESS</td>
<td>true</td>
</tr>
<tr>
<td>60</td>
<td>FILE</td>
<td>true</td>
</tr>
<tr>
<td>61</td>
<td>PROXY</td>
<td>true</td>
</tr>
<tr>
<td>62</td>
<td>RECORD</td>
<td>true</td>
</tr>
</tbody>
</table>
Table 14. FTP subcommand codes (continued)

<table>
<thead>
<tr>
<th>Code number</th>
<th>Subcommand</th>
<th>EXIT_IF_ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>SCHINESE</td>
<td>true</td>
</tr>
<tr>
<td>64</td>
<td>STREAM</td>
<td>true</td>
</tr>
<tr>
<td>65</td>
<td>GLOB</td>
<td>false</td>
</tr>
<tr>
<td>66</td>
<td>PROMPT</td>
<td>false</td>
</tr>
<tr>
<td>67</td>
<td>UCS2</td>
<td>true</td>
</tr>
<tr>
<td>68</td>
<td>!</td>
<td>true</td>
</tr>
<tr>
<td>70</td>
<td>DUMP</td>
<td>false</td>
</tr>
<tr>
<td>71</td>
<td>VERBOSE</td>
<td>false</td>
</tr>
<tr>
<td>72</td>
<td>CLEAR</td>
<td>true</td>
</tr>
<tr>
<td>73</td>
<td>CPROTECT</td>
<td>true</td>
</tr>
<tr>
<td>74</td>
<td>PRIVATE</td>
<td>true</td>
</tr>
<tr>
<td>75</td>
<td>PROTECT</td>
<td>true</td>
</tr>
<tr>
<td>76</td>
<td>SAFE</td>
<td>true</td>
</tr>
<tr>
<td>77</td>
<td>CCC</td>
<td>true</td>
</tr>
<tr>
<td>78</td>
<td>LANGUAGE</td>
<td>true</td>
</tr>
<tr>
<td>79</td>
<td>FEATURE</td>
<td>true</td>
</tr>
<tr>
<td>80</td>
<td>SRESTART</td>
<td>true</td>
</tr>
<tr>
<td>99</td>
<td>UNKNOWN</td>
<td>true</td>
</tr>
</tbody>
</table>

Note: LOCSITE will only EXIT_IF_ERROR when there are no parameters on the command.

FTP reply codes

When an FTP command is sent to the foreign host, TCP/IP displays the sequence of subcommands, if any, that you entered. In addition, the response from the FTP server is also displayed as a reply code. These replies ensure the synchronization of requests and actions during data transfer, and guarantee that the client always knows the state of the foreign host FTP server.

For a complete list of FTPD reply codes see the [z/OS Communications Server: IP and SNA Codes](#).

FTP client error codes

Table 15 lists error codes used as condition codes for batch jobs if (EXIT was specified or as returned codes from REXX execs, whenever the FTP client detects one of the described errors and the FTPDATA statement CLIENTERRCODES is TRUE. These codes are always displayed in message EZA1735I Error codes.

Table 15. Internal error codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>FTP_INTERNAL_ERROR</td>
</tr>
<tr>
<td>02</td>
<td>FTP_SERVER_ERROR</td>
</tr>
<tr>
<td>03</td>
<td>FTP_CATALOG_NOT_FOUND</td>
</tr>
<tr>
<td>04</td>
<td>FTP_INVALID_PARAM</td>
</tr>
</tbody>
</table>
Table 15. Internal error codes (continued)

<table>
<thead>
<tr>
<th>Code</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>FTP_OPEN_IOSSTREAM_FAILED</td>
</tr>
<tr>
<td>06</td>
<td>FTP_ALREADY_CONNECTED</td>
</tr>
<tr>
<td>07</td>
<td>FTP_USAGE</td>
</tr>
<tr>
<td>08</td>
<td>FTP_CONNECT_FAILED</td>
</tr>
<tr>
<td>09</td>
<td>FTP_TIMEOUT</td>
</tr>
<tr>
<td>10</td>
<td>FTP_SESSION_ERROR</td>
</tr>
<tr>
<td>11</td>
<td>FTP_LOGIN_FAILED</td>
</tr>
<tr>
<td>12</td>
<td>FTP_INPUT_ERR</td>
</tr>
<tr>
<td>13</td>
<td>FTP_INPUT_EOF</td>
</tr>
<tr>
<td>14</td>
<td>FTP_NOTFOUND</td>
</tr>
<tr>
<td>15</td>
<td>FTP_INVALID_ENVIRONMENT</td>
</tr>
<tr>
<td>16</td>
<td>FTP_NOTENABLED</td>
</tr>
</tbody>
</table>

For example, the internal error code 13 indicates that the internal error FTP_INPUT_EOF occurred, which means that there was an unexpected end-of-file in the input stream.

**Restarting a failed data transfer**

FTP supports two subcommands for restarting data transfers that fail. Select the appropriate subcommand based on the data transfer mode at the time the data transfer failed.

- If stream mode data transfer was used, use the **srestart** subcommand to restart the data transfer. See "SRestart subcommand—Restart a stream data transfer" on page 265 for a description of stream mode restart.
- If block or compressed mode was used, use the **restart** subcommand to restart the data transfer. See "REStart subcommand—Restart a checkpointed data transfer" on page 233 for a description of block or compressed mode restart.

To restart a block or compressed mode data transfer, checkpointing must have been initiated prior to the start of the transfer. You can use either the LOCSITE or SITE subcommands to initiate checkpointing.

- The LOCSITE subcommand parameter CHKPTINT controls the checkpoint interval at the client, and the parameters CHKPTPREFIX and RESTGET control the naming of the client's checkpoint data set.
- The SITE subcommand parameter CHKPTINT controls the checkpoint interval at the server.

Each of the LOCSITE/SITE parameters is also supported in the FTP.DATA file.

**Interfacing with JES**

The MVS Job Entry System (JES) enables you to perform the following functions:

- Submit jobs (consisting of JCL and data) to the job scheduler for execution
- Spool JCL messages and SYSOUT during execution
- Print the output
- View the output
FTP server provides the following functions in its JES interface:

- Submitting a job
- Displaying the status of all the user's jobs
- Receiving the spool output of the job (JCL messages and SYSOUT)
- Deleting a job
- Submitting a job and automatically receiving output
- Terminating access to JES

Steps for submitting a job

A job consists of job control language (JCL) and data.

Perform the following steps to submit a job using FTP.

1. Create the JCL and data that you want to submit, using the editor on your client.
   - If the FTP server is set up for JESINTERFACELEVEL 1, to be able to display the status, receive spool output for, and delete a job, the job name in the JCL must be the USERIDx, where x is a 1-character letter or number and USERID must be the user ID you use to log into the FTP server to submit the job. Otherwise if the job name in the JCL is not USERIDx, the job may be submitted but the DIR subcommand will not display the job, and the GET and DELETE subcommands will not be supported for the job.
   - If the FTP server is set up for JESINTERFACELEVEL 2, the job name can be any name you are authorized to view using the Security Authorization Facility (SAF), such as RACF. See "JESINTERFACELEVEL differences" on page 107 for more information on security and JESINTERFACELEVEL 2.

You can determine how the FTP server is set up and whether you have authority to view jobs by entering the STAT client command and the SITE command. For JESINTERFACELEVEL 2, the STAT command returns:

211-JESINTERFACELEVEL is 2
211-JESOWNER is USER1
211-JESJOBNAME is USER1*

If the SITE JESOWNER= completes successfully, then the user has SAF authority to other users' jobs. If the SITE JESJOBNAME= completes successfully, then the user has SAF authority to other jobs.

Note: The maximum LRecl for the submitted job is 254 characters. JES scans only the first 72 characters of JCL.

2. Start a session with the FTP server on the MVS system to which you want to submit the job.

3. After you have entered your user ID and password, specify that you want to interface to JES with a site parameter by entering the following:
   SITE FILEtype=JES

4. To submit the JCL file you have created, enter the following:
   PUT filename.filetype
When you are done, the JCL is then submitted to the JES internal reader and waits for an initiator to start the job. The job is submitted under the user ID that you used when you logged on to the system unless a different user ID is specified on the JOB card.

The default for filetype is SEQ, and when you want to go back to normal FTP file transfer mode, enter the following:

```
SITE FILEtype=SEQ
```

Refer to the z/OS MVS JCL Reference for more information about using JCL.

Displaying the status of a job

This section describes client operation when the MVS server has been placed in FILEtype=JES mode with the SITE command. After you have submitted your job, you can determine whether it is waiting for execution, running, or finished. The status of all the jobs that are on the JES spool for your user ID can be displayed. A display for the DIR subcommand with JESINTERFACELEVEL=1 might look like the following:

```
MYUSRIDA JOB05444 OUTPUT 3 spool Files
MYUSRIDB JOB05766 OUTPUT 6 spool Files
MYUSRIDC JOB05832 OUTPUT 6 spool Files
MYUSRIDD JOB05946 ACTIVE
MYUSRIDE JOB06021 INPUT
```

The first column displays the job name. The second column displays the job ID, assigned by JES. This 8-character job ID, consisting of the word JOB followed by a 5-digit number or the letter J followed by a 7-digit number, is assigned by JES to identify your job. The fourth column displays the status of the job. The fifth column displays the class of the job.

The following is a description of each status:

**Status** | **Description**
---|---
INPUT | The job was received, but not run yet.
HELD | The JCL specified that the job is to be put on hold.
ACTIVE | The job is running.
OUTPUT | The job has finished and has output to be printed or retrieved. For each OUTPUT job, there are spool files that consist of JCL messages, JES messages, initiator and terminator messages, and SYSOUT. For jobs with a status of OUTPUT, the number of spool files for each job is specified in the DIR display output.

**Note:** The LS subcommand gives the same results as the DIR subcommand, but does not provide the number of spool files. Providing this spool information consumes a lot of computer resources. The server provides the job ID only
as the result of an LS subcommand in order to support an MGET subcommand. Use the LS subcommand rather than the DIR subcommand when possible.

The following is a sample of the DIR command with JESINTERFACELEVEL=2.

```
dir  
EZA1701I >>> PORT 127,0,0,1,4,5 
200 Port request OK.  
EZA1701I >>> LIST 
125 List started OK for JESJOBNAME=USER1+, JESSTATUS=ALL and JESOWNER=USER  
EZA2284I JOBNAME JOBID OWNER STATUS CLASS  
EZA2284I USER1A JOB000083 USER1 OUTPUT A ABEND=806 3 spool files  
EZA2284I USER1J JOB000029 USER1 OUTPUT A (JCL error) 3 spool files  
EZA2284I USER1R JOB000028 USER1 OUTPUT A RC=000 5 spool files  
EZA2284I USER1S JOB000027 USER1 OUTPUT A RC=000 5 spool files  
EZA2284I USER1A JOB000026 USER1 OUTPUT A RC=000 5 spool files  
EZA2284I USER1 TSU00024 USER1 ACTIVE TSU  
EZA2284I USER1W JOB000081 USER1 INPUT A -DUP-  
250-JESENTRYLIMIT of 10 reached. Additional entries not displayed  
250 List completed successfully.  
EZA1460I Command:  
dir tsu00024  
EZA1701I >>> PORT 127,0,0,1,4,6 
200 Port request OK.  
EZA1701I >>> LIST tsu00024 
125 List started OK for JESJOBNAME=USER1+, JESSTATUS=ALL and JESOWNER=USER  
EZA2284I JOBNAME JOBID OWNER STATUS CLASS  
EZA2284I USER1 TSU00024 USER1 ACTIVE TSU  
--------  
EZA2284I STEPNAME ++++++++ PROCNAME OS390R5  
EZA2284I CPU TIME 0.503 ELAPSED TIME 3853.718  
250 List completed successfully.  
EZA1460I Command:  
dir job54  
EZA1701I >>> PORT 127,0,0,1,4,7 
200 Port request OK.  
EZA1701I >>> LIST job54 
125 List started OK for JESJOBNAME=USER1+, JESSTATUS=ALL and JESOWNER=USER  
EZA2284I JOBNAME JOBID OWNER STATUS CLASS  
EZA2284I USER1 JOB00054 USER1 OUTPUT A RC=000 5 spool files  
EZA2284I --------  
EZA2284I ID STEPNAME PROCSTEP C DDNAME BYTE-COUNT  
EZA2284I 001 JESE H JESMSGLG 1200  
EZA2284I 002 JESE H JESJCL 526  
EZA2284I 003 JESE H JESYMSG 1255  
EZA2284I 004 STEPS H SYSPRT 741  
EZA2284I 005 STEPS A SYSPRT 209  
EZA2284I 5 spool files  
250 List completed successfully.  
EZA1460I Command:  
```

Receiving spool output

You may retrieve JCL messages, JES messages, initiator and terminator messages, and SYSOUT data sets either individually or as a group.

Receiving individual spool files

Retrieving the spool files one at a time enables you to see whether a job ran correctly before you retrieve the rest of the output, giving you greater control over retrieving job information. If your FTP server is configured with JESINTERFACELEVEL=2, then a DIR command will display the output job's
To retrieve the spool output, use the FTP GET subcommand while in the FILEtype=JES mode, specifying the job ID and the number of the spool file you want.

You can specify a short form of the job ID by entering the letter J followed by a 4- or 5-digit job number. For example:

```
GET JOB05444.1 JOB05444.FILE1 (REPLACE
GET JOB05766.6 ASSEMBLY.FILE6
GET JOB06235.2 (REPLACE
GET JOB00275.4
GET J7438.3
```

In these examples, foreign_file is specified first, followed by local_file (on your client machine) with the appropriate options, such as REPLACE. The first example requests that the first spool file for JOB05444 be transmitted and replace the file on your client named JOB05444.FILE1. The second command requests that the sixth spool file for JOB05766 be transmitted to your client with the name ASSEMBLY.FILE6.

If you have specified FILEtype=JES, you can use the MGET subcommand to receive output from multiple jobs without specifying them one at a time. For example, you can enter:

```
MGET parameter
```

The FTP client requires an MGET subcommand parameter. The parameter is passed to the FTP server but is not used. The server returns all of the SYSOUT files for all the jobs in the HELD queue for your user ID for FTP servers configured for JESINTERFACELEVEL=1. For JESINTERFACELEVEL=2, all jobs that match the filters JESJOBNAME, JESOWNER and JESSTATUS are retrieved whether they are in the HELD queue or not. Note that the JESJOBNAME and JESOWNER SITE parameters allow wildcards (*,?). When using the MGET command the JESOWNER value should be OUTPUT.

**Notes:**

1. On an MVS FTP server, local_file must be specified.
2. Truncation can cause a loss of data.
3. A GET command performed on an empty data set erases the contents of the existing local data set.
4. Receiving the output of a job does not remove the job output from the queue. To remove the job output from the queue, you must issue a DELETE command.

**Receiving a group of spool files**

To retrieve all the spool files associated with the same job simultaneously into the same destination file, specify:

```
GET jobid.x
```

where x can be uppercase or lowercase. All the spool files are transferred together and put into file jobid.x. The following line appears between each retrieved JES spool file:

```
!! END OF JES SPOOL FILE !!```
This enables you to easily find the end of each spool file.

You can also specify a data set name to send the files to, such as:

```
GET jobid.x data_set_name
```

All the spool files are put into the file named file.name. This eliminates the need to retrieve each spool file separately. For example, GET J3456.X retrieves all the spool files for JOB03456 and puts them in a file named J3456.X.

The command MGET with any parameter produces the same results as issuing GET jobid.x commands for each job that is associated with your user ID.

**Notes:**
1. In JES2 (if JESINTERFACELEVEL=1), the spool files retrieved by GET and tallied by DIR must be in a hold queue (commonly class=H).
2. In JES3 (if JESINTERFACELEVEL=1), the spool files must be in a hold queue reserved for external writers. Ask your system programmer for the class that says (HOLD=EXTWTR) in the JES3 installation stream.
3. The maximum record length that can be received at the server is 254 characters before the record is truncated.
4. Receiving the output of a job does not remove the job output from the queue.
   To remove the job output from the queue, you must issue a DELETE command.

### Deleting a job

You can delete a job before or during execution, or you can delete the output of a job before you have retrieved it. You do this by using the DELETE subcommand while in the FILEtype=JES mode and the job ID. You can specify either the 8-character job ID or a short form of the job ID by entering the letter J followed by a 1- to 7-digit job number. For example,

```
DELETE JOB05444
DELETE J3672
```

When you issue the DELETE command, all spool output related to a job is deleted.

The host returns the message CANCEL SUCCESSFUL after it deletes the job.

### Steps for submitting a job and automatically receiving output

You can submit a job using FTP and automatically receive your output. Rather than using the JCL you built on the FTP client, this function uses the JCL you have built on the FTP server. Automatic retrieval of jobs works only if the file contains one job. It will not work for files with more than one job (multiple JOB cards) in it.

Perform the following steps to submit a job using FTP.

1. Create the JCL and data that you want to submit and save it on the MVS host where the FTP server resides.
   The JCL can reside in a sequential data set, or partitioned data set, or hfs file. If JESINTERFACELEVEL=1, then the job name in the JCL must be USERIDx,
   where x is a 1-character letter or number. Additionally, the output class for any data sets you wish to retrieve (MSGCLASS or SYSOUT files) contained in your JCL must specify a JES HOLD output class. If JESINTERFACELEVEL=2, then the JESJOBNAME and JESJOBOWNER must match the jobname and jobowner. Additionally, JESSTATUS must be set to ALL or OUTPUT and the logged in FTP user ID must have access to nodeid.userid.jobname.jobid.
2. Start a session with the FTP server on the MVS system to which you want to submit the job.

3. After you have entered your user ID and password, specify that you want to interface to JES with a site parameter by entering the following:
   
   SITE FILEtype=JES

4. Submit the JCL file you have created by entering the following command:

   GET jclfilename.jclfiletype outputfilename.outputfiletype

   The **outputfilename.outputfiletype** defines the data set at the FTP client site that is to contain the HELD job output when the job completes.

   The MVS FTP server reads the data set **jclfilename.jclfiletype** and submits it to the JES internal reader.

   You know you are done when you see the following replies. The next two replies indicate that the job has been submitted and the MVS FTP server is waiting for it to complete. The third reply indicates that the job has finished and the files have been copied to your output data set:

   125 Submitting job outputfilename.outputfiletype FIXrecfm 80
   125 When JOB05125 is done, will retrieve its output
   250 Transfer completed successfully.

   **Note:** When submitting a job and automatically receiving the output, remember that your session is suspended. You should use care, based on the anticipated run time of your job, when using this function. If your session times out, you must restart FTP and manually retrieve your output. The timeout is determined by the value of JESPUTGETTO in the server FTP.DATA data set. If no value is specified in FTP.DATA, the default value is 10 minutes.

**Terminating access to JES**

The FTP default for FILEtype is SEQ. When you want to end access to JES and return to FTP in its normal file transfer mode, specify the following:

   SITE FILEtype=SEQ

**JESINTERFACELEVEL differences**

The FTP JESINTERFACELevel 2 allows increased functionality for users over that available with JESINTERFACELevel 1. Jobs allowable by the system (JESSPOOL RACF class) whether they are in held or nonheld output class can be viewed. Information about jobs such as held/dup for jobs on the internal reader, CPU time of running jobs, and number of sysout data sets for completed jobs can be displayed.

If the user does a DIR subcommand without any operands, the display shows matches of the search criteria — one line per task. If the user does a DIR subcommand for a specific JOBID, the status of the job is listed and, if the status is OUTPUT, it lists information about all the sysout data sets created by this job (and available in the JES spool data set at the time of access). If a JOBID is supplied and the job is active, it lists information about the job’s step name, cpu busy, and elapsed time.
Whenever a DIR subcommand is executed (with or without arguments), the JESJOBNAME, JESENTRYLIMIT, JESOWNER and JESSTATUS filter keywords can be used to limit the number of entries that are returned to the user. A DIR subcommand with all JES keywords set to their default and JESOWNER set to USER1 returns information about all batch jobs that have USER1 registered as OWNER. If the value of JESOWNER is set to empty (space), it defaults to the logged-in user ID.

- To list jobs for all users, a value of JESOWNER=* must be specified. If the value of JESJOBNAME is set to empty (space), it defaults to the logged-in user ID suffixed with an asterisk (if user is USER1, then JESJOBNAME defaults to USER1*).
- To list all job names, a value of JESJOBNAME=* must be specified. If the value of JESSTATUS is set to empty, it defaults to all types of output.
- To list only the completed jobs, a value of JESSTATUS=OUTPUT must be specified.

The subcommands DIR, LS, MDELETE, and MGET allow for wildcard (*) filtering on a specific jobid only if JESINTERFACELevel 2 is active. A DIR subcommand results in the use of the LIST FTP command. An LS, MDELETE, or MGET subcommand results in the use of the NLST FTP command.

The following table summarizes the behavior of the different levels of JESINTERFACELevel. The term matching filters means all jobs that match parameters of JESJOBNAME, JESOWNER, JESSTATUS, and up to the JESENTRYLIMIT entries.

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>JESINTERFACELevel 1</th>
<th>JESINTERFACELevel 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR [*]</td>
<td>All jobs</td>
<td>All Jobs matching filters</td>
</tr>
<tr>
<td>DIR Jxx</td>
<td>All jobs (Jxx ignored)</td>
<td>Details for Jxx (*)</td>
</tr>
<tr>
<td>LS</td>
<td>All jobids</td>
<td>All jobids matching filters</td>
</tr>
<tr>
<td>LS Jxx with wildcards</td>
<td>All jobids</td>
<td>All Jobs matching filters (*)</td>
</tr>
<tr>
<td>LS Jxx no wildcards</td>
<td>All jobids (Jxx ignored)</td>
<td>All Jxx.nnn sysout ds names matching filters (*)</td>
</tr>
<tr>
<td>GET Jxx.1 [local-file]</td>
<td>Return single sysout file</td>
<td>Return single sysout file</td>
</tr>
<tr>
<td>GET Jxx.x [local-file]</td>
<td>All Jxx sysout files in one file with separators</td>
<td>All Jxx sysout files in one file with separators</td>
</tr>
<tr>
<td>MGET *</td>
<td>One file per job - each file with separators - local file names default to jobids</td>
<td>One file per job matching filters - each file with separators - local files names default to jobids</td>
</tr>
<tr>
<td>MGET jxx</td>
<td>As MGET * (Jxx ignored)</td>
<td>One file per job matching filters - each file with separators - local file names default to Jxx.nnn (*)</td>
</tr>
<tr>
<td>DELETE Jxx</td>
<td>Jxx deleted</td>
<td>Jxx Deleted</td>
</tr>
<tr>
<td>DELETE Jxx.1</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>MDELETE *</td>
<td>All jobs deleted</td>
<td>All jobs matching filters deleted</td>
</tr>
<tr>
<td>GET jclds outfile</td>
<td>Server jclds submitted, all output returned in outfile with separators</td>
<td>Server jclds submitted, all output returned in outfile with separators</td>
</tr>
</tbody>
</table>
JESINTERFACELevel 2 returns different entries from JESINTERFACELevel 1 with wildcard (*) filtering, assuming the default values are used for the JESOWNER, JESJOBNAME, and JESENTRYLimit plus JESSTATUS=OUTPUT. The format of the returned data might also be different.

FILETYPE=JES parameters on DIR, LS and MGET subcommands allow an asterisk (*) for all job IDs, or a specific job ID. If a job ID is specified, but no such job ID exists, an error reply will be returned to the client.

The NLST command is used under the covers for MGET and MDELETE to obtain lists of the resource names to issue individual GET or DELETE subcommands against. For job IDs, the list is simply a list of JES job IDs. For sysout data set IDs, the list is a list of JES job IDs suffixed with sysout data set numbers — in the same syntax that is supported by the GET and DELETE subcommands.

LIST, NLST, MDELETE, and MGET all show different behaviors for JESINTERFACELevel 2 over JESINTERFACELevel 1. This is because LIST and NLST use the filters for JESINTERFACELevel 2, and the commands DIR, LS, MDELETE, and MGET use LIST and NLST.

- For job IDs, an NLST command will return:
  - JOB000013
  - JOB000034
  - STC00067

- For sysout data set IDs, an NLST command will return:
  - JOB000013.1
  - JOB000013.2
  - JOB000013.3
  - JOB000013.4

**JES security**

If FTP.DATA does not change the JESINTERFACELevel to 2, the FTP server will behave as in previous releases. FTP clients will be allowed to submit jobs to JES, retrieve held output that matches their logged in user ID plus one character, and delete held jobs that match their logged in user ID plus one character.

If JESINTERFACELevel is set to 2, then FTP clients have the ability to retrieve and delete any job in the system permitted by the Security Access Facility (SAF) resource class JESSPOOL. For that reason, JESINTERFACELevel=2 should only be specified if the proper JES and SDSF security measures are in place to protect access to JES output. The SAF controls used for JESINTERFACELevel=2 are essentially a subset of those used by SDSF. Therefore, if an installation has customized SAF facilities for SDSF, then they are configured for FTP JES level 2.

Before customizing the FTP-to-JES interface, you must complete JES customization. For example, JESJOBS is an SAF class that controls which users can submit jobs to JES. JESSPOOL is the SAF that controls which users can access output jobs. Customize these SAF classes before beginning customization of the FTP-to-JES interface.

JESSPOOL defines resource names as <nodeid>, <userid>, <jobname>, <Dsid>, <dsname>. An FTP client can delete an output job if it has ALTER access to the resource that matches its nodeid, userid, and job name. If the FTP client has UPDATE access to the resource, it can list, retrieve, or GET the job output. (JESINTERFACELevel 2 uses the SAPI interface to JES, so UPDATE authority is required to list job status or retrieve job output.) Refer to z/OS JES2 Initialization
Changing JESSTATUS, JESOWNER, and JESJOBNAME

There are three filters used by the FTP server to control the display of jobs:

- JESSTATUS
- JESOWNER
- JESJOBNAME

JESSTATUS can be changed by an FTP client via the SITE command to filter jobs in INPUT, ACTIVE, or OUTPUT state. The SDSF resources checked for these states are `ISFCMD.DSP.INPUT.jesx`, `ISFCMD.DSP.ACTIVE.jesx`, and `ISFCMD.DSP.OUTPUT.jesx`, respectively. At login time (USER command), the default value is set to ALL if READ access is allowed to all three classes. Otherwise, it attempts to set it to OUTPUT, ACTIVE, and then INPUT if the appropriate READ access is allowed. If no READ access is allowed to any of the classes, JESSTATUS is set to OUTPUT but JESOWNER and JESJOBNAME cannot be changed from the default. In this way, SAF controls can be put in place to limit FTP clients to whatever status of jobs an installation requires.

At login time JESOWNER will have the value of the logged in user ID. Authority to change JESOWNER is obtained by READ access to RACF profiles `ISFCMD.FILTER.OWNER`. An FTP client who has READ access to `ISFCMD.FILTER.OWNER` will be allowed to change the JESOWNER parameter using the SITE command.

At login time, JESJOBNAME will have the value of the logged in user ID plus an asterisk (*). Authority to change JESJOBNAME is obtained by READ access to RACF profile `ISFCMD.FILTER.PREFIX`. An FTP client who has READ access to `ISFCMD.FILTER.PREFIX` will be allowed to change the JESJOBNAME parameter using the SITE command.

If a user is not authorized to the appropriate `ISFCMD.DSP.<status>.jesx`, any SITE JESSTATUS command is rejected with a reply:

```
200 User xxxxxxxx is not authorized to filter on JESSTATUS, JESSTATUS remains xxxxxxxx
```

If a user is not authorized to filter on JESOWNER, any SITE JESOWNER command is rejected with a reply:

```
200 User xxxxxxxx is not authorized to filter on JESOWNER, JESOWNER remains xxxxxxxx
```

If a user is not authorized to filter on JESJOBNAME, any SITE JESJOBNAME command is rejected with a reply:

```
200 User xxxxxxxx is not authorized to filter on JESJOBNAME, JESJOBNAME remains xxxxxxxx*
```

Displaying the status of jobs (LIST and NLST)

No security checks are made when returning information to the user on an NLST or LIST (DIR, MGET, or MDELETE) command that has a wild card character of * or ? for a jobid. The filter variables were reset when the user connection was made. Any attempts to change them caused the appropriate security check to be made.

When a user issues a DIR command for a specific jobid a check for READ access to JESSPOOL resource `nodeid.userid.jobname.jobid` is made. This check is
performed to prevent a different user (one who happens to know a jobid of another user's job) from obtaining status information, even though the user was not allowed to filter on OWNER. If a user is not allowed READ access, the following reply is sent:

**Browsing of SYSOUT data sets**
When a SYSOUT data set is requested for transfer to the FTP client, the server will check JESSPOOL resource nodeid.userid.jobname.jobid.Dsid.dsname for READ access. If a user is not allowed READ access to the resource, the following reply is sent:

**Deleting/purging of SYSOUT data sets**
When an FTP client requests to delete a SYSOUT data set, the server checks JESSPOOL resource nodeid.userid.jobname.jobid.Dsid.dsname for ALTER access. If a user is not allowed ALTER access to the resource, the following reply is sent:

**JES examples**
The following example shows the JCL file USER121.JCL.CNTL(SMFALL) being submitted to the JES. Before FTP commands are issued, only the data set USER121.FTP.EXAMPLE exists on MVSXA2.

```
User: ftp 9.67.113.24 621
System:
   IBM FTP CS V1R2
   FTP: using TCPCS
   FTP.DATA FILE NOT FOUND. USING HARDCODED DEFAULT VALUES.
   Connecting to 9.67.113.24, port 621
   220-FTPSERVE IBM FTP CS V1R2 at MVSIC03.TCP.RALEIGH.IBM.COM,
   19:03:
   on 1997-09-17
   220 Connection will close if idle for more than 5 minutes.
   NAME (<host>:tsouserid):
   User: user121
   System:
   >>>USER user121
   331 Send password please.
   Password:
   >>>PASS ********
   230 user121 is logged on. Working directory is "/u/user121".
   Command:

User: site file=jes
System:
   >>>SITE file=jes
   200 Site command was accepted
   Command:
   User: put 'user121.jcl.cntl(mvsjob)'
   System:
   >>>SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=27920
   200 Site command was accepted
   >>>PORT 9,67,112,25,4,37
   200 Port request OK.
   >>>STOR 'user121.jcl.cntl(mvsjob)'
   125 Sending Job to JES Internal Reader FIXrecfm 80
   250-It is known to JES as JOB02189.
   250 Transfer completed successfully.
   1066 bytes transferred in 3.118 seconds. Transfer rate 0.34 Kbytes/sec.
   Command:
```
User: dir
System: >>>PORT 9,67,112,25,4,38
200 Port request OK.
>>>LIST
125 List started OK.
USER121A JOB00067 INPUT
250 List completed successfully.
Command:

User: dir
System: >>>PORT 9,67,112,25,4,39
200 Port request OK.
>>>LIST
125 List started OK.
USER121A JOB00067 ACTIVE
250 List completed successfully.
Command:

User: dir
System: >>>PORT 9,67,112,25,4,40
200 Port request OK.
>>>LIST
125 List started OK.
USER121A JOB00067 OUTPUT 4 Spool Files
250 List completed successfully.
Command:

User: lcd 'user121.ftp.example.'
System: Local directory name set to USER121.FTP.EXAMPLE.
Command:

User: lpwd
System: Local directory is USER121.FTP.EXAMPLE.
Command:

User: dir
System: >>>PORT 9,67,112,25,4,41
200 Port request OK.
>>>LIST
125 List started OK.
USER121A JOB00067 OUTPUT 4 Spool Files
250 List completed successfully.
Command:

User: get job00067.x spoolall
System: >>>PORT 9,67,112,25,4,42
200 Port request OK.
>>>RETR job00067.x
125 Sending all SPOOL files for requested JOBID.
250 Transfer completed successfully.
Command:
User: get job00067.1 spool1
System:
   >>>PORT 9,67,112,25,4,43
   200 Port request OK.
   >>>RETR job00067.1
   125 Sending data set USER121.USER121A.JOB00067.D000002.JESMSGLG
   250 Transfer completed successfully.
   1962 bytes transferred in 0.739 seconds. Transfer rate 2.65 Kbytes/sec.
   Command:

User: get job00067.2 spool2
System:
   >>>PORT 9,67,112,25,4,44
   200 Port request OK.
   >>>RETR job00067.3
   125 Sending data set USER121.USER121A.JOB00067.D000003.JESYSMSG
   250 Transfer completed successfully.
   1982 bytes transferred in 2.123 seconds. Transfer rate 0.93 Kbytes/sec.
   Command:

User: get job00067.3 spool3
System:
   >>>PORT 9,67,112,25,4,45
   200 Port request OK.
   >>>RETR job00067.3
   125 Sending data set USER121.USER121A.JOB00067.D000004.JESYSMSG
   250 Transfer completed successfully.
   1982 bytes transferred in 2.123 seconds. Transfer rate 0.93 Kbytes/sec.
   Command:

User: get job00067.4 spool4
System:
   >>>PORT 9,67,112,25,4,46
   200 Port request OK.
   >>>RETR job00067.4
   125 Sending data set USER121.USER121A.JOB00067.D0000103.?
   250 Transfer completed successfully.
   1227 bytes transferred in 0.380 seconds. Transfer rate 3.23 Kbytes/sec.
   Command:
After executing the FTP commands, the following data sets now exist on MVSXA2:

USER121.FTP.EXAMPLE.SPOOLALL
USER121.FTP.EXAMPLE.SPOOL1
USER121.FTP.EXAMPLE.SPOOL2
USER121.FTP.EXAMPLE.SPOOL3
USER121.FTP.EXAMPLE.SPOOL4

Note: In most situations, the INPUT status is too fast to be captured by issuing DIR. However, if the ACTIVE or OUTPUT status of the job is captured, the INPUT status has been passed successfully.

The following are examples displayed by the shown DIR command.

DIR with JESJOBName=*, JESOwner=*, JESSTATUS=ALL JESENTRYLimit=200 shows all jobs in the system.
DIR

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>JOBID</th>
<th>OWNER</th>
<th>STATUS</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>TSU00017</td>
<td>USER1</td>
<td>INPUT</td>
<td>A</td>
</tr>
<tr>
<td>USER1A</td>
<td>JOB00001</td>
<td>USER1</td>
<td>INPUT</td>
<td>A -HELD-</td>
</tr>
<tr>
<td>USER1A</td>
<td>JOB00002</td>
<td>USER1</td>
<td>INPUT</td>
<td>A -DUP-</td>
</tr>
<tr>
<td>USER2B</td>
<td>JOB00022</td>
<td>USER2</td>
<td>ACTIVE</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>STC00002</td>
<td>USER2</td>
<td>STEP=</td>
<td>PROCM=</td>
</tr>
<tr>
<td>MASTER</td>
<td>+MASTER+</td>
<td>USER2</td>
<td>CPUT=</td>
<td>7.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER2</td>
<td>ELAPT=</td>
<td>7.27</td>
</tr>
<tr>
<td>USER3A</td>
<td>JOB00061</td>
<td>USER3</td>
<td>OUTPUT</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>USER4</td>
<td>USER4</td>
<td>OUTPUT</td>
<td>D</td>
</tr>
</tbody>
</table>

DIR with JESJOBName=* JESOwner=USER1 JESSTATUS=ALL JESENTRYLimit=200 shows all jobs owned by USER1

DIR

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>JOBID</th>
<th>OWNER</th>
<th>STATUS</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>TSU00017</td>
<td>USER1</td>
<td>INPUT</td>
<td>A</td>
</tr>
<tr>
<td>USER1A</td>
<td>JOB00001</td>
<td>USER1</td>
<td>INPUT</td>
<td>A -HELD-</td>
</tr>
<tr>
<td>USER1A</td>
<td>JOB00002</td>
<td>USER1</td>
<td>INPUT</td>
<td>A -DUP-</td>
</tr>
</tbody>
</table>

DIR with JESJOBName=* JESOwner=USER14 JESSTATUS=ALL JESENTRYLimit=200 shows all jobs owned by USER14

DIR 250 No tasks found for JESJOBName=*, JESSTATUS=ALL and JESOwner=USER14

DIR T* with JESJOBName=* JESOwner=USER1 JESSTATUS=ALL JESENTRYLimit=200 shows all TSO jobs.

DIR T*

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>JOBID</th>
<th>OWNER</th>
<th>STATUS</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>TSU00017</td>
<td>USER1</td>
<td>INPUT</td>
<td>A</td>
</tr>
</tbody>
</table>

DIR with JESJOBName=* JESOwner=* JESSTATUS=ALL JESENTRYLimit=5 shows the first 5 jobs in the system

DIR

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>JOBID</th>
<th>OWNER</th>
<th>STATUS</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>TSU00017</td>
<td>USER1</td>
<td>INPUT</td>
<td>A</td>
</tr>
<tr>
<td>USER1A</td>
<td>JOB00001</td>
<td>USER1</td>
<td>INPUT</td>
<td>A -HELD-</td>
</tr>
<tr>
<td>USER1A</td>
<td>JOB00002</td>
<td>USER1</td>
<td>INPUT</td>
<td>A -DUP-</td>
</tr>
<tr>
<td>USER2B</td>
<td>JOB00022</td>
<td>USER2</td>
<td>ACTIVE</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>STC00002</td>
<td>USER2</td>
<td>STEP=</td>
<td>PROCM=</td>
</tr>
<tr>
<td>MASTER</td>
<td>+MASTER+</td>
<td>USER2</td>
<td>CPUT=</td>
<td>7.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER2</td>
<td>ELAPT=</td>
<td>7.27</td>
</tr>
</tbody>
</table>

JESENTRYLIMIT of 5 reached. Additional entries not displayed

If JESINTERFACELevel is set to 2, the DIR or LIST command for a specific jobid allows you to display specific SYSOUT data sets.

DIR JOB00061 with JESJOBName=* JESOwner=* JESSTATUS=ALL JESENTRYLimit=5 displays SYSOUT data sets from JOB00061.

DIR JOB00061

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>JOBID</th>
<th>OWNER</th>
<th>STATUS</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER3A</td>
<td>JOB00061</td>
<td>USER3</td>
<td>OUTPUT</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>JOB00061</td>
<td>USER3</td>
<td>OUTPUT</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>JOB00061</td>
<td>USER3</td>
<td>OUTPUT</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>JOB00061</td>
<td>USER3</td>
<td>OUTPUT</td>
<td>D</td>
</tr>
</tbody>
</table>

Chapter 4. Transferring data using the File Transfer Protocol (FTP) 115
If JESINTERFACELevel is set to 2 then the GET command should produce results similar to the ones below ($\text{JESJOBName}=$* $\text{JESOwner}=$* $\text{JESSTATUS}=$ALL $\text{JESENTRYLimit}=200$ will display SYSOUT data sets from JOB00061).

```
<table>
<thead>
<tr>
<th>ID</th>
<th>STEPNAME</th>
<th>PROCSTEP</th>
<th>DDNAME</th>
<th>REC-COUNT</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>JESE</td>
<td>H</td>
<td>JESMSGLG</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>JESE</td>
<td>H</td>
<td>JESJCL</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>A</td>
<td>JESYSMSG</td>
<td></td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>
```

If JESINTERFACELevel is set to 2 then the MGET command should produce results similar to those below, which are identical to the GET of just the JOB00061.

At the end of each SYSOUT data set of a MGET, the following line will be stored:

```
!! END OF JES SPOOL FILE !!
```

If MGET is specified with an asterisk, it works the same as in JESINTERFACELevel 1. MGET will get all jobs that match the JES filters. Also, in the example below, there are two input jobs for user1, no active jobs, and three output jobs. Care should be taken with the JES filters because an MGET from the client actually appears to the server as an NLST followed by several GETs. For example, if JESSTATUS is set to ALL and there are many jobs returned in the NLST that are in ACTIVE or INPUT status, then the MGET will not retrieve the number of output files expected and there will be no message that "250 No tasks found for $\text{JESJOBName}=$...". ($\text{JESJOBName}=$* $\text{JESOwner}=$USER1 $\text{JESSTATUS}=$ALL $\text{JESENTRYLimit}=1024$.)
Performing DB2 SQL queries with FTP

FTP enables you to submit a Structured Query Language (SQL) SELECT query to the DB2 subsystem and receive the results of the SQL query. FTP can perform this function as either the server or the client.

For information on installing the SQL query function for the FTP client or server, refer to the z/OS Communications Server: IP Configuration Reference.

SQL data types supported by FTP

FTP access to SQL supports the following data types:

- DATE
- TIME
- TIMESTAMP
- VARCHAR (variable length, up to 254 characters)
• CHAR (fixed length, up to 254 characters)
• DECIMAL
• INTEGER (full word)
• SMALLINT (half word)
• FLOAT (single or double precision)
• LONG VARCHAR (VARCHAR(n), where n is greater than 254)
• GRAPHIC
• VARGRAPHIC
• LONG VARGRAPHIC

Mixed data (double-byte character set and single-byte character set) is supported in CHAR, VARCHAR, and LONG VARCHAR data types, but column alignment in the output file might not be maintained.

Creating the input data set

Before performing a DB2 SQL query using FTP, you must create an MVS data set that contains the SQL query you want to perform.

You can create queries on the client and use the FTP PUT command to send the queries to the MVS system to be processed. Or, you can prepare a group of SQL queries on the MVS system and perform them regularly.

Note: FTP can process only one SQL query per file.

For example, a data set on an MVS system named userid.SQL.IN contains the following SQL query:

SELECT LASTNAME, EMPID, YEARS_EMPLOYED FROM EMPLOYEE_TABLE WHERE YEARS_EMPLOYED > 25

You either created that data set on the MVS system with TSO, or you used the FTP PUT command to put the data set on the MVS system.

Setting the characteristics for the SQL query

After creating a data set to use for your query, you must log on to FTP and set the file type for the query:

SITE/LOCSITE FILEtype=SQL

There are several commands that are relevant to the client and server in SQL mode. The server commands use SITE, and the client commands use LOCSITE. The following list describes the commands:

SITE/LOCSITE DB2=

Specifies the name of the DB2 subsystem that you want to perform your queries. See "Specifying the DB2 subsystem to perform the query" on page 119 for more information about DB2 subsystems.

SITE/LOCSITE SPRead or NOSPRead

Specifies whether you want the output to be in spreadsheet or report format. See "Specifying the output format" on page 120 for more information about output format.

SITE/LOCSITE SQLCol=
Specifies whether you want the column headings to use the DB2 column names or labels. Valid values include Names, Labels, or Any. See "Assigning column headings for the SQL query result table" for more information about column headings.

To return to normal FTP processing after performing queries, or other processes, specify:

SITE/LOCSITE FILEtype=SEQ

On MVS systems, RECfm=VB is a recommended format that enables you to view the results of the SQL query. Issue the following command to specify that new data sets should be created with the RECfm=VB attribute:

SITE/LOCSITE RECfm=VB

To prevent the automatic sending of a SITE command that might override your SITE setting, toggle SENDSITE to OFF. For more information about the SENDSITE command, see "SENDSite subcommand—Toggle the sending of site information" on page 239.

Specifying the DB2 subsystem to perform the query

An MVS system can run several DB2 systems simultaneously, each known by a subsystem name of up to four characters. For example, you can have a DB2 test system called DB2T and a DB2 production system called DB2P.

FTP connects to a DB2 system to have it execute a DB2 query. You can specify what DB2 system FTP should connect to with the following SITE or LOCSITE parameter:

SITE/LOCSITE DB2=

For example, if you want the FTP server to have the DB2T system perform your queries, specify:

SITE DB2=DB2T

If you want the FTP client to have the DB2P system perform your queries, specify:

LOCSITE DB2=DB2P

The default DB2 system name is DB2. You can change the default with the DB2 parameter in the FTP.DATA data set. See "Changing local site defaults using FTP.DATA" on page 63 for more information about the FTP.DATA data set.

Assigning column headings for the SQL query result table

When you create a DB2 table, you can assign descriptive labels to the table columns. For example, a column name could be XCM554, but the label could be WEEKLY PAY. For information about assigning names and labels, refer to the DB2 DESCRIBE statement in the DB2 SQL Reference.

The SQLCol parameter of the SITE command enables you to specify whether you want names or labels to appear at the top of the columns in your output file. The default is Names. Issue the following command if you want a database column name to appear at the top of each column in your output file:

SITE/LOCSITE SQLCol=Names
Issue the following command if you want a label to appear at the top of each column:
SITE/LOCSITE SQLCol=Labels

If you specify the Labels parameter, and a column in your query does not have a label defined in the database, the FTP server supplies a column heading. For more information about column headings, see "FTP-supplied column headings".

Issue the following command if you want either a label or a name to appear at the top of each column:
SITE/LOCSITE SQLCol=Any

If you specify the Any parameter, the label appears as the column heading. However, if the column does not have a label, the name appears at the top of the column.

**FTP-supplied column headings**

The FTP client and server provide column headings in the result table when DB2 does not. This occurs when a result table contains expression columns or when labels are requested and a database column that appears in the result table does not have a label defined.

FTP builds a column heading for expression columns. For example,
Select employee, salary/52 from ABC.Staff

results in two columns. The first column gets its name from DB2, while the second column is built by the server. The server uses the heading COL002 for the second column because it supports the SQL limit of 750 columns.

**Specifying the output format**

You have two choices for the format of your output data set: spreadsheet format and report format. The default is NOSPRead (report format), but you can change the default for your FTP server by changing the FTP.DATA data set. See "Changing local site defaults using FTP.DATA" on page 63 for more information.

**Spreadsheet Format:** You can have the output of the SQL query formatted to load directly into a spreadsheet program running on a PC or a workstation. To get the spreadsheet format, issue the following command:
SITE SPRead or LOCSITE SPRead

The SPRead format option puts a TAB character before the first character of each column entry, except the first column. See your spreadsheet program documentation for instructions about how to import the output of the SQL query.

**Report Format:** The NOSPRead format option puts one or more blank spaces between the columns, and it lists the SQL query, the column headings, and the resulting columns. Each section is separated with horizontal dashed lines. An output data set in NOSPRead, or report, format is easier to view and print.

To get the report format, issue the following command:
SITE NOSPRead or LOCSITE NOSPRead

The following is an example of the results contained in the NOSPRead format of the SQL.OUTPUT data set.
A lowercase letter in the first position of each dashed line specifies what part of the output follows, enabling a program to read and interpret the contents. For example, `s` indicates that the SQL query follows, `h` indicates a header, `d` indicates that the rest of the data set is the actual data, and `e` indicates that an error message follows.

The width of the output data set depends on the width of the results from the DB2 query.

### Submitting the query

After you have created a data set that contains an SQL query, logged on to FTP, and set the appropriate SITE or LOCSITE parameters, you are ready to execute the contents of the data set. You can do this from either an FTP client or an FTP server.

#### Performing an SQL query from an FTP client

To have the FTP client perform SQL queries and have the results sent to an FTP server, specify:

```
LOCSITE FILEtype=SQL
```

Next, perform a PUT command specifying the name of the file on the client that contains the SQL query.

For example, if the client has a file named `userid.SQL.IN` that contains an SQL query, you can specify:

```
PUT SQL.IN SQL.OUT
```

The FTP client then submits the query found in `SQL.IN` to the DB2 subsystem on the client and sends the resulting rows of output to the server to be put into `SQL.OUT` on the server.

To return to normal FTP processing, specify:

```
LOCSITE FILEtype=SEQ
```

#### Performing an SQL query from an FTP server

To have the FTP server perform the query and have the results sent to the client, specify:

```
SITE FILEtype=SQL
```

Then perform a GET command specifying the name of the file on the server that contains the SQL query.

For example, if the server has a file named `userid.SQL.IN` that contains an SQL query, you can specify:
GET SQL.IN SQL.OUT

The FTP server then submits the query found in SQL.IN to the DB2 subsystem on the server and sends the resulting rows of output to the client to be put into SQL.OUT on the client.

Examples of SQL query output
This section shows examples of SQL query output using different options.

With NOSPRead and SQLCol=Names
The following output is from a query using NOSPRead and SQLCol=Names.

```
+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
| EMPLOYEE,AGE          | EMPLOYEE,AGE          | EMPLOYEE,AGE          | EMPLOYEE,AGE          | EMPLOYEE,AGE          | EMPLOYEE,AGE          | EMPLOYEE,AGE          | EMPLOYEE,AGE          |
|-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------|
| SELECT EMPLOYEE,AGE   | FROM ABC.STAFF        | WHERE AGE < 60        | Steve Jasinski        | 23                    | Alison Cook           | 22                    |
| h----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------|
```

Note: The period symbol (.) represents a TAB character.

With SPRead and SQLCol=Names
The following output is from a query with SPRead and SQLCol=Names.

```
+-----------------------+
| EMPLOYEE .AGE         |
| Steve Jasinski . 23   |
| Alison Cook . 22      |
```

The following output examples are for the query:

```
SELECT DISTINCT ABC.STAFF.TLA, ABC.STAFF.SALARY
FROM ABC.STAFF, ABC.HOURS
WHERE (ABC.STAFF.TLA = ABC.HOURS.TLA) AND
      (ABC.HOURS.TOTAL > 40)
```

With NOSPRead and SQLCol=Names
The following are output from queries using NOSPRead and SQLCol=Names.

```
+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
| SELECT DISTINCT ABC.STAFF.TLA, ABC.STAFF.SALARY
| FROM ABC.STAFF, ABC.HOURS
| WHERE (ABC.STAFF.TLA = ABC.HOURS.TLA) AND
| (ABC.HOURS.TOTAL > 40)
| h----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------|
| TLA SALARY             |
| ACO 20050.00           |
| SJJ 19040.00           |
+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
```

and

```
+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
| SELECT * FROM ABC.STAFF
| h----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------|
| EMPLOYEE TLA AGE SALARY|
| Steve Jasinski SJJ 23 28040.00|
| Alison Cook ACO 22 28040.00|
| Mark Ballam MFB 63 87420.55|
+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
```

With NOSPRead and SQLCol=Labels
The following output is from a query using NOSPRead and SQLCol=Labels.
FTP with DBCS support

This section describes how to use FTP to exchange DBCS data sets between hosts supporting DBCS file transfer.

The TCP/IP FTP server and client programs access data sets containing data that is usually in EBCDIC format. To transfer these data sets to or from an ASCII-based host requires translation tables. The transfer of DBCS data uses two tables—one for DBCS characters and one for SBCS characters.

Selecting a DBCS translation table

The LOADDDBCSTABLES statement in TCPIP.DATA is used by both the FTP server and client to determine which DBCS translation table data sets can be loaded. Refer to the z/OS Communications Server: IP Configuration Reference for more information about the loading and customizing of DBCS translation tables for FTP.

The FTP server and client can be configured to load a number of DBCS translation tables. These are used during data set transfers to convert MVS host DBCS characters and non-MVS DBCS characters. The command

\[ \text{TYPE B n} \]

or corresponding client subcommand is used to enter DBCS transfer mode and select a DBCS table.

Selecting an SBCS translation table

The SBCS table used to transfer DBCS data is the SBCS table that is established for the data connection.

SBCS tables are used by the control connection to transfer commands; they are also used by the data connection. Often the same SBCS table is used, but you may want to select a different table to be used for data transfers. How you specify the SBCS table for the data connection depends on whether the translation is to be done by the FTP server or the FTP client.
When the EBCDIC-to-ASCII translation is done by the FTP server, you can issue a SITE SBDataconn command to select the SBCS table to be used by the server for data transfers.

When the EBCDIC-to-ASCII translation is done by the FTP client, you can use the following parameters in your local FTP.DATA file to establish the SBCS tables:

- **CTRLConn**: Establishes the SBCS tables the client uses for control connections.
- **SBDataconn**: Establishes the SBCS tables the client uses for data connections.

Alternatively, you can use the TRANSLATE option of the FTP command to change the SBCS and DBCS translation table hierarchy for both the control and data connection. The TRANSLATE option results in the same SBCS table for both the control and the data connection.

**Note:** The TRANSLATE option can be used as long as the table maintains the integrity of the portable character set.

Another alternative when the FTP client is to perform the translation is to use the LOCSITE SBDataconn subcommand to change the SBCS table used by the client for the data connection.

### DBCS subcommands

DBCS data sets are transferred using the standard FTP subcommands PUT and GET. However, before the transfer commences, the current transfer type for the session must be set to the required DBCS type. To set the transfer type to DBCS for an FTP session, you must issue the appropriate FTP subcommand to the client or the server, depending on where the DBCS conversion is to be done. The FTP subcommands for DBCS support are listed in Table 16.

#### Table 16. FTP subcommands for DBCS support

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG5</td>
<td>Sets the transfer type to BIG-5</td>
<td>135</td>
</tr>
<tr>
<td>EUckanji</td>
<td>Sets the transfer type to EUCKANJI</td>
<td>162</td>
</tr>
<tr>
<td>HAngeul</td>
<td>Sets the transfer type to HANGEUL</td>
<td>170</td>
</tr>
<tr>
<td>Ibmkanji</td>
<td>Sets the transfer type to IBMKANJI</td>
<td>173</td>
</tr>
<tr>
<td>JIS78kj</td>
<td>Sets the transfer type to JIS78KJ</td>
<td>174</td>
</tr>
<tr>
<td>JIS83kj</td>
<td>Sets the transfer type to JIS83KJ</td>
<td>175</td>
</tr>
<tr>
<td>Ksc5601</td>
<td>Sets the transfer type to KSC5601</td>
<td>176</td>
</tr>
<tr>
<td>QUOte</td>
<td>Sends an uninterpreted string of data</td>
<td>229</td>
</tr>
<tr>
<td>SChinese</td>
<td>Sets the transfer type to SCHINESE</td>
<td>236</td>
</tr>
<tr>
<td>SJiskanji</td>
<td>Sets the transfer type to SJISKANJI</td>
<td>263</td>
</tr>
<tr>
<td>TChinese</td>
<td>Sets the transfer type to TCHINESE</td>
<td>273</td>
</tr>
<tr>
<td>Typse</td>
<td>Specifies the transfer type</td>
<td>276</td>
</tr>
</tbody>
</table>

### Server commands and client subcommands

Table 17 on page 125 shows examples of the server command that would be generated for each client subcommand:
### Table 17. FTP TYPE subcommand aliases

<table>
<thead>
<tr>
<th>Client subcommand</th>
<th>Server command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG5</td>
<td>TYPE B 8</td>
<td>Big-5 transfer type</td>
</tr>
<tr>
<td>EUCKANJI</td>
<td>TYPE B 2</td>
<td>Extended UNIX Code kanji transfer type</td>
</tr>
<tr>
<td>HANGEUL</td>
<td>TYPE B 5</td>
<td>Hangeul transfer type</td>
</tr>
<tr>
<td>IBMKANJII</td>
<td>TYPE F 1</td>
<td>IBM (EBCDIC) kanji transfer type</td>
</tr>
<tr>
<td>JIS78KJ</td>
<td>TYPE B 4 A</td>
<td>JIS 1978 kanji using ASCII shift-in transfer type</td>
</tr>
<tr>
<td>JIS78KJ (ASCII)</td>
<td>TYPE B 4 A</td>
<td>ASCII shift-in escape sequence</td>
</tr>
<tr>
<td>JIS78KJ (JISROMAN)</td>
<td>TYPE B 4 R</td>
<td>JISROMAN shift-in escape sequence</td>
</tr>
<tr>
<td>JIS78KJ (JISROMAN NOSO)</td>
<td>TYPE B 4 R N</td>
<td>Pure DBCS data transfer</td>
</tr>
<tr>
<td>JIS83KJ</td>
<td>TYPE B 3 A</td>
<td>JIS 1983 kanji using ASCII shift-in transfer type</td>
</tr>
<tr>
<td>JIS83KJ (ASCII)</td>
<td>TYPE B 3 A</td>
<td>ASCII shift-in escape sequence</td>
</tr>
<tr>
<td>JIS83KJ (JISROMAN)</td>
<td>TYPE B 3 R</td>
<td>JISROMAN shift-in escape sequence</td>
</tr>
<tr>
<td>JIS83KJ (JISROMAN NOSO)</td>
<td>TYPE B 3 R N</td>
<td>Pure DBCS data transfer</td>
</tr>
<tr>
<td>KSC5601</td>
<td>TYPE B 6</td>
<td>Korean Standard Code KSC-5601 transfer type</td>
</tr>
<tr>
<td>SCHINESE</td>
<td>TYPE B 9</td>
<td>Simplified Chinese transfer type</td>
</tr>
<tr>
<td>SJISKANJII</td>
<td>TYPE B 1</td>
<td>Shift JIS kanji transfer type</td>
</tr>
<tr>
<td>SJISKANJII (Sosi)</td>
<td>TYPE B 1 S A</td>
<td>Shift-out/shift-in characters X'1E'/X'1F'</td>
</tr>
<tr>
<td>SJISKANJII (Sosi ASCII)</td>
<td>TYPE B 1 S A</td>
<td>Shift-out/shift-in characters X'1E'/X'1F'</td>
</tr>
<tr>
<td>SJISKANJII (Sosi EBCDIC)</td>
<td>TYPE B 1 S E</td>
<td>Shift-out/shift-in characters X'0E'/X'0F'</td>
</tr>
<tr>
<td>SJISKANJII (Sosi SPACE)</td>
<td>TYPE B 1 S S</td>
<td>Shift-out/shift-in characters X'20'/X'20'</td>
</tr>
<tr>
<td>SJISKANJII (NOSO)</td>
<td>TYPE B 1 N</td>
<td>Pure DBCS data transfer</td>
</tr>
<tr>
<td>TCHINESE</td>
<td>TYPE B 7</td>
<td>Traditional Chinese (5550) transfer type</td>
</tr>
</tbody>
</table>

### Mapping DBCS aliases to CCSIDs

The code sets supported by the DBCS for FTP options conform to standard coded character set identifiers (CCSIDs). Table 18 shows how CCSIDs map to DBCS keywords.

For more information about CCSIDs, refer to Character Data Representation Architecture Reference and Registry, GC09-2207.

### Table 18. Mapping of DBCS keywords to CCSIDs

<table>
<thead>
<tr>
<th>DBCS keyword</th>
<th>CCSID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG5</td>
<td>00947</td>
<td>IBM Big-5 DBCS</td>
</tr>
</tbody>
</table>
Table 18. Mapping of DBCS keywords to CCSIDs (continued)

<table>
<thead>
<tr>
<th>DBCS keyword</th>
<th>CCSID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUCKANJI</td>
<td>00954</td>
<td>Japanese EUC (G0, G1 and G2 only)</td>
</tr>
<tr>
<td>HANGEUL</td>
<td>00926</td>
<td>Korean DBCS-PC</td>
</tr>
<tr>
<td>JIS78KJ</td>
<td>00955</td>
<td>JIS X0208–1978</td>
</tr>
<tr>
<td>JIS83KJ</td>
<td>05048</td>
<td>JIS X0208–1990</td>
</tr>
<tr>
<td>KSC5601</td>
<td>00951</td>
<td>IBM Korean Standard code</td>
</tr>
<tr>
<td>SCHINESE</td>
<td>01380</td>
<td>Simplified Chinese DBCS-PC</td>
</tr>
<tr>
<td>SJISKANJII</td>
<td>00301</td>
<td>Japanese DBCS-PC</td>
</tr>
<tr>
<td>TCHINESE</td>
<td>00927</td>
<td>Traditional Chinese DBCS-PC</td>
</tr>
</tbody>
</table>
Chapter 5. FTP subcommands

The FTP subcommands are listed in Table 19. The minimum abbreviation, a description, and the page reference for each subcommand are also included. You must be in the FTP environment to use the FTP subcommands. See “Using the FTP command” on page 29 for more information.

Table 19. FTP subcommands

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Provides information to use FTP.</td>
<td>172</td>
</tr>
<tr>
<td>!</td>
<td>Passes a z/OS UNIX System Services command to the local z/OS shell. This command must be issued while using FTP in the z/OS shell.</td>
<td>130</td>
</tr>
<tr>
<td>ACount</td>
<td>Sends host-dependent account information.</td>
<td>131</td>
</tr>
<tr>
<td>APpend</td>
<td>Appends a data set on your local host to a file on the foreign host.</td>
<td>132</td>
</tr>
<tr>
<td>AScii</td>
<td>Sets the transfer type to ASCII.</td>
<td>134</td>
</tr>
<tr>
<td>BIG5</td>
<td>Sets the transfer type to BIG5. BIG is the minimum abbreviation for BIG5.</td>
<td>135</td>
</tr>
<tr>
<td>Binary</td>
<td>Sets the transfer type to IMAGE.</td>
<td>137</td>
</tr>
<tr>
<td>BLock</td>
<td>Sets the data transfer mode to block mode. This is equivalent to specifying the MOde B subcommand.</td>
<td>138</td>
</tr>
<tr>
<td>CCc</td>
<td>Turns off integrity protection on the command channel.</td>
<td>139</td>
</tr>
<tr>
<td>CD</td>
<td>Changes the working directory.</td>
<td>140</td>
</tr>
<tr>
<td>CDUp</td>
<td>Changes to the parent of the current working directory.</td>
<td>143</td>
</tr>
<tr>
<td>CLEar</td>
<td>Sets the protection level for data transfers to CLEAR.</td>
<td>145</td>
</tr>
<tr>
<td>Close</td>
<td>Disconnects from the foreign host.</td>
<td>146</td>
</tr>
<tr>
<td>COMpress</td>
<td>Sets the data transfer mode to compressed mode. This is equivalent to specifying the MOde C subcommand.</td>
<td>147</td>
</tr>
<tr>
<td>CProtect</td>
<td>Sets the protection level on commands. CProtect CLEar is equivalent to the CCc command.</td>
<td>148</td>
</tr>
<tr>
<td>CWd</td>
<td>Changes the working directory (Synonymous with CD).</td>
<td>140</td>
</tr>
<tr>
<td>DEBug</td>
<td>Enables or disables general internal tracing.</td>
<td>149</td>
</tr>
<tr>
<td>DELEte</td>
<td>Deletes a single file on the foreign host.</td>
<td>153</td>
</tr>
<tr>
<td>DELimit</td>
<td>Displays the delimiter character between the file_name and file_type.</td>
<td>154</td>
</tr>
<tr>
<td>Dlr</td>
<td>Lists the directory entries for files on the foreign host.</td>
<td>155</td>
</tr>
<tr>
<td>DUMP</td>
<td>Enables or disables extended internal tracing.</td>
<td>159</td>
</tr>
<tr>
<td>EBcdic</td>
<td>Sets the transfer type to EBCDIC.</td>
<td>161</td>
</tr>
<tr>
<td>EUckanji</td>
<td>Sets the transfer type to EUCKANJI.</td>
<td>162</td>
</tr>
<tr>
<td>FEature</td>
<td>Queries FTP Server for features it supports.</td>
<td>164</td>
</tr>
<tr>
<td>File</td>
<td>Sets the file structure to file. This is equivalent to specifying the STRucture F subcommand.</td>
<td>165</td>
</tr>
<tr>
<td>Get</td>
<td>Copies a file from the foreign host to your local host.</td>
<td>166</td>
</tr>
<tr>
<td>Subcommand</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>GLOB</td>
<td>Toggles globbing (the expansion of metacharacters in file names) for the MDelete, MGet, and MPut subcommands.</td>
<td>168</td>
</tr>
<tr>
<td>HANGEUL</td>
<td>Sets the transfer type to HANGEUL.</td>
<td>170</td>
</tr>
<tr>
<td>HELP</td>
<td>Displays help information for FTP.</td>
<td>172</td>
</tr>
<tr>
<td>IBMKANJI</td>
<td>Sets the transfer type to IBMKANJI.</td>
<td>173</td>
</tr>
<tr>
<td>JIS78KJ</td>
<td>Sets the transfer type to JIS78KJ.</td>
<td>174</td>
</tr>
<tr>
<td>JIS83KJ</td>
<td>Sets the transfer type to JIS83KJ.</td>
<td>175</td>
</tr>
<tr>
<td>KSC5601</td>
<td>Sets the transfer type to KSC5601.</td>
<td>176</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>Sets the language used for FTP replies from the server.</td>
<td>178</td>
</tr>
<tr>
<td>LCMD</td>
<td>Changes the current directory on the local host.</td>
<td>179</td>
</tr>
<tr>
<td>LMKDIR</td>
<td>Creates a PDS on the local host.</td>
<td>181</td>
</tr>
<tr>
<td>LOCSTITE</td>
<td>Specifies information that is used by the local host to provide service specific to that host system.</td>
<td>183</td>
</tr>
<tr>
<td>LOCSTAT</td>
<td>Displays FTP status information for the local host.</td>
<td>201</td>
</tr>
<tr>
<td>LPWD</td>
<td>Displays the name of the active working directory on the local host.</td>
<td>202</td>
</tr>
<tr>
<td>LS</td>
<td>Lists the names of files on the foreign host.</td>
<td>203</td>
</tr>
<tr>
<td>MDELETE</td>
<td>Deletes multiple files on the foreign host.</td>
<td>205</td>
</tr>
<tr>
<td>MGET</td>
<td>Copies multiple files from the foreign host to your local host.</td>
<td>207</td>
</tr>
<tr>
<td>MKDIR</td>
<td>Creates a directory on the foreign host.</td>
<td>211</td>
</tr>
<tr>
<td>MODE</td>
<td>Specifies the mode or data format of the transfer.</td>
<td>213</td>
</tr>
<tr>
<td>MPUT</td>
<td>Copies multiple files on your local host to the foreign host.</td>
<td>214</td>
</tr>
<tr>
<td>NOOP</td>
<td>Checks whether the foreign host is still responding.</td>
<td>216</td>
</tr>
<tr>
<td>OPEN</td>
<td>Opens a connection to a foreign host.</td>
<td>217</td>
</tr>
<tr>
<td>PASS</td>
<td>Supplies a password to the foreign host.</td>
<td>218</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>Sets the protection level for data transfers to PRIVATE.</td>
<td>219</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Toggles interactive prompting for MDelete, MGet, and MPut commands. This function is similar to specifying the FTP command with the -i option, which turns off interactive prompting.</td>
<td>220</td>
</tr>
<tr>
<td>PROTECT</td>
<td>Sets the protection level for data transfers on the data connections.</td>
<td>221</td>
</tr>
<tr>
<td>PROXY</td>
<td>Executes an FTP subcommand on a secondary control connection.</td>
<td>222</td>
</tr>
<tr>
<td>PUT</td>
<td>Copies a file on your local host to the foreign host.</td>
<td>225</td>
</tr>
<tr>
<td>PWDD</td>
<td>Displays the name of the active working directory on the foreign host.</td>
<td>227</td>
</tr>
<tr>
<td>QUIT</td>
<td>Leaves the FTP command environment.</td>
<td>228</td>
</tr>
<tr>
<td>QUOTE</td>
<td>Sends an uninterpreted string of data.</td>
<td>229</td>
</tr>
<tr>
<td>RECORD</td>
<td>Sets the file structure to record. This is equivalent to specifying the STRUcture R subcommand.</td>
<td>231</td>
</tr>
<tr>
<td>Subcommand</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>REName</td>
<td>Renames a file on the foreign host.</td>
<td>232</td>
</tr>
<tr>
<td>REStart</td>
<td>Restarts a checkpointed data transfer.</td>
<td>233</td>
</tr>
<tr>
<td>RMDir</td>
<td>Removes a directory.</td>
<td>234</td>
</tr>
<tr>
<td>SAfe</td>
<td>Sets the protection level on data transfers to safe.</td>
<td>235</td>
</tr>
<tr>
<td>SCHinese</td>
<td>Sets the transfer type to SCHINESE.</td>
<td>236</td>
</tr>
<tr>
<td>SENDPort</td>
<td>Enables or disables automatic transmission of the FTP server PORT command.</td>
<td>238</td>
</tr>
<tr>
<td>SENDSite</td>
<td>Enables or disables automatic transmission of the SItesubcommand.</td>
<td>239</td>
</tr>
<tr>
<td>SItes</td>
<td>Sends information to the foreign host using site-specific commands.</td>
<td>240</td>
</tr>
<tr>
<td>SJiskanji</td>
<td>Sets the transfer type to SJISKANJI.</td>
<td>263</td>
</tr>
<tr>
<td>SRestart</td>
<td>Restarts an interrupted stream mode data transfer.</td>
<td>265</td>
</tr>
<tr>
<td>SStatus</td>
<td>Displays status information for the foreign host.</td>
<td>267</td>
</tr>
<tr>
<td>SREam</td>
<td>Sets the data transfer mode to stream mode. This is equivalent to specifying the MOde Subcommand.</td>
<td>269</td>
</tr>
<tr>
<td>STRucture</td>
<td>Sets the file transfer structure.</td>
<td>270</td>
</tr>
<tr>
<td>SUnique</td>
<td>Toggles the storage methods.</td>
<td>271</td>
</tr>
<tr>
<td>SYstem</td>
<td>Displays the name of the foreign host operating system.</td>
<td>272</td>
</tr>
<tr>
<td>TChinese</td>
<td>Sets the transfer type to TCHINESE.</td>
<td>273</td>
</tr>
<tr>
<td>TSO</td>
<td>Passes a TSO command to the local host TSO environment.</td>
<td>275</td>
</tr>
<tr>
<td>TYpe</td>
<td>Specifies the transfer type.</td>
<td>276</td>
</tr>
<tr>
<td>UCs2</td>
<td>Changes the data transfer type to Unicode UCS-2. UC is the minimum abbreviation for UCs2.</td>
<td>280</td>
</tr>
<tr>
<td>User</td>
<td>Identifies you to a foreign host or changes your TSO user ID password.</td>
<td>281</td>
</tr>
<tr>
<td>Verbose</td>
<td>Enables or disables verbose mode.</td>
<td>283</td>
</tr>
</tbody>
</table>
! subcommand—Invoke a z/OS UNIX System Services function

Purpose

In a z/OS UNIX environment, use the ! subcommand to invoke z/OS UNIX functions.

Format

```
! [shell_command]
```

Parameters

shell_command

Specifying the ! subcommand with a shell command enables you to invoke z/OS UNIX, perform the subcommand, and return to the FTP environment.

If no shell command is specified, the ! subcommand invokes z/OS UNIX. There you can specify any number of shell commands before typing exit to return to the FTP environment.
ACcount subcommand—Supply account information

Purpose

Use the ACcount subcommand to supply account information to a host.

Format

```
ACcount account_information
```

Parameters

account_information

Specifies the account information required by the host. Refer to your foreign-host FTP server documentation for the information required by that host.

Usage

- The z/OS FTP server does not require any account information.
- You may have to use the ACcount subcommand when the foreign host requires passwords for read and write access to its files or data sets. If you are not prompted by the foreign host for the passwords, use the ACcount subcommand to send these passwords to the foreign host.
APPend subcommand—Append a local data set

Purpose

Use the APPend subcommand to append a local data set to a remote host.

Format

APPend local_data_set destination_file

Parameters

local_data_set
The name of the data set on your local host to be appended.

destination_file
The name of the file on the remote host to which your data set is appended. If the destination file does not already exist at the remote host, a new file is created. If the server is a z/OS UNIX server, the local file can be appended to an HFS file or an MVS data set.

Examples

In the following example, an FTP command is issued from MVSXA2 to MVSXA3. MVSXA2 has a data set MVSUSER.FTP.EXAMPLE with one member. The member, APPEND01, contains:

; THIS FILE ORIGINALLY RESIDED IN MVSXA2, AND
; WILL BE APPENDED TO A FOREIGN FILE IN MVSXA3.
;

MVSXA3 has a data set, MVSUSER.FTP.EXAMPLE, with one member, APPEND02. The member contains:

; THIS FILE ORIGINALLY RESIDED IN MVSXA3, AND
; WILL BE USED TO RECEIVE ANOTHER FILE FROM MVSXA2.
;

User: append
System: Usage: APPEND localfile foreignfile
Command:

User: lpwd
System: Local directory is MVSUSER.
Command:

User: append 'mvsuser.ftp.example(append01)' 'mvsuser.ftp.example(append02)'
System: SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
200 Site command was accepted
PORT 1,1,2,2,4,16
200 Port request OK.
APPE 'mvsuser.ftp.example(append02)'
125 Appending to data set MVSUSER.FTP.EXAMPLE(APPEND02)
250 Transfer completed successfully.
520 bytes transferred in 1.100 seconds.
Transfer rate 0.47 Kbytes/sec.
Command:
Usage

- FTP maintains the attributes of a data set that is transmitted between a client and a server. However, when you use the APPend subcommand, FTP may truncate data records and you might lose data. If the data set name already exists at the receiving site and the logical record length (LRecl) of the data set at the receiving site is less than the LRecl of the transmitted data set, FTP truncates the transmitted data set.

- If the remote host is an MVS or VM host, and if the data set on the remote host has a fixed-record format, the format and record length of the data set on the remote host are always preserved.

- Records from the data set on your local host are truncated or padded with blank spaces when necessary.

- To append to a file on a remote host, you must define a working directory on that host, and you must have write privileges to the files in that directory.

- The z/OS FTP Server does not request that unused space be released from a data set created during APPEND processing. If you are using the z/OS FTP Server and wish for the Server to request that unused space be released on a newly-created data set, use the PUT subcommand instead of APPEND.

Context

- See “CD subcommand—Change the directory on the remote host” on page 140 for more information about working with current directories.

- See Appendix A, “Specifying data sets and files” on page 383 for more information about naming conventions.

- APPend can be used with the PROXY subcommand to transfer files from a host on a secondary connection to a host on a primary connection. See “PROXY subcommand—Execute FTP subcommand on secondary control connections” on page 222 for more information.
AScii subcommand—Change the data transfer type to ASCII

Purpose
Use the AScii subcommand to change the data transfer type to ASCII.

Format
```
\[Z(qOOOO\[Z(qOOOO
```

Parameters
There are no parameters for this subcommand.

Usage
Use the ASCII subcommand to direct FTP to translate outgoing files into ASCII before sending them to the other host, and to convert incoming files from ASCII to the file system code page before storing them.

Context
For more information about transfer methods, see Table 11 on page 51.
**BIG5 subcommand—Change the data transfer type to BIG5**

**Purpose**

Use the BIG5 subcommand to change the data transfer type to BIG5.

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the Site/LOCSIte SBDataconn or Site XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

**Format**

```
BIG5
  (Ascii NOSo NOType
   Sosi
   Ebcdic
   Space)
```

**Parameters**

- **Sosi**
  Transferred data contains the shift-out and shift-in characters specified by one of the following parameters – Ascii, Ebcdic, or Space. If no parameter is specified, ASCII is used as the default.
  
  If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

- **Ascii**
  When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

- **Ebcdic**
  When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

- **Space**
  When combined with the Sosi parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

- **NOSo**
  Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to and from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

- **NOType**
  Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.
Usage

- The BIG5 client subcommand is equivalent to the TYPE B 8 command.
- The minimum abbreviation for BIG5 is BIG.

Context

See “FTP with DBCS support” on page 123 for more information.
**BINary subcommand—Change the data transfer type to Image**

**Purpose**

Use the BINary subcommand to change the data transfer type to image (binary).

**Format**

```
>Z(qOOOO>Z(qOOOO
BINary
```

**Parameters**

There are no parameters for this subcommand.

**Usage**

Use the image transfer type to transfer files between client and server without any translation of the file data. When using the image transfer type, data is sent as contiguous bits packed into 8-bit bytes. Use the image transfer type for efficient storage and retrieval of data sets or files, and for the transfer of binary data.

**Context**

For more information about data transfer methods, see [Table 11 on page 51](#).
BLock subcommand—Set the block data transfer mode

Purpose

Use the BLock subcommand to set the data transfer mode to block mode. This is equivalent to specifying the MOde B subcommand. See “MOde subcommand—Set the data transfer mode” on page 213 for more information.

Format

\[ \text{B} \text{Lock} \]

Parameters

There are no parameters for this subcommand.
CCc subcommand—Turn off integrity protection

Purpose

Use the CCc subcommand to turn off integrity protection on the control connection. This command must be integrity-protected, and must be preceded by a successful security mechanism negotiation.

Format

```
CCc
```

Parameters

There are no parameters for this subcommand.

Usage

Since turning off integrity protection potentially allows an attacker to insert commands onto the control connection, some FTP servers may refuse to honor this command.
CD subcommand—Change the directory on the remote host

Purpose

Use the CD subcommand to change the working directory or file group on the remote host.

Format

```
   CD directory
```

Parameters

directory

Specifies the name of a file directory, a fully qualified data set, or a prefix on the remote host.

Examples

**Changing the directory of a z/OS FTP server**

If the remote server is z/OS FTP, *directory* can specify either an HFS name, a common prefix for a group of MVS data sets, or the qualifiers of a Partitioned Data Set (PDS).

When the CD subcommand is issued, the *directory* specified is appended to the current working directory. For example, if the current working directory is TCPUSR14.TEST, and you issue the CD subcommand:

```
CD FILES
```

the new working directory becomes TCPUSR14.TEST.FILES.

To override the existing directory rather than append to the directory, issue the *directory* parameter within single quotation marks. For example, if the current working directory is TCPUSR14.TEST, and you issued the CD subcommand:

```
CD 'FTP.FILES'
```

the new working directory would be FTP.FILES. If the subdirectory name contains white space, such as NEW SUBDIRECTORY, then the syntax for the CD command would be as follows:

```
CD 'NEW SUBDIRECTORY'
```

The command syntax must specify the full subdirectory name (including the blank) delimited within single quotation marks.

If a PDS exists with the exact name of the current working directory, FTP considers the working directory to be that PDS. Otherwise, FTP considers the working directory to be a common prefix qualifier for sequential data sets.

If a PDS exists with the same name as the current working directory, but you want the current working directory to be treated as a common prefix for sequential data sets, specify the working directory with a period (.) at the end. For example, if a PDS named TCPUSR14.TEST exists, the subcommand:

```
CD 'TCPUSR14.TEST'
```
makes the PDS TCPUSR14.TEST the current working directory. A subsequent PUT of file name1 adds a member name1 to the TCPUSR14.TEST PDS. In contrast, the statement:

CD 'TCPUSR14.TEST.'

makes the current working directory, TCPUSR14.TEST., a prefix for sequential data sets. A subsequent PUT command used to copy data set name1 would create the sequential data set TCPUSR14.TEST.name1.

To back up one level of the current working directory, issue the CD subcommand with two periods (..) at the end. For example, if the working directory is jones.source, the statement,

CD..

makes jones. the working directory. You can also use the CDUp command to back up one level of the current working directory. See the "CDUp subcommand—Change to the parent of the working directory" on page 143 for more details.

The following sample commands and responses are displayed as a result of the CD subcommand.

For an MVS data set:

```
  cd hsmtest
  >>>CWD hsmtest
  250 "USER17.HSMTEST." is working directory name prefix.

  Command:
```

For an HFS file:

```
  cd '/u/user121/A/B/C'
  >>>CWD '/u/user121/A/B/C'
  250 HFS directory /u/user121/A/B/C is the current working directory

  Command:
```

Changing the directory of a VM FTP server

If the remote host is using TCP/IP for VM, the directory can be specified in either of the following ways:

- `user_id minidisk_address`
- `user_id.minidisk_address`

For example, to access the 191 minidisk of user ID jones, enter one of the following:

- jones 191
- jones.191

Testing throughput with *DEV.NULL

If you have a z/OS FTP server, you can use the PUT command to copy a large number of files (or one large file) without having the files actually stored on a z/OS FTP server file system. This is useful for testing purposes because you do not have to worry about allocating the disk space on the server system.

To use this function, first change the working directory to *DEV.NULL by using one of the following commands:

- CD *DEV.NULL
**CWD *DEV.NULL**

This affects the working directory only for PUT commands. You then use the PUT command to copy the file to the server system. The input data set must be valid, and the output file can either be new or already exist. In either case, the file is not actually stored.

The following response shows information such as the number of bytes transferred and the rate of transfer.

To end the use of the *dev.null directory for PUT commands, issue another CHANGE Directory command.

```
Command: cd *dev.null
        >>>CWD *dev.null
250-Working directory for PUT is NULL Device;
250 for GET is HFS directory /u/user31
Command: put a.b a.bbbbb
        >>>SITE VARrecfm Lrecl=128 Recfm=VB BlockSize=6144
200 Site command was accepted
        >>>PORT 14,0,0,0,4,14
200 Port request OK.
        >>>STOR a.bbbbb
125 Storing data set in the Null directory (*dev.null).
250 Transfer completed successfully.
82 bytes transferred in 0.245 seconds. Transfer rate 0.33 Kbytes/sec.
Command: quit
        >>>QUIT
221 Quit command received. Goodbye.
```

**Usage**

You can also use the CWD and CW subcommands to change the current working directory. These subcommands are synonyms of the CD subcommand.
CDUp subcommand—Change to the parent of the working directory

Purpose

Use the CDUp subcommand as a special case of the CD subcommand to change the working directory to the next higher directory level. You can use it to simplify the implementation of programs for transferring directory trees between operating systems that have different syntaxes for naming the parent directory.

Format

```
CDUp
```

Parameters

There are no parameters for this subcommand.

Examples

Change the working directory to the next higher directory level:

```
cd 'a.b.c.d'
>>>CWD 'a.b.c.d'
257 "'A.B.C.D.'" is working directory name prefix.
Command: pwd

>>>PWD
257 "'A.B.C.D.'" is working directory
Command: cdup

>>>CDUP
257 "'A.B.C.'" is working directory name prefix.
Command: pwd

>>>PWD
257 "'A.B.C.'" is working directory
Command: cdup

>>>CDUP
257 "'A.B.'" is working directory name prefix.
Command: pwd

>>>PWD
257 "'A.B.'" is working directory
```

Change the working directory to the next higher directory level for an HFS file:
cd '/u/user121/A/B/C'
>>>CWD '/u/user121/A/B/C'
250 HFS directory /u/user121/A/B/C is the current working directory
Command: pwd

>>>PWD
257 "/u/user121/A/B/C" is the HFS working directory
Command: cdup

>>>CDUP
250 HFS directory /u/user121/A/B is the current working directory
Command: pwd

>>>PWD
257 "/u/user121/A/B" is the HFS working directory
Command: cdup

>>>CDUP
250 HFS directory /u/user121/A is the current working directory
Command: pwd

>>>PWD
257 "/u/user121/A" is the HFS working directory.
Command:
CLEar subcommand—Set the protection level for data transfers to CLEAR

Purpose

Use the CLEar subcommand to set the protection level for data transfers on the data connections to clear. This subcommand is equivalent to specifying the PROTect CLEar subcommand.

Format

```plaintext
CLEar
```

Parameters

There are no parameters for this subcommand.

Examples

To set the protection level to clear, enter:
```
clear
```

Usage

See the "PROTect subcommand—Set the protection level for data transfers" on page 221 for additional protection level information.

The CLEar subcommand is not valid when there is no active security mechanism.
**CLose subcommand—Disconnect from a remote host**

**Purpose**

Use the CLose subcommand to disconnect from the remote host and remain in FTP.

**Format**

```
> CLose
```

**Parameters**

There are no parameters for this subcommand.

**Usage**

The FTP session remains active on your local host, but the session to the remote host is terminated. You can use the Open subcommand to establish a new session with either the same or a different remote host. If you establish a new session with the same remote host, values set by the SItes subcommand during the previous session are cleared. The remote host default values for the parameters of the SItes subcommand are used for the new session.

**Context**

- See "Open subcommand—Connect to the FTP server" on page 217 for information about the Open subcommand.
- CLose can be used with the PROXy subcommand to close a secondary control connection. See "PROXy subcommand—Execute FTP subcommand on secondary control connections" on page 222 for more information.
COMpress subcommand—Set the compressed data transfer mode

Purpose

Use the COMpress subcommand to set the data transfer mode to compressed mode. This is equivalent to specifying the MOde C subcommand. See "MOde subcommand—Set the data transfer mode" on page 213 for more information.

Format

```
> [Z(qOOO] [Z(qOOO
COMpress
```

Parameters

There are no parameters for this subcommand.
CProtect subcommand—Set the protection level on commands

Purpose

Set the protection level on commands to protection-level.

Format

```
CProtect [protection-level]
```

Parameters

The valid protection levels are:

- **clear**
  - Unprotected commands
- **safe**
  - Commands integrity-protected by cryptographic checksum
- **private**
  - Commands confidentiality and integrity-protected by encryption

If an ADAT command succeeds, the default command protection level is **safe**; otherwise, the only possible level is **clear**. If no level is specified, the current level is used. **CProtect CLEar** is equivalent to the **CCc** command.
DEBug subcommand—Set general trace options

Purpose

Use the DEBug subcommand to enable or disable general internal tracing.

Format

```
DEBug
```

Parameters

One or more of the following options can be specified:

- `?` Displays the status of the traces.
- `ACC` The ACC trace shows the details of the login process.
- `ALL` This parameter is used to set all of the trace points.
  
  **Note:** Both the FSC and the SOC traces will be set to level 1 when the ALL parameter is processed.
- `BAS` This parameter is used to set a select group of traces that offer the best overall debug data without the detailed information of some of the traces. Specifying this parameter is the same as `DEBUG CMD INT FSC SOC`.
- `CMD` The CMD trace shows each command and the parsing of the parameters for the command.
- `FLO` The FLO trace shows the flow of control within FTP. It is useful to show which services of FTP are used for an FTP request.
- `FSC(n)` The FSC trace shows details of the processing of the file services

Chapter 5. FTP subcommands 149
subcommands APPend, PUt, Get, MGet, and MPut. This trace can generate very detailed information and therefore allows you to specify levels of detail for the trace points.

The level 1 tracing that is specified by entering FSC or FSC(1) is the level normally used unless more data is requested by TCP/IP service group. The level \((n)\) can be a number from 1 to 8.

**INT**

The INT trace shows the details of the initialization and termination of the FTP session with the server.

**NONE**

This parameter is used to turn off all of the traces.

**PAR**

The PAR trace shows details of the FTP command parser. It is useful for debugging problems in the handling of the command parameters.

**SEC**

The SEC trace shows the processing of security functions such as TLS and GSSAPI negotiations.

**SOC\((n)\)**

The SOC trace shows details of the processing during the setup of the interface between the FTP application and the network as well as details of the actual amounts of data that is processed.

This trace can generate very detailed information and therefore allows you to specify levels of detail for the trace points. The level \((n)\) can be a number from 1 to 8.

**SQL**

The SQL trace shows details of the processing for SQL requests, such as requests when LOCStite FILETYPE=SQL is in effect.

**TIMestamps**

This is a special parameter used to request that each trace entry have a timestamp showing the time the entry was made. This is useful during long running file transfers to track the duration of the transfer and also to match client trace to the server trace, which also has a timestamp.

**UTL**

The UTL trace shows the processing of utility functions such as LCd and LOCStite.

**Xyyy**

This syntax is used to turn off (reset) a trace that is named by yyy. For example, DEBUG XPAR XACC will turn off the PAR and the ACC traces.

### Examples

The following example shows sample client traces with DEBug.

debuge none fsc soc time
Active client traces - FSC(1) SOC(1)
get a 'user33.abc111' (replace
11:21:33 CG0204 get: F=1 p=FSA ARTW=0000
11:21:33 CG3356 rcvFile: entered
11:21:33 MR1200 set_filename: entered with pathname 'user33.abc111'
11:21:33 CG2078 mvs_rcvFile: entered
11:21:33 MR1200 set_filename: entered with pathname 'user33.abc111'
11:21:33 CG1944 newGDGname: entered
11:21:33 MV0874 seq_create_file: entered with dsn=USER33.ABC111
11:21:34 MV1545 seq_create_file: entered with dname=SYS000009

z/OS V1R4.0 CS: IP User’s Guide and Commands
By default, DEBug is off. When the FTP environment is entered, you can activate
DEBug by any of the following methods:

- Use the TRACE or -d parameter on the FTP command.
- Code one or more DEBUG statements in the client’s FTP.DATA.
- Code the TRACE statement in the client’s FTP.DATA.

Once FTP is started, you can change the DEBug settings with the DEBug
subcommand.

- The trace supports the DEBug parameters 1 and 2 that are used with previous
  product releases. Specifying DEBug BAS provides the same tracing as
  parameter 1 formerly provided. The DUMP subcommand now provides the
  extended tracing that parameter 2 formerly provided.
- The state of the traces points is displayed as a response to the DEBug
  subcommand. To see the states without making a change, enter DEBUG ?.
- The setting of the traces is additive as shown by the following::

```
DEBUG NONE CMD
EZA2851I Active traces: CMD
DEBUG PAR
EZA2851I Active traces: CMD PAR
```

- Entering DEBug with no parameters will toggle the trace on and off. The state of
  the traces when the trace is toggled off is remembered so that toggling it on
  restores the previous trace settings. If no traces were active previously, then
  toggling activates the BAS trace points

```
DEBUG NONE CMD
EZA2851I Active traces: CMD
DEB
EZA2851I Active traces: NONE
DEB
EZA2851I Active traces: CMD
DEB FSC
EZA2851I Active traces: CMD FSC(1)
DEB
```
EZA2851I Active traces: NONE
DEB
EZA2851I Active traces: CMD FSC(1)
DEB NONE
EZA2851I Active traces: NONE
DEB
EZA2851I Active traces: CMD INT FSC(1) SOC(1)

- The timestamp option is demonstrated with the following example:
  
  ```
  deb fsc(1)
  PC0304 parseCmd: subcommand: deb
  PC0307 parseCmd: parameter 1: fsc(1)
  Active traces: CMD FSC(1)
  Command:
  deb time
  PC0304 parseCmd: subcommand: deb
  PC0307 parseCmd: parameter 1: time
  Active traces: CMD FSC(1)
  Command:
  deb soc(1)
  11:39:37 PC0304 parseCmd: subcommand: deb
  11:39:37 PC0307 parseCmd: parameter 1: soc(1)
  Active traces: CMD FSC(1) SOC(1)
  Command:
  
  For the FSC and SOC trace options only one level of tracing can be defined at any time. However, when level 2 is defined, levels 1 and 2 are active. When level 3 is defined, levels 1, 2 and 3 are active. This progression also applies to levels 4 and 5.

  ```
  deb fsc(2) soc(1)
  Active traces: FSC(2) SOC(1)
  Command:
  deb fsc(1) soc(2)
  Active traces: FSC(1) SOC(2)
  Command:

  Note: The DEBug FSC command will accept level values 6–8, but will only provide level 5 trace data. Likewise, DEBug SOC will accept level values 4–8, but will only provide level 3 trace data.
DELEte subcommand—Delete files

Purpose

Use the DELEte subcommand to delete a file on the remote host.

Format

```
DELEte foreign_file
```

Parameters

`foreign_file`

Specifies the name of the file to be deleted on the remote host.

Context

See Appendix A, “Specifying data sets and files” on page 383 for information about file naming conventions.
DELIimit subcommand—Display the file name delimiter

Purpose

Use the DELImit subcommand to display the character that is used as the delimiter between the file name and the file type.

Format

```
> [Z(qOOOO> [Z(nOOOO
```

Parameters

There are no parameters for this subcommand.

Usage

- The DELImit subcommand should be used for information purposes only.
- You cannot change which character is used as the delimiter.
Dlr subcommand—Obtain a list of directory entries

Purpose

Use the Dlr subcommand to obtain a list of directory entries or a list of files in a file group on the remote host, or a list of the members of the partitioned data set, as well as auxiliary information about the files.

Format

```
Dlr [name] [/DISK]
```

Parameters

name

Specifies the name of the directory or file group. The default is the current directory or file group.

(DISK)

Stores the results of the Dlr subcommand as data set FTP.DIROUTP in the local current working directory. If the local current working directory is an MVS PDS, the member DIROUTP is stored. If the local current working directory is an HFS directory, the results are stored in a file named diroutp.

Examples

- List the data sets with a common high-level qualifier as the current working directory:

```
>>>PORT 9,67,58,227,4,62
200 Port request OK.
>>>LIST
125 List started OK.
Volume Unit Referred Ext UsedRecfm Lrecl Blksz Dsorg Dsname
WRKLB2 3380 1997/07/02 1 1 F 80 80 PS TEST.A
WRKLB2 3380 1997/07/09 4 7 F 80 80 PO TEST.PDS
WRKLB2 3380 1997/07/01 1617 VB 200 32000 PS TEST.XYZ1
250 List completed successfully.
Command:
```

Note: The Dlr output for a RECFM=U data set for the FTP display always shows the same value for lrecl as it shows for blksz.

- List the files for an HFS directory:
cd '/u/user121/ftp.example'

>>>CWD '/u/user121/ftp.example'
250 HFS directory /u/user121/ftp.example is the current working directory
Command:
dir

>>>PORT 9,67,112,25,4,61
200 Port request OK.

>>>NLST
125 List started OK
total 64
-rw-r----- 1 USER121 SYS1 6720 Feb 7 18:48 append02
-rw-r----- 1 USER121 SYS1 3360 Feb 6 18:51 file1
-rw-r----- 1 USER121 SYS1 3883 Feb 6 18:51 file2
-rw-r----- 1 USER121 SYS1 3883 Feb 6 18:51 file3
-rw-r----- 1 USER121 SYS1 7277 Feb 6 18:51 file4
-rw-r----- 1 USER121 SYS1 3360 Feb 6 18:51 file5
250 List completed successfully.

Command:

• List the members of a partitioned data set containing load modules:


cd 'sys1.linklib'

>>> CWD 'sys1.linklib'
250-Local directory might be a load library
250 "SYS1.LINKLIB" partitioned data set is working directory
Command:
dir d*

>>> PASV
227 Entering Passive Mode (127,0,0,1,4,112)

>>> LIST d*
125 List started OK
Name Size TTR Alias-of AC ------- Attributes ------- Amode Rmode
DD 03DBD8 031506 IRRENV00 01 FO RN RU 31 24
DELDSD 03DBD8 031506 IRRENV00 01 FO RN RU 31 24
DELGROUP 03DBD8 031506 IRRENV00 01 FO RN RU 31 24
DELUSER 03DBD8 031506 IRRENV00 01 FO RN RU 31 24
DG 03DBD8 031506 IRRENV00 01 FO RN RU 31 24
DMOC1001 000710 03370C 00 FO 31 ANY
DMOC1002 000710 03370C 00 FO 31 ANY
DMOC2001 000710 03370C 00 FO 31 ANY
DMOC2002 000710 03370C 00 FO 31 ANY
DMOC3001 000710 03370C 00 FO 31 ANY
DMOC3002 000710 03370C 00 FO 31 ANY
DMOC4001 000710 03370C 00 FO 31 ANY
DMOC4002 000710 03370C 00 FO 31 ANY
DMOC5001 000710 03370C 00 FO 31 ANY
DMOC5002 000710 03370C 00 FO 31 ANY
DMOC6001 000710 03370C 00 FO 31 ANY
DMOC6002 000710 03370C 00 FO 31 ANY
DMOC7001 000710 03370C 00 FO 31 ANY
DMOC7002 000710 03370C 00 FO 31 ANY
DMOC8001 000710 03370C 00 FO 31 ANY
DMOC8002 000710 03370C 00 FO 31 ANY
DMOC9001 000710 03370C 00 FO 31 ANY
DMOC9002 000710 03370C 00 FO 31 ANY
250 List completed successfully.

Command:

• List the members of a partitioned data set from a text library:
Usage

- To make a file group the current working directory, use the CD command. The method you use to specify a directory or file group is host-dependent.
- The Dir subcommand provides a complete list of directory entries and gives additional information about the files.
- You can use special characters for pattern matching when specifying the name. These characters depend on the host FTP server.
- Special characters you can use for the z/OS FTP server:
  - * A single asterisk by itself indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters. An asterisk as the last qualifier will indicate that 0 or more qualifiers can occupy that position.
  - ** A double asterisk indicates that 0 or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters. It must be preceded and followed by either a period or a blank.
  - % A single percent sign by itself indicates that exactly one alphanumeric or national character can occupy that position.
  - %...% One to eight percent signs (%) can be specified in each qualifier.

In DIRECTORYMode, ** is not allowed and % and * may be specified in the last qualifier only.

The following shows examples of how the z/OS FTP server special characters can be used.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSAM.DATA.SET</td>
<td>VSAM.DATA.SET only</td>
</tr>
<tr>
<td>VSAM.DATA.SET%</td>
<td>VSAM.DATA.SET1</td>
</tr>
<tr>
<td>VSAM.DATA.SET%</td>
<td>VSAM.DATA.SET2</td>
</tr>
<tr>
<td></td>
<td>- but not VSAM.DATA.SET30</td>
</tr>
<tr>
<td>VSAM.DATA.SET%</td>
<td>VSAM.DATA.SET30</td>
</tr>
<tr>
<td>VSAM.DATA.SET%</td>
<td>VSAM.DATA.SET31</td>
</tr>
<tr>
<td></td>
<td>- but not VSAM.DATA.SET1 or VSAM.DATA.SET2</td>
</tr>
</tbody>
</table>
### Entry | Returns
--- | ---
VSAM.*.SET | VSAM.DATA1.SET  
VSAM.DATA2.SET  
- but not VSAM.DATA.SET.KSDS
VSAM.*A | VSAM.A  
VSAM.BA  
VSAM.BBA  
- but not VSAM.B  
or | VSAM.AB
VSAM.DATA.* | VSAM.DATA.SET1  
VSAM.DATA.SET2  
VSAM.DATA.SET.KSDS  
- but not VSAM.DATA1.SET
VSAM.DATA* | VSAM.DATA1  
VSAM.DATA23  
- but not VSAM.DATA.SET.KSDS
VSAM.DATA** | VSAM.DATA1  
VSAM.DATA23  
VSAM.DATA.SET1  
VSAM.DATA1.SET  
VSAM.DATA.SET.KSDS  
- but not VSAM1.DATA.SET
**.DATA | VSAM.DATA  
NONVSAM.WORK.DATA  
DATA  
- but not VSAM.DATA.SET
** | Will return all data sets within the current working directory.  
If the current working directory is null, this command has the potential to read all available catalogs to which the user has access. This can take a considerable length of time.

### Context
- See Appendix A, “Specifying data sets and files” on page 383 for more information about pattern matching and about specifying data sets and files.
- To get a list containing only the file names in a directory, use the LS subcommand (see "LS subcommand—Obtain a list of file names" on page 203).
- To make a file group the current working directory, see "CD subcommand—Change the directory on the remote host" on page 140.
- To change the local directory, see "LCd subcommand—Change the local working directory" on page 179.
DUMP subcommand—Set extended trace options

Purpose

Use the DUMP subcommand to enable or disable extended internal tracing.

Note: Extended tracing has the potential to generate a large amount of trace data and should not be set unless requested to debug a specific problem in the code.

Format

```
DUMP
```

Parameters

One or more of the following options can be specified:

? Displays the status of the traces.

n Specifies the ID number of a specific extended trace point that is to be activated in the FTP code. The number has a range of 1–99.

ALL This parameter is used to set all of the trace points.

FSC Activates all of the extended trace points in the file services code. The ID numbers for FSC are 20 to 49.

NONE This parameter is used to turn off all of the traces.

SOC Activates all of the extended trace points in the network services code. The ID numbers for SOC are 50 to 59.

SQL Activates all of the extended trace points in the SQL services code. The ID numbers for SQL are 70 to 79.

Xyyy This syntax is used to turn off (reset) a trace that is named by yyy. For example, DUMP X21 X22 XSQL will reset the extended trace points 21 and 22 and all of the SQL trace points.

Examples

The following is an example of a dump trace.
dump 21 22
Active client dumpIDs - 21 22
get a 'user33.abc111' (replace
12:38:31 MV0456 (21) TU_DSN ....:
0A31D1E4 00020001 00DE4E2 C5D9F3F3 4BC1C2C3 *......USER33.ABC*
0A31D1F4 F1F1F100 00000000 00000000 *1111111111111111*
0A31D204 - 0A31D223 All zeros (0x20 bytes) 12:38:31 MV0512 (21) TU_STATS ....:
0A31D150 00040000 00010100 *......... *
12:38:31 MV0521 (21) TU_DISP ....:
0A31D158 00050000 00000000 *........ *
12:38:31 MV2113 (22) RN PA=
0A3296C0 14070000 00000000 0A3296D8 00000000 *..........oQ....*
0A3296D0 00000000 *...... *
12:38:31 MV2115 (22) RN DD=
0A3296B0 00010000 008E4E2B E2F0F0F0 F0F94B4B *......SYS00009..*
0A3296C0 14070000 0000 *...... *
12:38:31 MV2117 (22) RN DSN=
0A3296EC 00050000 00DE4E2 C5D9F3F3 4BC1C2C3 *......USER33.ABC*
0A3296FC F1F1F14B 60614B4B 4B4B4B4B 4B4B4B4B *1111111111111111*
0A32970C 6C6D6E6F 4B4B4B4B 4B4B4B4B 4B4B7A7B *%_>?..........:#*
0A32971C 7C7D7E4B *@'=. *
>>> PORT 9,67,113,57,4,59
200 Port request OK.
>>> RETR a
125 Sending data set /u/user33/a
250 Transfer completed successfully.
820 bytes transferred in 0.005 seconds. Transfer rate 164.00 Kbytes/sec

Usage

The setting of the traces is additive. This is demonstrated by the following example:

dump none 21
EZA2850I Active dumpIDs: 21

dump 22
EZA2850I Active dumpIDs: 21 22

Entering dump with no parameters is the same as entering dump with the ?
parameter.

The range of 99 extended trace points is defined to allow easy extension of the
trace points by the TCP/IP service team. Additional trace points can be added to the
code without any changes to the external mechanism to control the traces.

Refer to [z/OS Communications Server: IP Diagnosis](https://www.ibm.com/support/docview/go.do?uid=swg24036836) for the definition of the extended trace points.
**EBcdic subcommand—Change the data transfer type to EBCDIC**

**Purpose**

The EBcdic subcommand enables you to change the data transfer type to EBCDIC.

**Format**

```
> [Z(qOOOO] [Z(qOOOO
```

**Parameters**

There are no parameters for this subcommand.

**Usage**

Use the EBcdic subcommand to direct FTP to transfer data with no translation.

**Context**

For more information about transfer methods, see [Table 11 on page 51](#).
EUckanji subcommand—Change the data transfer type to EUCKANJI

Purpose

Use the EUckanji subcommand to change the data transfer type to Extended UNIX Code (EUC) kanji.

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the SIte/LOCSIte SBDataconn or SIte XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

```
 EUckanji
```

Parameters

**Sosi**
Transferred data contains the shift-out and shift-in characters specified by one of the following parameters – Ascii, Ebcdic or Space. If no parameter is specified, ASCII is used as the default.

If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

**Ascii**
When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

**Ebcdic**
When combined with the Sosi parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

**Space**
When combined with the Sosi parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

**NOSo**
Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to and from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

**NOType**
Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.
Usage

The EUckanji client subcommand is equivalent to the TYPE B 2 server command.

Context

See "FTP with DBCS support on page 123" for more information.
FEature subcommand—Query FTP server for features it supports

Purpose
Ask the FTP server which features it supports. FTP clients use this command to determine which languages the server supports, and which features the server supports.

Format
```
uestra
```

Parameters
There are no parameters for this subcommand.

Usage
The minimum abbreviation for the feature subcommand is fe.
File subcommand—Set the file structure to File

Purpose

Use the File subcommand to set the file structure to File. This is equivalent to specifying the STRucture F subcommand. See “STRucture subcommand—Set the file structure” on page 270 for more information.

Format

```
>File
```

Parameters

There are no parameters for this subcommand.
Get subcommand—Copy files

Purpose

Use the Get subcommand to copy a file from the remote host to your local host.

Format

```plaintext
Get foreign_file local_file (REPLACE)
```

Parameters

- `foreign_file`
  Specifies the name of the file to be retrieved from the remote host.

- `local_file`
  Specifies the name of the local file created as a result of the Get subcommand.

  If the current local working directory is a PDS, `local_file` is the name of the member in the PDS. If the current local working directory is a data set prefix, the local file is a sequential data set with the `local_file` name appended to the current local working directory. If the current local working directory is an HFS directory, the local file is an HFS file in that directory.

  You can override the usage of the current local working directory in the local file name by specifying `local_file` as a complete data set name enclosed in single quotation marks ('). If `local_file` is not specified, the `local_file` name is the same as the `foreign_file` name.

- (REPLACE)
  Causes a data set on your local host to be overwritten, if it already exists. If the data set already exists, and you do not use the replace parameter, the existing data set is not overwritten. A message informing you of this is displayed.

  If the data set already exists and you specify REPLACE, the data in the file is overwritten, but not reallocated. This means the local data set retains its existing characteristics.

Usage

- FTP uses either the characteristics of the local file, if it exists, or uses the values specified with the LOCSTite subcommand. Characteristics of the transmitted (foreign file) data set are unknown.

  When you use the Get subcommand, FTP may truncate data records and you might lose data, if one of the following occurs:

  - If you are creating a new data set at the client and the value of LRecl, as shown by the LOCSTat command, is a value less than the LRecl of a received data set, then FTP truncates the received data set.
  - If the data set name already exists at the receiving site and the logical record length (LRecl) of the data set at the receiving site is less than the LRecl of the transmitted data set, then FTP truncates the transmitted data set.

  You could also encounter truncated data records or lost data when you use Get with the REPLACE option.

  **Attention:** A Get subcommand issued for an empty or nonexistent foreign file erases the contents of the existing local data set.
– If FTP does not support directory content transfers in partitioned data sets, it is not possible to FTP load modules.

– If the data set is migrated, it will be replaced regardless of the replace option.

- If the name specified for `local_file` is not acceptable to your local host, the file is not transferred.
- To get a file from the remote host, you must have a defined working directory on that host and you must have read privileges to the files in this working directory.
- If the data set has been preallocated, you must specify DSORG=PS on the DCB statement in the JCL.

**Context**

- See [Appendix A, “Specifying data sets and files” on page 383](#) for more information about naming conventions.
- See “CD subcommand—Change the directory on the remote host” on page 140 and “ACCOUNT subcommand—Supply account information” on page 131 for more information about working directories.
- Get can be used with the PROXY subcommand to transfer files from a host on a primary connection to a host on a secondary connection. See [“PROXY subcommand—Execute FTP subcommand on secondary control connections” on page 222](#) for more information.
GLob subcommand—Toggle expansion of metacharacters

Purpose

Use the GLob subcommand to toggle globbing (the expansion of metacharacters in file names) for the MDelete, MGet, and MPut subcommands.

Format

```
GLob
```

Parameters

There are no parameters for this subcommand.

Examples

Assume that the files m1 and m1* exist in the directory /u/user33/mpp1.

```
Command:
pwd
>>> PWD
257 "/u/user33/mpp1" is the HFS working directory

Command:
lpwd
Local directory name set to hierarchical file /u/user33

Command:
prompt
Interactive mode is off

Command:
mget m1*
>>> PORT 9,67,113,57,4,43
200 Port request OK.
>>> NLST m1*
125 List started OK
250 List completed successfully.
>>> PORT 9,67,113,57,4,44
200 Port request OK.
>>> RETR m1
125 Sending data set /u/user33/mpp1/m1
250 Transfer completed successfully.
200 bytes transferred in 0.050 seconds. Transfer rate 4.00 ...
Kbytes/sec.

>>> PORT 9,67,113,57,4,45
200 Port request OK.
>>> RETR m1*
125 Sending data set /u/user33/mpp1/m1*
250 Transfer completed successfully.
200 bytes transferred in 0.020 seconds. Transfer rate 10.00 ...
Kbytes/sec.
```
With globbing off, at most one file will match the pattern. Also, the NLST command is not sent to look for pattern matches.

Usage

Glob acts as a toggle that turns metacharacter expansion on or off. By default, Glob is on.

Context

For more information on globbing, refer to z/OS UNIX System Services User's Guide.
HAngeul subcommand—Change the data transfer type to HANGEUL

Purpose

Use the HAngeul subcommand to change the data transfer type to Hangeul.

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the Site/LOCSite SBDDataconn or Site XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

```
>>HAngeul (Ascii NOSo NOType Sosi Ebcdic Space)
```

Parameters

**Sosi**

Transferred data contains the shift-out and shift-in characters specified by one of the following parameters – Ascii, Ebcdic or Space. If no parameter is specified, ASCII is used as the default.

If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

**Ascii**

When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

**Ebcdic**

When combined with the Sosi parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

**Space**

When combined with the Sosi parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

**NOSo**

Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

**NOType**

Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The HAngeul client subcommand is equivalent to the TYPE B 5 server command.
Context

See "FTP with DBCS support" on page 123 for more information.
HElp and ? subcommands—Display help information

Purpose

Use the HElp command to get assistance with the FTP subcommands.

Format

```
HElp
```

Parameters

- **All**
  The client displays a description of all the subcommands it implements.

- **PROXY**
  When used, displays a description of all subcommands available on the proxy subcommand.

- **subcommand**
  Displays a description of the specified subcommand. The subcommand name can be abbreviated to its minimum abbreviation.

- **Server**
  Displays the help that the foreign host offers for the specified subcommand.
  
  If you do not specify a subcommand, FTP displays a list of the commands that the foreign host recognizes.

Usage

- If you enter the HElp subcommand without a parameter, you see the HElp FTP MENU, which lists the subcommands recognized by the FTP client and a description of the help information available.

- If you enter the ? subcommand by itself, you see introductory information about FTP.

Note: To receive help from a server on a secondary control connection, enter `PROXY HElp SERVER`. See “PROxy subcommand—Execute FTP subcommand on secondary control connections” on page 222 for more information.
Ibmkanji subcommand—Change the data transfer type to IBMKANJI

Purpose

Use the Ibmkanji subcommand to change the data transfer type to IBM kanji.

Format

```
Ibmkanji (NOType)
```

Parameters

- **(NOType)**
  
  Suppresses sending of the TYpe command for host servers that do not support this data transfer type.

Usage

This subcommand causes no conversion to be performed on the transferred file. It has exactly the same effect as the EBcdic TYpe command alias.

Context

See “FTP with DBCS support” on page 123 for more information.
JIS78kj subcommand—Change the data transfer type to JIS78KJ

Purpose

Use the JIS78kj subcommand to change the data transfer type to JIS78KJ (1978 edition).

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the SIte/LOCSIte SBDDataconn or SIte XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

```
     JIS78kj
     (Ascii
        Jisroman
         NOSo
        NOType
     )
```

Parameters

Ascii
Use ASCII shift-in escape sequence ESC ( B in the transferred data.
If neither Ascii nor Jisroman is specified, the ASCII shift-in sequence is used.

Jisroman
Use Jisroman shift-in escape sequence ESC ( J in the transferred data.

NOSo
Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

NOType
Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

- The JIS78kj or JIS78kj (ASCII client subcommands are equivalent to the TYPE B 4 A server command.
- The JIS78kj (JISROMAN client subcommand is equivalent to the TYPE B 4 R server command.
- The JIS78kj (JISROMAN NOSO client subcommand is equivalent to the TYPE B 4 R N server command.

Context

See [FTP with DBCS support](#) on page 123 for more information.
JIS83kj subcommand—Change the data transfer type to JIS83KJ

Purpose

Use the JIS83kj subcommand to change the data transfer type to JIS83KJ (1983 edition).

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the Site/LOCSIte SBDataconn or Site XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

```
JIS83kj

Ascii

Jisroman

Noso

NOType
```

Parameters

Ascii

Use ASCII shift-in escape sequence ESC ( B in the transferred data.

If neither Ascii nor Jisroman is specified, the ASCII shift-in sequence is used.

Jisroman

Use Jisroman shift-in escape sequence ESC ( J in the transferred data.

NOSo

Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

NOType

Suppresses the sending of the corresponding Type command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

- The JIS83kj or JIS83kj (ASCII client subcommands are equivalent to the TYPE B 3 A server command.
- The JIS78kj (JISROMAN client subcommand is equivalent to the TYPE B 3 R server command.

Context

See “FTP with DBCS support” on page 123 for more information.
Ksc5601 subcommand—Change the data transfer type to KSC-5601

Purpose

Use the Ksc5601 subcommand to change the data transfer type to KSC-5601.

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the Site/LOCSite SBDdataconn or Site XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

\[ \text{Ksc5601} \]

Parameters

Sosi
Transferred data contains the shift-out and shift-in characters specified by one of the following parameters — Ascii, Ebcdic or Space. If no parameter is specified, ASCII is used as the default.

If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

Ascii
When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

Ebcdic
When combined with the Sosi parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

Space
When combined with the Sosi parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSo
Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

NOType
Suppresses the sending of the corresponding TYpe command to the server.

Use this parameter when translation is to be done by the FTP client only.
Usage

The Ksc5601 client subcommand is equivalent to the TYPE B 6 server command. See "FTP with DBCS support" on page 123 for more information.
LAnguage subcommand—Set the language used for FTP replies from the server

Purpose
Request the language to be used for FTP server replies, or reset the language in use to the server’s default language.

Format
```
LANG language -subtag
```

Parameters
- `language`
  A 2-character code identifying the language to be used for FTP replies. Valid codes are defined in RFC 1766.
- `-subtag`
  A 2-character code identifying a variation of language. Valid codes are defined in RFC 1766.

Usage
- The language subcommand is equivalent to the LANG `language -subtag` command.
- LAnguage with no parameters sets the language for FTP replies to the server’s default language. For most FTP servers, the default is US English encoded in 7-bit ASCII.
- LAnguage with a parameter requests the server to use `language` for FTP replies. As specified in RFC 2640, it also directs the server and client to use UTF-8 encoding of pathnames on the control connection.
- This subcommand is not available unless EXTENSIONS UTF8 is encoded in FTP.DATA.
- The subcommand is not available if you have disabled UTF-8 encoding with a LOCSItte command or FTP start option.

Context
The client should use the FEature subcommand to determine which languages the server offers prior to using the LAnguage subcommand to request a language for FTP replies. See “FEature subcommand—Query FTP server for features it supports” on page 164 for information on using the FEature subcommand.
**LCd subcommand—Change the local working directory**

**Purpose**

Use the LCd subcommand to change the current working directory on the local host.

**Format**

- **In a TSO Environment:**
  
  ```
  LCd qualifier
  ```

- **In a z/OS UNIX System Services Environment:**
  
  ```
  LCd qualifier
  ```

**Parameters**

*qualifier*

Specifies either a common prefix for a group of sequential data sets or the qualifiers of a PDS.

**Note:** In a z/OS UNIX environment, you can omit the qualifier on the LCd subcommand. Doing so changes the current working directory to your home directory. If you do not have a home directory, the working directory is not changed, and no message is issued.

**Examples**

- **Change the local current working directory:**
  
  ```
  lcd ftp.test1
  ```

  Local directory name set to partitioned data set USER14.FTP.TEST1.

  **Command:**

  ```
  ```

- When the LCd subcommand is issued, *qualifier* is appended to the current local working directory. For example, if the current local working directory is TCPUSR14.TEST and you issue the LCd subcommand LCD FILES, the new working directory becomes TCPUSR14.TEST.FILES.

- To override the existing directory rather than append to the directory, issue the *qualifier* in single quotation marks (’). For example, if the current local working directory is TCPUSR14.TEST and you issued the LCd subcommand LCD ’FTP.FILES’, the new working directory is FTP.FILES.

- If a PDS exists with the exact name of the current local working directory, FTP considers the working directory to be that PDS. Otherwise, FTP considers the working directory to be a common prefix qualifier for sequential data sets. If a PDS exists with the same name as the current local working directory, but you want the current local working directory to be treated as a common prefix for sequential data sets, specify the working directory with a period (.) at the end. For example, if a PDS named TCPUSR14.TEST exists, the subcommand LCD ’TCPUSR14.TEST’ makes the PDS TCPUSR14.TEST the current local working directory. A subsequent Get command used to copy data set name1 would add the member name1 to the TCPUSR14.TEST PDS. In contrast, the statement LCD
'TCPUSR14.TEST.' would make the current local working directory TCPUSR14.TEST., a prefix for sequential data sets. A subsequent Get command used to copy data set name1 would create the sequential data set TCPUSR14.TEST.name1.

- To back up one level of the current local working directory, issue the LCD subcommand with two periods (..) at the end. For example, if the working directory is jones.source, the statement LCD .. makes jones. the working directory.

**Testing throughput with *DEV.NULL**

You can use the Get subcommand to copy a large number of files (or one large file) without having the files actually stored in the client’s file system. This is useful for testing purposes because you do not have to worry about allocating the disk space on the client system.

To use this function, first change the working directory to *DEV.NULL by entering the following subcommand:

LCD *DEV.NULL

This affects the working directory only for Get commands. You then use the Get command to copy the file to the client system. The input data set must be valid, and the output file can either be new or already exist. In either case, the file is not actually stored.

The following response shows information such as the number of bytes transferred and the rate of transfer.

| Command: | lsd *dev.null |
| Working Directory for GET is NULL Device |
| for PUT is HFS directory /tmp |
| Command: |
| get 'user2.junk(junk)' example |
| >>> EPSV |
| 229 Entering Extended Passive Mode (|||1034|) |
| >>> RETR 'user2.junk(junk)' |
| 125 Sending data set USER2.JUNK(JUNK) |
| 250 Transfer completed successfully. |
| 65 bytes transferred in 0.070 seconds. Transfer rate 0.93 Kbytes/sec. |

To end the use of the *dev.null directory for Get commands, issue another change local working directory subcommand.

**Usage**

When you enter an FTP session, the working directory on the local host is set according to the environment in which the FTP client is invoked: $HOME in z/OS UNIX, your MVS user ID in TSO.
LMkdir subcommand—Create a directory on the local host

Purpose

Use the LMkdir subcommand to create a PDS (or HFS directory) on the local host. This subcommand provides you with an easy way to create a directory in the local host for data transfer.

Format

\[ \text{LMkdir} \text{ data_set} \]

Parameters

data_set

Specifies the name of the directory to be created.

Examples

In this example, before LMkdir is issued, the local host had the following data sets:

- MVSUSER.ISPF.ISPPROF
- MVSUSER.JCL.CNTL
- MVSUSER.SMFTEST
- MVSUSER.TCPIP.DATA

After the LMkdir was issued, the local host had the following data sets under MVSUSER:

- MVSUSER.FTP.EXAMPLE
- MVSUSER.ISPF.ISPPROF
- MVSUSER.JCL.CNTL
Usage

- FTP provides no subcommand to display a list of local directory entries. You should use TSO ISPF facility to check whether the directory is created by the LMkdir subcommand.
- If you are running FTP in a z/OS UNIX environment, you can use the ! command to check the status of HFS directories.
- The `data_set` value is appended to the local current working directory to form the name of the created PDS. To override the local current working directory, specify: `data_set`

The data set characteristics of the newly allocated PDS are determined by the settings of the local site variables. See ![LOCSite subcommand—Specify site information to the local host](#) and ![LOCStat subcommand—Display local status information](#) for information on setting and displaying the data set characteristics.
LOCSIte subcommand—Specify site information to the local host

Purpose

Use the LOCSIte subcommand to specify information that is used by the local host to provide services specific to that host system.

Format

```
LOCSIte options
```

options:

- **ASAtrans**
- **AUTOMount**
- **AUTORecall**
- **BLKsize** = size
- **Blocks**
- **BLOCKSsize** = size
- **BUfno** = number
- **CHKptint** = number
- **CHKPTrefix**
  - home
  - local
- **CHMod**
  - ooo filename
  - symbolic filename
- **CONDdisp**
  - Catalog
  - Delete
- **CTRLConn**
  - 7BIT
  - iconv_ascii
  - FTP_STANDARD_TABLE
- **CYlinders**
- **DATAClass** = data_class
- **DATASETmode**
- **DB2** = db2_name
- **DBdsn** = data_set_name
- **Directory** = size
- **DIRECTORYMode**
- **ENCODING** = SBCS, MBCS
<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILEtype</td>
<td>type</td>
</tr>
<tr>
<td>FWFriendly</td>
<td></td>
</tr>
<tr>
<td>ISPFSTATS</td>
<td></td>
</tr>
<tr>
<td>LRec</td>
<td>length</td>
</tr>
<tr>
<td>MBDATACONN</td>
<td>(file_system_codepage, network_transfer_codepage)</td>
</tr>
<tr>
<td>MGmtclass</td>
<td>mgmtclass</td>
</tr>
<tr>
<td>MIGratevol</td>
<td>volume_ID</td>
</tr>
<tr>
<td>NOASAtm</td>
<td></td>
</tr>
<tr>
<td>NOAUTOMount</td>
<td></td>
</tr>
<tr>
<td>NOAUTOREcall</td>
<td></td>
</tr>
<tr>
<td>NOFWFriendly</td>
<td></td>
</tr>
<tr>
<td>NOISPFSTATS</td>
<td></td>
</tr>
<tr>
<td>NOQUOTESoverride</td>
<td></td>
</tr>
<tr>
<td>NORDW</td>
<td></td>
</tr>
<tr>
<td>NORESTGet</td>
<td></td>
</tr>
<tr>
<td>NOSBSUB</td>
<td></td>
</tr>
<tr>
<td>NOSPRead</td>
<td></td>
</tr>
<tr>
<td>NOTRAILINGblanks</td>
<td></td>
</tr>
<tr>
<td>NOTRUNcate</td>
<td></td>
</tr>
<tr>
<td>NOUCSSUB</td>
<td></td>
</tr>
<tr>
<td>NOUCSTRUNC</td>
<td></td>
</tr>
<tr>
<td>NOWRAPRecod</td>
<td></td>
</tr>
<tr>
<td>NOWRATAPEFastio</td>
<td></td>
</tr>
<tr>
<td>PRimary</td>
<td>amount</td>
</tr>
<tr>
<td>Qdisk</td>
<td>volume_serial</td>
</tr>
<tr>
<td>QUOTESoverride</td>
<td></td>
</tr>
<tr>
<td>READTAPEFormat</td>
<td>tape_format</td>
</tr>
<tr>
<td>RDW</td>
<td></td>
</tr>
<tr>
<td>RECfm</td>
<td>format</td>
</tr>
<tr>
<td>RESTGet</td>
<td></td>
</tr>
<tr>
<td>RETpd</td>
<td>days</td>
</tr>
</tbody>
</table>
Parameters

ASAtrans
Permits the FTP client to interpret characters in the first column of ASA files being transferred as print control characters.

AUTOMount
Permits automatic mounting of volumes for data sets on volumes that are not mounted. If AUTOMount is specified and an unmounted volume is needed, a message is automatically issued to the MVS operator console requesting that the volume be mounted. The MVS operator must then mount the volume and reply to the message before FTP can proceed.

AUTOREcall
Permits automatic recall of migrated data sets.

BLKsize
Specifies the block size of a newly allocated data set. BLKsize is functionally equivalent to BLOCKSsize. The BLOCKSsize parameter is obsolete, but it is accepted to provide compatibility with previous releases of z/OS TCP/IP.

When specified without a size, no block size is used when allocating the new data set. When specified without a size, the equal sign (=) is optional.
Specify BLKsize with no value if you are also specifying \texttt{DATAClass=\text{data\_class}} and you want the SMS data class to provide the BLKsize value, or if you are specifying \texttt{DCbdsn=\text{data\_set\_name}} and you want to use the block size from the DCBDSN data set. If BLKsize=\text{size} is specified with either the DATAClass or DCbdsn parameter, the value specified by the LOCSite BLKsize parameter overrides the DATAClass or DCbdsn block size.

**Notes:**

1. If you specify BLKsize without a size, FTP does not specify the block size when allocating new data sets.
2. Be especially careful specifying both BLKsize= and Blocks. While there are conditions where this is tolerated, if a valid BLKsize cannot be determined, the data set will not be created when the allocation is attempted.

\textit{size}

Specifies the block size of a newly allocated data set. The valid range is 0–32 760.

BLKsize=0 is a special case. When BLKsize=0 is specified, the operating system attempts to determine a block size for the new data set. FTP does not create the new data set unless the system is able to establish a nonzero block size.

\textit{Blocks}

Specifies that primary and secondary space allocations are in blocks.

If both PRImary and SECondary are specified as 0, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the Blocks parameter is ignored.

\textit{BLOCKS\text{size}}

Specifies the block size of a newly allocated data set. BLOCKS\text{size} is functionally equivalent to BLKsize. BLOCKS\text{size} is obsolete but it is accepted to provide compatibility with previous releases of z/OS TCP/IP. See the BLKsize parameter for more information.

\textit{BUfno}

Specifies the number of access method buffers that is used when data is read from or written to a data set. The valid range is 1–35. The default value is 5.

\textit{CHKptint}

Specifies the checkpoint interval for the sending site in a file transfer request. If the checkpoint interval is 0, no checkpointing occurs and no marker blocks are transmitted. The default value is 0.

CHKptint must be set by LOCSite when the client site is the sending site for the file transfer (PUT command). Do not increase the checkpoint interval above 0 for a PUT unless the receiving site supports the RESStart command and is able to process checkpoint markers in the file transfer data stream.

CHKptint is valid only for block and compressed modes of data transfer.

\textit{number}

Specifies the checkpoint interval for the sending site in a file transfer request. This value is used to determine when checkpoint marker blocks are to be transmitted so that transmission can be restarted based on the information in the last marker.
A large checkpoint interval means that a large amount of data is sent between markers and therefore few markers are sent. A smaller checkpoint interval means that less data is sent in between markers and therefore more markers are sent.

The costs involved with using a nonzero checkpoint interval are:

- The markers themselves are transmitted, which means more bytes are being sent across the network (approximately 44 bytes per marker).

- For each marker that is sent by the client, a reply must be sent by the server on the control connection. The reply acknowledges the marker and provides the server’s corresponding marker.

To estimate the appropriate checkpoint interval, use the following formula. You need to know the record length of the file you are transferring and how much data you think can be transmitted reliably.

\[
\text{CHKPTINT} = \frac{\text{amount of data in interval}}{\text{record length of the file}}
\]

Do not execute a CHKptint more often than once every 200KB. For example, if the file you are transferring has 80-byte records:

\[
\text{CHKPTINT} = \frac{200\,\text{KB}}{80\,\text{bytes}} = \frac{200 \times 1024\,\text{bytes}}{80\,\text{bytes}} = 2560
\]

**CHKPTPrefix**

Specifies a key which is used to determine the hlq of the checkpoint data set. The name will be hlq.FTP.CHKPOINT or hlq.pds_name(CHKPOINT).

- **home**
  - The default.

- **userid**
  - Creates the data set 'userid.FTP.CHKPOINT'.

- **local**
  - Creates a file named current_path.FTP.CHKPOINT or, if the path happens to be a PDS, then the file name is current_path.pds_name(CHKPOINT).

Note the exceptions for z/OS UNIX and BATCH jobs in the following chart using 'path' for current path and 'FN' for FTP.CHKPOINT:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>z/OS UNIX</th>
<th>Batch</th>
<th>TSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>path.FN</td>
<td>userid.FN</td>
<td>tso_prefix.FN</td>
</tr>
<tr>
<td>Userid</td>
<td>userid.FN</td>
<td>userid.FN</td>
<td>userid.FN</td>
</tr>
<tr>
<td>Local</td>
<td>path.FN</td>
<td>path.FN</td>
<td>path.FN</td>
</tr>
<tr>
<td>Local with path=pdse</td>
<td>N/A</td>
<td>path.pds_na</td>
<td>path.pds_na</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(CHKPOINT)</td>
<td>(CHKPOINT)</td>
</tr>
</tbody>
</table>

**CHMod**

Changes the permission bits for a file.

- **ooo filename**
  - ooo is the octal mask and filename is the name of the file whose mask is to be changed.

- **symbolic filename**
  - symbolic is a value in the format accepted by the z/OS UNIX CHMOD command (for example, a+r).
**Note:** Only r, w, and x permission bits are accepted in *symbolic*.

If *filename* does not begin with a slash character (/), it is appended to the current working directory. If *filename* does begin with a slash character (/), it is interpreted as a complete directory name.

The file name specified must be an HFS file name. The setting of QUOTESoverride is ignored and all quotation marks are treated as part of the file name.

The CHMOD keyword must be the only or last keyword on a LOCSItle subcommand.

**CONDdisp**
Specifies the disposition of the data set if a retrieve operation for a new data set ends before all of the data is written.

**Catlg**
Specifies that a data set is kept and cataloged when an FTP file transfer ends prematurely.

**Delete**
Specifies that a data set is deleted when an FTP file transfer ends prematurely.
Delete is ignored if the file transfer failed as a result of the FTP client being terminated or if the client has received checkpoint information during data transfer.

**CTRLConn**
Specifies the ASCII code page to be used for control connections. The valid subcommands are:

LOCSITE CTRLConn=7BIT
LOCSITE CTRLConn=iconv_ascii
LOCSITE CTRLConn=*

7BIT
Indicates 7-bit ASCII is to be used.

iconv_ascii
Is a name recognized by iconv to indicate an ASCII code page. For a list of code pages supported by iconv, refer to code set converters in the [z/OS C/C++ Programming Guide](https://www.ibm.com/support/docview/it/62969722). 

**FTP_STANDARD_TABLE**
Indicates that the FTP internal tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), are to be used on the control connection.

* Indicates that the ASCII used at initialization is to be used.

**Note:** Setting the control connection code page using LOCSItle CTRLCONN disables UTF-8 encoding of the control connection. You must start the client again to restore UTF-8 encoding.

**CYlinders**
Specifies that primary and secondary space allocations are in cylinders.

If both PRIMARY and SECONDARY are specified as 0, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the CYlinders parameter is ignored.
**DATAClass**
Specifies the SMS data class, as defined by your organization, for the target host. Specifying DATAClass with no parameter value cancels the dataclass specification. The equal sign (=) is optional in this case.

See “Specifying values for new data sets” on page 86 for more information about specifying attributes when allocating new data sets.

**data_class**
Specifies the SMS data class, as defined by your organization, for the target host. If values are specified for any of the following LOCSIte parameters, the values specified by the LOCSIte parameter override the value specified in the SMS dataclass:
- BLKsize
- Directory
- LRecl
- PRImary
- RECfm
- RETpd
- SECondary

If the DCbdsn parameter is specified on the LOCSIte subcommand, the LRecl, RECfm, BLOCKS|ize, and RETpd (if specified) of the DCBDSN data set overrides the values specified in the data class.

If the MGmtclass parameter is specified on the LOCSIte subcommand, and the requested management class specifies a retention period, the retention period value of the management class may override the retention period value of the dataclass.

**DATASetmode**
Specifies that all the data set qualifiers below the current directory are treated as entries in the directory (disables DIRECTORYMode).

**DB2**
Specifies the name of the DB2 subsystem.

**db2_name**
The name of the DB2 subsystem.

**DCbdsn**
Specifies the name of the MVS data set to be used as a model for allocation of new data sets. Specifying DCbdsn with no parameter value cancels the DCbdsn specification.

**data_set_name**
Specifies the name of the data set. The file name must be an MVS data set name. HFS file names are not allowed on the DCbdsn parameter. The setting of QUOtesoverride is ignored. If the file name is enclosed in single quotation marks, it overrides the current working directory; otherwise it is appended to the current working directory.

**Notes:**
1. Specify LOCSIte RECfm, LRecl, and BLKsize parameters with no values to allow characteristics from the model DCB to be used.
2. To override the model characteristics of RECfm, LRecl, BLKsize, or RETpd, specify a value on the LOCSIte command.
3. If MGmtclass is specified, the RETpd value of the MGmtclass may override the RETpd value.

Specifying a GDG data set with a relative index produces an error message. The following examples are unsupported specifications:

LOCSITE DCBDSN=MYGDG(0)
LOCSITE DCBDSN=MYGDG(–nnn) or
LOCSITE DCBDSN=MYGDG(+nnn)

See "Steps for using a DCBDSN model to create a new data set" on page 87 for more information about DCBDSN.

Directory
Specifies the number of directory blocks to be allocated for the directory of a PDS.

Specify Directory=0 to allocate without specifying the number of directory blocks. Specify Directory=0 when you are also specifying DATAClass=dataclass and you want the SMS dataclass to provide the Directory size.

The size you specify with the Directory parameter overrides the DATAClass directory specification.

size
Specifies the number of directory blocks to be allocated for the directory of a PDS. The valid range is 1–16 777 215. A size of 0 indicates that the directory blocks should be taken from the SMS data class.

DIRECTORYMode
Specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory. In directory mode, this data set qualifier is the only one used by the MPut subcommand.

DIRECTORYMode has no effect on files residing in an HFS.

ENCODING
Indicates the kind of encoding that is used for conversions between codepages for data transfers.

SBCS
Single Byte encoding. Code pages are specified using the SBDATACONN statement. This is the default value.

MBCS
Multibyte encoding. Code pages are specified using the MBDATACONN statement.

FILEtype
Specifies the file type of the data set.

type
The file type of the data set can be:

Type     Description
SEQ      Sequential or partitioned data sets
SQL      SQL query function

FWFriendly
Specifies that the FTP client is firewall-friendly. This means that data connections will be set up from the FTP client to the FTP server.
**Note:** When the FTP server has an IPv6 address, data connections are always set up from the FTP client to the FTP server without reference to the FWFriendly setting.

**ISPFSTATS**
Allows FTP to create or update ISPF Member statistics when Get or MGet subcommands are issued.

**LRecl**
Used to specify the logical record length (LRecl) of a newly allocated data set.

Specify LRecl with no value when you are also specifying DATAClass=\textit{data\_class} and you want the SMS dataclass to provide the LRecl value, or when you are specifying DCbdsn=\textit{data\_set\_name} and you want to use the LRecl from the DCBDSN data set. If LRecl=\textit{length} is specified with either DATAClass or DCbdsn, the length specified by the LOCSIte LRecl parameter overrides the DATAClass or DCbdsn LRecl.

\textit{length}
Specifies the logical record length of a newly allocated data set. The valid range is 0–32 760. A special value of x (LRecl=x) is also supported to indicate that a logical record length can exceed 32 760 for variable-length spanned records.

Specifying LRecl=0 has the same effect as specifying LRecl with no parameters.

**MBDATACONN=\textit{(file\_system\_codepage, network\_transfer\_codepage)}**
Specifies the codepages for the file system and for the network transfer used when data is translated during a data transfer.

\textit{file\_system\_codepage}
Specifies the name of the file system codepage. The supported codepages are IBM-1388 and UTF-8.

\textit{network\_transfer\_codepage}
Specifies the network transfer code page. The supported codepage is IBM-5488.

**MGmtclass**
Used to specify the SMS management class as defined by your organization for the target host. Specifying MGmtclass with no \textit{mgmtclass} cancels the mgmtclass specification. The equal sign (=) is optional in this case.

\textit{mgmtclass}
Specifies the SMS management class as defined by your organization for the target host. If the mgmtclass specified has a setting for RETpd, the value specified by the mgmtclass may override the setting of the LOCSIte RETpd parameter, the RETpd value of a model data set if the DCbdsn parameter is specified, and the RETpd value defined in an SMS data class if DATAClass is specified. See “Specifying values for new data sets” on page 86 for more information about specifying attributes when allocating new data sets.

**MIGratevol**
Specifies the volume ID for migrated data sets if they do not use IBM storage management systems. If you do not specify MIGratevol, the default \textit{volume\_serial} is MIGRAT.

\textit{volume\_ID}
The volume ID for migrated data.
NOASAtrans
Treats ASA file transfers as regular file transfers; that is, the ASA characters are treated as part of the data and are not converted to print control characters.

NOAUTOMount
Prevents automatic mounting of volumes for data sets on volumes that are not mounted.

NOAUTORecall
Prevents automatic recall of migrated data sets.

Note: A migrated data set can be deleted even though NOAUTORecall is specified, because migrated data sets are not recalled for deletion.

NOFWFriendly
Specifies that the FTP client is not firewall friendly. This means that data connections will be set up from the FTP server to the FTP client. This is the default behavior for FTP data connections.

Note: When the FTP server has an IPv6 address, data connections are always set up from the FTP client to the FTP server without reference to the FWFriendly setting.

NOISPFSTATS
Does not allow FTP to create or update ISPF Member statistics when Get or MGet subcommands are issued.

NOQUOtesoverride
A single quote at the beginning of the file name, as well as all other single quotation marks contained in the file name, is treated as part of the actual file name. The entire file name, including the leading single quotation mark, is appended to the current working directory.

NORDW
Specifies that variable record descriptor words (RDWs) are discarded during FTP transmission of variable format data sets. This applies to transfers in stream mode only.

NORESTGet
Prevents opening the checkpoint data set for a Get request. Thus, checkpoint will not be active or recognized. Using this parameter when opening the checkpoint data set might cause a problem.

NOSBSUB
Specifies that substitution is not allowed for data bytes that cannot be translated in a single-byte character translation. This causes a data transfer failure if a character cannot be translated during the transfer.

NOSPRead
Specifies that the output is in report format rather than spreadsheet format when the file type is SQL.

NOTRAILingblanks
Specifies that the FTP client does not preserve the trailing blanks that are in a fixed format data set when the data is sent to a foreign host.

NOTRUNcate
Specifies that truncation is not permitted. The FTP client will set an error and fail file transfer if a record that is longer than LRECL of the new file is detected.
**Note:** If WRAPRECORD is set then the data is wrapped, not truncated, no error will be set, and the file transfer will continue.

**NOUCSSUB**
In Unicode-to-EBCDIC conversion, the data transfer is terminated if any Unicode character cannot be converted into the EBCDIC code set.

**NOUCSTRUNC**
In Unicode-to-EBCDIC conversion, truncation of EBCDIC data is not allowed. The data transfer is aborted if the logical record length of the receiving data set is too small to contain the data after conversion to EBCDIC.

**Note:** The setting of the CONDdisp parameter determines what happens to the target data set if the transfer is aborted.

**NOWRAPrecord**
Indicates that data is truncated if no new line character is encountered before the logical record length of the receiving file is reached.

**NOWRTAPEFastio**
Indicates that ASCII Stream data that is being written to tape must be written using the LE Run-Time Library.

**PRImary**
Used to specify the amount of tracks, blocks, or cylinders for primary allocation. When specified with a value of 0, no primary value is used when allocating the data set.

Specify PRImary with 0 when you are also specifying DATAClass=\texttt{data\_class} and you want the SMS data class to provide the PRImary amount.

To enable the SMS data class to determine the space allocation, both PRImary and SECondary must be specified as 0. The tracks, blocks, cylinders setting is ignored in this case. If PRImary with \texttt{amount} not equal to 0 is specified with DATAClass, the value specified by the LOCSIt parameter overrides the DATAClass space allocation.

\texttt{amount}
Specifies the amount of tracks, blocks, or cylinders for primary allocation. For allocating partitioned data sets, this is the amount that is allocated for the primary extent.

For allocating sequential data sets this is the maximum amount that is allocated for the primary extent. If a lesser amount is needed to hold the data being transferred, only the amount actually needed to hold the data is allocated. The valid range is 1–16,777,215.

**Qdisk**
Used to display statistics about available space on a volume. If Qdisk is entered without a specific \texttt{volume\_serial}, statistics about available space are displayed for each volume that is defined with “Use Attribute=storage”.

\texttt{volume\_serial}
Displays statistics about available space on a specific volume.

**QUOtesoverride**
Specifies single quotation mark at the beginning and end of a file name should override the current working directory instead of being appended to the current working directory. This is the way single quotation marks are used in all
previous MVS FTP servers, and is the default. Any single quotation mark inside the beginning and ending quotation mark are treated as part of the file name.

QUOtesoverride indicates the usage of single quotation mark appearing at the beginning of, or surrounding, a file name. The setting of this keyword affects all FTP subcommands that have a path name as a parameter except keywords on the LOCSite subcommand.

READTAPEFormat
Used to provide information about an input data set on tape. If specified without the tape_format (which is the default), processing of input tapes will not take advantage of the record format information prior to open. The equal sign (=) is optional in this case.

READTAPEFormat has no effect on, and is not affected by DATAClass, DCbdsn, LRecl, RECfm, or any other parameters associated with creating a data set.

tape_format
Specifies the format of the records on the input tape data set. Valid formats are:

F Fixed record length
V Variable record length
S Spanned records
X Logical record length is undefined (Lrecl X)
blank Unspecified

The formats are mutually exclusive. Spanned implies variable, and Lrecl X implies spanned. If specified, the tape_format must be the most inclusive identifier in the list that matches the tape. If it is not the most inclusive identifier, an error message will be issued. For example, if tape_format is S (spanned), and the tape contains records with undefined length (Lrecl X), the request will fail. An unspecified format will avoid this type of error. However, the following should be considered:

- Specify a value for READTAPEFormat in all the following cases. Failure to specify a format will likely cause errors in processing the tape.
  - The record length is undefined (Lrecl X).
  - The records are spanned (Recfm is VBS, VS).
  - The records are variable (Recfm is V, VB, VBA) and RDW is specified.

- Specify a value for READTAPEFormat for all input tapes with one of the listed formats to ensure best results.

RDW
Specifies that variable record descriptor words (RDWs) are treated as if they were part of the record and not discarded during FTP transmission of variable format data sets. This applies to transfers in stream mode only.

Note: RDW information is stored in binary. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.
**RECfm**

Used to specify the record format of a data set. When specified without the `format`, no record format is used when allocating the data set. The equal sign (=) is optional in this case.

Specify RECfm with no value when you are also specifying DATAClass=\textit{data\_class} and you want the SMS dataclass to provide the RECfm format, or when you are specifying DCbsdns=\textit{data\_set\_name} and you want to use the record format from the DCBDSN data set.

If RECfm=\textit{format} is specified with either DATAClass or DCbsdns, the value specified by the LOCSite RECfm parameter overrides the DATAClass or DCbsdns record format.

**format**

Specifies the record format of a data set. Valid record formats are: F, FA, FB, FBA, FBM, FBS, FBSA, FBSM, FM, FS, FSA, FSM, U, UA, UM, V, VA, VB, VBA, VBM, VBS, VBSA, VBSM, VM, VS, VSA, and VSM. The characters used to specify these record formats have the following meanings:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Fixed record length</td>
</tr>
<tr>
<td>V</td>
<td>Variable record length</td>
</tr>
<tr>
<td>U</td>
<td>Undefined record length</td>
</tr>
<tr>
<td>B</td>
<td>Blocked records</td>
</tr>
<tr>
<td>S</td>
<td>Spanned records (if variable) / standard records (if fixed)</td>
</tr>
<tr>
<td>A</td>
<td>Records contain ISO/ANSI control characters</td>
</tr>
<tr>
<td>M</td>
<td>Records contain machine code control characters</td>
</tr>
</tbody>
</table>

**RESTGet**

Allows opening the checkpoint data set for the Get request. This is the default when the RESTGet statement has not been added to the FTP.DATA file.

**RETpdp**

Used to specify the number of days that a newly allocated data set should be retained.

Specify RETpdp with no value when you are also specifying DATAClass=\textit{data\_class} or MGmtclass=\textit{mgmt\_class} and you want SMS to provide the RETpdp value, or when you are specifying DCbsdns=\textit{data\_set\_name} and you want to use the RETpdp from the DCBDSN data set. If more than one of the LOCSite parameters (RETpdp, MGmtclass, DATAClass, or DCbsdns) are specified, the order of precedence (highest to lowest) is:

1. MGmtclass
2. RETpdp
3. DCbsdns
4. DATAClass

If a retention period is associated with an SMS management or data class, or with a model DCBDSN data set, the value of the retention period may be overridden to another nonzero value, but it cannot be overridden to have no retention period specified for the newly created data sets.

**days**

Specifies the number of days that a newly allocated data set should be
retained. The valid range is 0–9999. A value of 0 indicates a retention period of 0 days so that the data set expires the same day as it was created.

**SBDDataconn**
Specifies the conversions between file system and network code pages to be used for data transfers.

```plaintext
LBOSITE SBDDataconn=dsname
LBOSITE SBDDataconn=(file_system_cp,network_transfer_cp)
LBOSITE SBDDataconn=*
```

**dsname**
Specifies the fully qualified name of an MVS data set or HFS file that contains the EBCDIC-to-ASCII and ASCII-to-EBCDIC translate tables generated by the CONVXLAT utility.

**Notes:**
1. The name must not be enclosed in quotation marks. If quotation marks appear, they are treated as part of the name. (QUOtesoverride is ignored.)
2. The HFS name is case sensitive. The MVS name is not.
3. The name cannot begin with a left parenthesis `[(`.
4. The SBDDataconn keyword must be the only keyword or the last keyword on a LOCStie subcommand.

**file_system_cp**
Specifies the name of the file system code page recognized by iconv. For a list of code pages supported by iconv, refer to code set converters in the z/OS C/C++ Programming Guide.

**network_transfer_cp**
Specifies the network transfer code page recognized by iconv. For a list of code pages supported by iconv, refer to code set converters in the z/OS C/C++ Programming Guide.

**FTP_STANDARD_TABLE**
Indicates that the FTP internal tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), are to be used on the data connection.

**SBSUB**
Specifies that substitution is allowed for data bytes that cannot be translated in a single byte character translation. The substitution character is specified by the SBSUBCHAR parameter.

**SBSUBCHAR**
Specifies the value that is used for substitution when SBSUB is also specified. The value is one of the following:

**SPACE**
When the target codeset is ASCII, replace untranslatable characters with 'X'20' during SBCS data transfers. When the target codeset is EBCDIC, replace untranslatable characters with 'X'40' during SBCS data transfers.

**nn** Replace untranslatable characters with *nn* during SBCS data transfers, where *nn* is a hexadecimal value from 00 to FF.
SECondary
Specifies the amount of tracks, blocks, or cylinders for secondary allocation.

Specify SECondary=0 when you are also specifying DATAClass=dataclass and you want the SMS dataclass to provide the SECondary value. To enable the SMS data class to determine the space allocation, both PRImary and SECondary must be specified as 0. The tracks, blocks, cylinders setting is ignored in this case. If SECondary is specified as other than 0 with DATAClass, the value specified by the SIte SECondary parameter overrides the DATAClass space allocation.

amount
Specifies the amount of tracks, blocks, or cylinders for secondary allocation. The valid range is 0–16 777 215. If you specify an amount of 0, FTP allocates without specifying secondary space.

SPRead
Specifies that the output is in spreadsheet format when the file type is SQL.

SQLCol
Specifies the column headings of the SQL output file.

any
The label of the DB2 SQL table column heading is the first choice for column heading, but if there is no label, the name becomes the column heading.

labels
Labels are the DB2 SQL table column headings. If any of the columns do not have labels, FTP supplies a column heading in the form of COLnnn.

names
Uses the names of the DB2 SQL table column headings. The labels are ignored.

STOrclass
Specifies the SMS storage class as defined by your organization for the target host. Cancels the storage class specification when specified without a storage_class parameter value. The equal sign (=) is optional in this case.

See "Specifying values for new data sets" on page 86 for more information about specifying attributes when allocating new data sets.

storage_class
Specifies the SMS storage class as defined by your organization for the target host. The SMS storage class may override settings for the VOLume or Unit site parameters.

TRacks
Specifies that primary and secondary space allocations are in tracks.

If both PRImary and SECondary are specified as 0, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the TRacks parameter is ignored.

TRAILingblanks
Specifies that the FTP client preserves the trailing blanks in a fixed format data set when the data is sent to a foreign host.

TRUNcate
Specifies that truncation is permitted. The FTP client does not set an error when a truncated record is detected and the file transfer continues.
UCOUNT
Specifies how many devices to allocate concurrently to support the allocation request.

unit_count
Specifies number of devices to allocate. Valid value is 1–59. When specified without a value, the FTP server does not specify a unit count when allocating data sets.

P Parallel mount request.

The setting for UCOUNT may be overridden by the SMS storage class, if one is specified. Therefore, it is recommended that UCOUNT not be coded if an SMS storage class is in use.

UCSHOSTCS
Specifies the EBCDIC code set to be used when converting to and from Unicode. If you do not specify a code_set, the current code set is used.

code_set
Name of the EBCDIC code set to be used when converting to and from Unicode.

UCSSUB
In Unicode-to-EBCDIC conversion, the EBCDIC substitution character is used to replace any Unicode character that cannot successfully be converted. Data transfer continues.

UCSTRUNC
In Unicode-to-EBCDIC conversion, truncation of EBCDIC data is allowed. The data transfer continues even if EBCDIC data is truncated.

UMask
Defines the file mode creation mask. The file mode creation mask defines which permission bits are not to be set on when a file is created. When a file is created, the permission bits requested by the file creation are compared to the file mode creation mask, and any bits requested by the file creation which are not allowed by the file mode creation mask are turned off.

The format of the UMask keyword is UMask ooo.

When a file is created, the specified permission bits for the file are 666 (-rw-rw-rw-). If the file mode creation mask is 027, the requested permissions and the file mode creation mask are compared:

<table>
<thead>
<tr>
<th>Binary</th>
<th>Octal</th>
<th>Permission Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>110110110</td>
<td>666</td>
<td>-rw-rw-rw-</td>
</tr>
<tr>
<td>000010111</td>
<td>027</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>----------------</td>
</tr>
<tr>
<td>11010000</td>
<td>640</td>
<td>-rw-r-----</td>
</tr>
</tbody>
</table>

The actual permission bits set for the file when it is created is 640 (-rw-r-----).

Notes:
1. The default value for UMask is 027.
2. You cannot use FTP to create HFS files with execute permissions. If you require execute permissions, use the LOCSite CHMod command to change permissions after the file has been created.

Unit
Specifies the unit type for allocation of new data sets.
The setting for Unit may be overridden by the SMS storage class, if one is specified. Therefore, it is recommended that Unit not be coded if an SMS storage class is in use.

**unit_type**
The unit type (for example, 3380) for the allocation of new data sets on direct access devices. If `unit_type` is not specified, the unit type used for allocation is set back to the system default.

**VCOUNT**
Specifies the number of volumes an allocated data set may span. When specified without a `volume_count` value, the FTP server uses a volume count of 50 when allocating data sets.

**volume_count**
Valid values are from 1–255.

**VOLume**
Specifies the volume serial number for allocation of new data sets.

**volume_serial**
The serial number of the volume to use for allocation.

**volume_serial_list**
A list of one or more volume serial numbers for allocation. Delimit each volume serial number from the previous one with a comma.

If `volume_serial_list` or `volume_serial` is not specified, the volume serial number used for allocation is set back to the system default.

The setting for VOLume may be overridden by the SMS storage class, if one is specified. Therefore, VOLume should not be coded if an SMS storage class is in use.

**WRAPrecord**
Indicates that data is wrapped to the next record if no new line character is encountered before the logical record length of the receiving file is reached.

**WRTAPEFastio**
Indicates that ASCII Stream data that is being written to tape is allowed to be written using BSAM I/O.

**Usage**

- Because more than one parameter can be specified with the LOCSIt command, `parameter` can be repeated many times, with each `parameter` separated by a blank space.
- Issue the HEIp LOCSIt command to display a list of services available on the local host.
- The site-dependent information set with the LOCSIt command remains active until you issue a new LOCSIt command. The new LOCSIt command adds to or changes the parameters established by previous LOCSIt subcommands.
- If you specify one or more incorrect parameters with the LOCSIt command, an error message specifying the incorrect parameter is displayed. All correct parameters are set, regardless of any incorrect parameters, and do not need to be reissued.
Context

- See “HEIp and ? subcommands—Display help information” on page 172 for more information on the HEIp subcommand.
- To check the effect of the LOCSIte command on the attributes at the local host, see “LOCSTat subcommand—Display local status information” on page 201.
LOCSTat subcommand—Display local status information

Purpose

Use the LOCSTat subcommand to display local status information.

Format

```
LOCSTat
```

Parameters

There are no parameters for this subcommand.

Examples

The following example shows the output from a LOCSTat subcommand.

```
Command: locstat
Trace: FALSE, Send Port: TRUE
Send Site with Put command: TRUE
Connected to: 9.67.113.57, Port: 6321, logged in
Local Port: 1045
Proxy connection to: 9.67.112.25, Port: 6121, logged in
Proxy Local Port: 1046
Data type: a, Transfer mode: s, Structure: f
Automatic recall of migrated data sets.
Automatic mount of direct access volumes.
Partitioned data sets will be created with 23 blocks
File type is SEQ (Sequential - the default).
Number of access method buffers is 1.
Mgmtclass for the new data sets is MGMT1
Storclass for the new data sets is STOR1
Dataclass for the new data sets is DATA1
RDW’s from VB/VBS files are retained as part of data.
Retention period is 2
DB2 subsystem name is DB2
Volid of Migrated Data Sets is MIGRAT
Data sets will be allocated like data set 'USER33.TEST.S.A'
Trailing blanks in records read from RECFM F data sets are discarded.
Record format: FB, Lrecl: 80, Blocksize: 8000.
Data sets will be allocated on CPDLB0
Data sets will be allocated using unit SYSDA
Wrapping data into next record.
Checkpoint interval is 10000
Prompting: ON, Globbing: ON
ASA control characters transferred as C control character sequences
New data sets deleted if a store operation terminates abnormally
Single quotes will override the current working directory
UMASK value is 027
Using 'USER33.FTP.DATA' for local site configuration parameters
```
LPwd subcommand—Display the current working-level qualifier

Purpose

Use the LPwd subcommand to display the name of the current working directory on the local host.

Format

```
LPwd
```

Parameters

There are no parameters for this subcommand.

Examples

Display the name of the current working directory:

```
lpwd
Local directory is partitioned data set USER14.FTP.TEST1.
Command:
```
LS subcommand—Obtain a list of file names

Purpose

Use the LS subcommand to list only the names of a set of remote files, file group, or directory.

Format

```
[Disk] LS name
```

Parameters

- **name**: Specifies the set of remote files whose names are to be listed. The default is the entire current directory or file group.
- **Disk**: Stores the results of the LS subcommand in the `user_id.FTP.LSOUTPUT` data set. The results are not displayed on the screen.

**Note**: If the local current working directory is an HFS directory, the results are stored in a file named `LSOUTPUT`.

Examples

The following is a sample response that is displayed as a result of the LS subcommand.

```
>>>PORT 9,67,58,227,4,63
  200 Port request OK.
>>>NLST
  125 List started OK.
  A.X
  CHR.TXT
  OBEY.TCPIP
  PROFILE.EXEC
  SPF.ISPPROF
  USERTRAN.TCXPXB
  250 List completed successfully.
Command:
```

The following is a sample entry and response that is displayed as a result of the LS subcommand listing HFS files.
cd '/u/user121/ftp.example'

>>>CWD '/u/user121/ftp.example'
250 HFS directory /u/user121/ftp.example is the current working directory
Command:
ls

>>>PORT 9,67,112,25,4,62
200 Port request OK.

>>>NLST
125 List started OK
append02
file1
file2
file3
file4
file5
250 List completed successfully.
Command:

Usage

- To make a file group the current working directory, use the CD subcommand. The method you use to specify a directory or file group is host-dependent.
- You can use special characters for pattern matching when specifying the name. These characters depend on the host FTP server. See the Dir subcommand—Obtain a list of directory entries on page 155 for information on using special characters with the z/OS FTP server.
- If the current local directory is a PDS, only a member named LSOUTPUT is created. If the current local directory is not a PDS, the local directory, not the user ID, is used as the high-level qualifier for the data set name.

Context

- See Appendix A, “Specifying data sets and files” on page 383 for more information about pattern matching and about specifying data sets and files.
- To make a file group the current working directory, see the CD subcommand—Change the directory on the remote host on page 140.
- To get a list of complete directory entries with auxiliary information about the files, see the Dir subcommand—Obtain a list of directory entries on page 155.
- To change the local directory, see the Lcd subcommand—Change the local working directory on page 179.
MDelete subcommand—Delete multiple files

Purpose

Use the MDelete subcommand to delete multiple files.

Format

```
MDelete foreign_file
```

Parameters

`foreign_file`

Specifies the name of the file to be deleted on the remote host.

Because more than one file can be deleted with the MDelete subcommand, the `foreign_file` parameter of the MDelete subcommand can be repeated many times, with each `foreign_file` separated by a blank space.

Examples

The following is a sample entry and the response that is displayed as a result of the MDelete subcommand for multiple HFS files.

```
  cd '/u/user121/ftp.example'
  >>>CWD '/u/user121/ftp.example'
  250 HFS directory /u/user121/ftp.example is the current working directory
  Command: mdelete file1 file2 file3
  >>>PORT 9,67,112,25,4,75
  200 Port request OK.
  >>>NLST file1
  125 List started OK
  250 List completed successfully.
  >>>PORT 9,67,112,25,4,77
  200 Port request OK.
  >>>NLST file2
  125 List started OK
  250 List completed successfully.
  >>>PORT 9,67,112,25,4,76
  200 Port request OK.
  >>>NLST file3
  125 List started OK
  250 List completed successfully.
  >>>DELE file1
  250 /u/user121/ftp.example/file1 deleted.
  >>>DELE file2
  250 /u/user121/ftp.example/file2 deleted.
  >>>DELE file3
  250 /u/user121/ftp.example/file3 deleted.
```

Usage

- If you specify one or more incorrect foreign files with the MDelete subcommand, an error message specifying the incorrect foreign file is displayed. All correct foreign files are deleted, regardless of any incorrect foreign files, and the MDelete subcommand does not need to be reissued for these files.
HFS file names require special handling for certain special characters. All special characters that the operating system requires to be preceded by an escape character in commands issued to the shell must be preceded by the backslash (\) escape character, except for the single quote (‘), double quote (“), or blank ( ).

Context

See Appendix A, “Specifying data sets and files” on page 383 for more information about naming conventions.
MGet subcommand—Copy multiple files

Purpose
Use the MGet subcommand to copy multiple files from a remote host to your local host and create a corresponding number of local files.

Format

```
MGet foreign_file
           (REPLACE)
```

Parameters

`foreign_file`
Specifies the name of the file to be retrieved from the remote host.

Because more than one file can be copied with the MGet subcommand, the `foreign_file` parameter of the MGet subcommand can be repeated many times, with each `foreign_file` separated by a blank space. You can use special characters for pattern matching when specifying the `foreign_file` with the MGet subcommand. These characters are dependent on the foreign host FTP server.

`REPLACE`
Causes a data set on your local host to be overwritten if it already exists. If the data set already exists, and you do not use the replace parameter, the existing data set is not overwritten. A message informing you of this is displayed.

If the data set already exists and you specify REPLACE, the data in the file is overwritten, but not reallocated. This means the local data set retains its existing characteristics.

Examples

The following is a sample entry and response that is displayed as a result of the MGet subcommand for multiple HFS files.
cd 'u/user121/ftp.example'

>>>CD '/u/user121/ftp.example'
250 HFS directory /u/user121/ftp.example is the current working directory
Command:
mget file1 file2 file3

>>>PORT 9,67,112,25,4,90
200 Port request OK.

>>>NLST file1
125 List started OK
250 List completed successfully.

>>>PORT 9,67,112,25,4,91
200 Port request OK.

>>>NLST file2
125 List started OK
250 List completed successfully.

>>>PORT 9,67,112,25,4,92
200 Port request OK.

>>>NLST file3
125 List started OK
250 List completed successfully.

>>>PORT 9,67,112,25,4,93
200 Port request OK.

>>>RETR file1
125 Sending data set /u/user121/ftp.example/file1
250 Transfer completed successfully.
3464 Bytes transferred in 1.031 seconds. Transfer rate 3.36 kbytes/sec.

>>>PORT 9,67,112,25,4,94
200 Port request OK.

>>>RETR file2
125 Sending data set /u/user121/ftp.example/file2
250 Transfer completed successfully.
3993 Bytes transferred in 0.923 seconds. Transfer rate 4.33 kbytes/sec.

>>>PORT 9,67,112,25,4,95
200 Port request OK.

>>>RETR file3
125 Sending data set /u/user121/ftp.example/file3
250 Transfer completed successfully.
3993 Bytes transferred in 0.791 seconds. Transfer rate 5.05 kbytes/sec.
Command:

The following is a sample entry and response that is displayed as a result of the MGet subcommand using a wildcard character in the file name.
Usage

- When you use the MGet subcommand, FTP may truncate data records and you might lose data if:
  - You are creating a new data set at the client and the value of LRecl, as shown by the LOCStat command, is a value less than the LRecl of a received data set, FTP truncates the received data set.
  - The data set name already exists at the client and the logical record length (LRecl) of the data set at the client is less than the LRecl of the transmitted data set, FTP truncates the transmitted data set.

  You can encounter this situation when you use MGet with the REPLACE option.

- If the name specified for foreign_file is not acceptable to your local host, the file is not transferred. To get a file from the remote host, you must have a defined working directory on that host, and you must have read privileges to the files in this working directory.

- If you specify one or more incorrect foreign files with the MGet subcommand, an error message specifying the incorrect foreign file is displayed. All correct foreign files are retrieved, regardless of any incorrect foreign files, and do not need to be reissued.

- HFS file names require special handling for certain special characters. Except for single quote ('), double quote ("), or blank ( ), all special characters that the operating system requires to be preceded by an escape character in commands issued to the shell must be preceded by the backslash (\) escape character.

- The MGet command is not applicable to generation data groups (GDGs).

- MGet can be used with the PROXy subcommand to transfer files from a host on a primary connection to a host on a secondary connection. See "PROXY subcommand—Execute FTP subcommand on secondary control connections" on page 222 for more information.

- If the data set is migrated, it will be replaced regardless of the replace option.
Context

- See "CD subcommand—Change the directory on the remote host" on page 140 for more information about working directories.
- See Appendix A, “Specifying data sets and files” on page 383 for more information about naming conventions.
MKdir subcommand—Create a directory on the remote host

Purpose

Use the MKdir subcommand to create a directory on the remote host.

Format

```bash
MKdir directory
```

Parameters

directory  
Specifies the name of the directory to be created.

Examples

In this example, a directory is created on the remote host (1.1.2.3 in this example). Both EXAMPLE and FTP.EXAMPLE are created in the remote host, showing the difference between specifying and omitting quotation marks in the directory name.

```
User: ftp 9.67.113.24.621
System: IBM FTP CS V1R2
FTP: using TCPCS
FTP.DATA FILE NOT FOUND. USING HARDCODED DEFAULT VALUES.
Connecting to 1.1.2.3, port 21
220-EZAFTSRV IBM FTP CS V1R2 at EMU.ABC.OZ, 15:34:32 on 2000-08-03.
220 Connection will not timeout.
NAME (<host>:tsouserid):
User: user121
System: >>>USER user121
331 Send password please.
Password:
User: >>>PASS ********
230 USER121 is logged on. Working directory is '/u/user121".
Command:
User: dir
System: >>>PORT 9,67,112,25,4,96
200 Port request OK.
>>>LIST
total 2768
-rwxr-xr-t 2 USER121 SYS1 389120 Feb 5 16:03 ftpdka
-rwxr-xr-t 2 USER121 SYS1 962560 Feb 5 16:04 ftpsrvka
-rw-r----- 1 USER121 SYS1 11648 Jan 20 14:30 g.s
drwxr-x--- 3 USER121 SYS1 0 Oct 21 17:50 msg
-rw-r----- 1 USER121 SYS1 1458 Jan 10 19:25 s.k
drwxr-x--- 2 USER121 SYS1 0 Feb 6 15:59 tcpip
drwxr-x--- 2 USER121 SYS1 0 Feb 6 17:29 test
250 List completed successfully.
Command:
User: mkdir example
System: >>>MKD example
257 "/u/user121/example" created.
Command:
```
Usage

- The **MKdir** subcommand sends a request to the remote host FTP server to create a directory with name directory in the current working directory. When the request is sent to an MVS server, a PDS is created if either a fully qualified MVS data set name is entered (for example, 'USER33.TEST.PDS'), or if the current working directory is an MVS data set name. For example, if the current working directory is USER33.TEST, a command MKdir PDS creates a PDS named USER33.TEST.PDS.

- For an MVS server, the data set characteristics used to allocate the new PDS are determined by the settings of the **SITE** parameters of the server.

Context

- See "**SITE** subcommand—Send site-specific information to a host" on page 240 for information on setting the **SITE** parameters.

- See "**STATus** subcommand—Retrieve status information from a remote host" on page 267 for information on displaying the **SITE** parameters.
MOde subcommand—Set the data transfer mode

Purpose

Use the MOde subcommand to define how bits of data are to be transmitted.

Format

```
MOde B
MOde C
MOde S
```

Parameters

- **B**: Sets the block mode. In block mode, data is transmitted as a series of data blocks, preceded by one or more header bytes. Block mode preserves the logical record boundaries of the data set or file. When MOde is set to B, the data transfer type must be EBCDIC.

  Specifying MOde B is equivalent to specifying the BLock subcommand.

- **C**: Sets the compressed mode. In compressed mode, data is transmitted as a series of data blocks, preceded by one or more header bytes. Compressed mode preserves the logical record boundaries of the data set or file. In compressed mode, data is transmitted without repetitive characters and blanks. When MOde is set to C, the data transfer type must be EBCDIC.

  **Note**: Because additional processing time is required for both the sender and receiver to compress or decompress the data, evaluate the time factor before you compress a file.

  Specifying MOde C is equivalent to specifying the COMpress subcommand.

- **S**: Sets the stream mode. In stream mode, data is transmitted as a stream of bytes. Any data transfer type can be used with stream mode. Stream mode is efficient because data block information is not transferred.

  Specifying MOde S is equivalent to specifying the STREam subcommand.

Usage

- To use MOde C, the receiving host must support the compressed data mode.
- Data compression increases CPU processing costs even if the amount of data transferred is not large.

Context

- For the syntax of the BLock, COMpress, and STREam subcommands, see "BLock subcommand—Set the block data transfer mode" on page 138, "COMpress subcommand—Set the compressed data transfer mode" on page 147, or "STREam subcommand—Set the stream data transfer mode" on page 269.
- For more information about transfer methods, see Table 11 on page 51.
MPut subcommand—Copy multiple data sets to the remote host

Purpose

Use the MPut subcommand to copy multiple data sets from your local host to the remote host.

Format

```
MPut local_data_set
```

Parameters

`local_data_set`

Specifies the name of the file on your local host being sent to the remote host.

Because more than one data set can be copied with the MPut subcommand, the `local_data_set` parameter of this subcommand can be repeated many times, with each `local_data_set` separated by a blank space. You can use the asterisk (*) character for pattern matching when specifying the `local_data_set` with the MPut subcommand.

Examples

The following is a sample entry and response that is displayed as a result of the MPut subcommand to send selected files.

```
Command:
mput file*
Mput FILE1 (Yes|No|Quit|Stop prompting)? yes
>>>PORT 9,67,113,57,5,128
200 Port request OK.
>>>STOR FILE1
125 Storing data set /u/user31/temp/FILE1
250 Transfer completed successfully.
164 bytes transferred in 0.010 seconds. Transfer rate 16.40 Kbytes/sec.
Mput FILE2 (Yes|No|Quit|Stop prompting)? no
Mput FILE3 (Yes|No|Quit|Stop prompting)? yes
>>>PORT 9,67,113,57,5,129
200 Port request OK.
>>>STOR FILE3
125 Storing data set /u/user31/temp/FILE3
250 Transfer completed successfully.
164 bytes transferred in 0.010 seconds. Transfer rate 16.40 Kbytes/sec.
Command:
dir
>>>PORT 9,67,113,57,5,130
200 Port request OK.
>>>LIST
125 List started OK
total 16
  -rw-r----- 1 USER31 SYS1 162 Aug 14 13:20 FILE1
  -rw-r----- 1 USER31 SYS1 162 Aug 14 13:21 FILE3
250 List completed successfully.
Command:
```
Usage

- FTP maintains the attributes of a data set that is transmitted between a client and a server. However, when you use the MPut subcommand, FTP may truncate data records and you might lose data, if:
  - You are creating a new file at the server and the value of LRecl, as shown by the STAtus subcommand, is a value less than the LRecl of the transmitted data set and SENDSite has been set to OFF, FTP truncates the transmitted data set.
  - The data set name already exists at the receiving site and the logical record length (LRecl) of the data set at the receiving site is less than the LRecl of the transmitted data set, FTP truncates the transmitted data set.
- By default, if you use the MPut subcommand, the remote host creates files with the same names as those specified in local_data_set and overwrites any existing files with those names.
  To put files on the remote host with unique file names, you must have set unique storage on before issuing the MPut command. Use the SUnique subcommand to change the storage method.
- To send a data set to the remote host, you must have a defined working directory on the remote host and write privileges to the files in this working directory.
- If you specify one or more incorrect parameters with the MPut subcommand, an error message specifying the incorrect parameter is displayed. All correct files are transferred, regardless of any incorrect parameters, and do not need to be reissued.
- The MPut command is not applicable to GDGs.

Context

- See “SUnique subcommand—Changes the storage method” on page 271 for information about changing the storage method on the remote host.
- See Appendix A, “Specifying data sets and files” on page 383 for more information about naming conventions.
- MPut can be used with the PROXy subcommand to transfer files from a host on a secondary connection to a host on a primary connection. See “PROxy subcommand—Execute FTP subcommand on secondary control connections” on page 222 for more information.
NOop subcommand—Test the connection

Purpose

Use the NOop subcommand to determine whether the foreign host is still responding.

Format

```
>>> NOop
```

Parameters

There are no parameters for this subcommand.

Examples

- If the foreign host is responding, you receive one of the following responses:
  
  200 OK or 200 NOOP command successful.

- If the foreign host does not respond or is not connected, you receive appropriate error messages, such as:

  EZA1534I Control connection with ipaddr dies.
  EZA1457I You must first issue the 'OPEN' command.

Usage

You can use the NOop command to keep a connection alive that would otherwise be disconnected if it were idle for longer than the system timeout period.
Open subcommand—Connect to the FTP server

Purpose

Use the Open subcommand to open a connection to the remote host FTP server in the following situations:

- If, after closing a connection, you want to open another connection without leaving the FTP environment.
- If you were unable to open a connection when you specified a foreign_host with the FTP command.

Format

```
Open host_name port_number
```

Parameters

- **host_name**
  Specifies the host name or IP address of the foreign host.

- **port_number**
  Identifies a port on the foreign host. The default is well-known port 21.

Usage

If you are already connected to a host, you must disconnect from the host before you can connect to a different host with the Open subcommand. The only exception to this is if you are using the PROxy Open command. See “PROxy subcommand—Execute FTP subcommand on secondary control connections” on page 222 for more information.

Context

See “CLOase subcommand—Disconnect from a remote host” on page 146 for more information about closing a connection.
PAss subcommand—Supply a password

Purpose

Use the PAss subcommand to supply a password to a host.

Format

```
> [password] [newpass/newpass] [:userdata]  

> [account_information] 
```

Parameters

- **password**
  Specifies your password on the remote host.

- **/newpass/newpass**
  Resets a password.

- **:userdata**
  The optional user data must be separated from the password information by a colon (:) and can be any combination of up to 200 nonblank characters and numbers except the colon. Care should be taken when using the backslash character (\) in combination with other characters which might be interpreted as an escape sequence by the C compiler.

- **account_information**
  An optional parameter that will be supplied to the remote FTP server if the server requests account information after receiving the password.

Usage

- The PAss subcommand must be preceded by the User subcommand. For some sites, the password completes your identification for access control on the remote host.

- On z/OS hosts, a PAss command is automatically generated by the User command. If you do not enter the password as part of the user command, you are prompted for it.

- The optional user data is used by the z/OS FTP server as a character string that is passed to the server’s FTCHKPWD user exit routine.

Context

See [User subcommand—Identify yourself to a host or change your TSO user ID password on page 281](#) for more information.
PRIvate subcommand—Set the protection level for data transfers to PRIVATE

Purpose

Use the PRIvate subcommand to set the protection level for data transfers on the data connections to private. This subcommand is equivalent to specifying the PROTect PRIvate subcommand.

Format

```
PRIvate
```

Parameters

There are no parameters for this subcommand.

Examples

To set the protection level to private, enter:

```
private
```

Usage

- This subcommand is not valid when there is no active security mechanism.
- Data transmissions are confidentiality and integrity protected by encryption.
PROMpt subcommand—Toggle interactive prompting for M* commands

Purpose

Use the PROMpt subcommand to toggle interactive prompting for MDelete, MGet, and MPut commands. Prompting is the default action unless the FTP session was started with the -i option, which turns off interactive prompting.

Format

```
\[-\[Z(qOOOO>[ZU3OOOO
   PROMp
```

Parameters

There are no parameters for this subcommand.

Examples

The following example shows using the MPut command with interactive prompting on.

```
Command:
mput file*
  Mput FILE1 (Yes|No|Quit|Stop prompting)? yes
  >>>PORT 9,67,113,57,5,128
  200 Port request OK.
  >>>STOR FILE1
  125 Storing data set /u/user31/temp/FILE1
  250 Transfer completed successfully.
  164 bytes transferred in 0.010 seconds. Transfer rate 16.40 Kbytes/sec.
  Mput FILE2 (Yes|No|Quit|Stop prompting)? no
  Mput FILE3 (Yes|No|Quit|Stop prompting)? yes
  >>>PORT 9,67,113,57,5,129
  200 Port request OK.
  >>>STOR FILE3
  125 Storing data set /u/user31/temp/FILE3
  250 Transfer completed successfully.
  164 bytes transferred in 0.010 seconds. Transfer rate 16.40 Kbytes/sec.
Command:
```

Context

See "Using the FTP command" on page 29 for more information on the -i option.
PROTect subcommand—Set the protection level for data transfers

Purpose
Use the PROTect subcommand to set the protection level for data transfers on the data connections.

Format

```
>    PROTect    CLEAR
  PRIVATE
  SAFE

```

Parameters

**CLEAR**
Data transmissions are not protected. Specifying PROTect CLEAR is equivalent to specifying the CLEar subcommand.

**PRIVATE**
Data transmissions are confidentially and integrity protected. Specifying PROTect PRIVATE is equivalent to specifying the PRIvate subcommand.

**SAFE**
Data transmissions integrity are protected by cryptographic checksum.

Examples

To set the protection level to private, enter:

```
prot private
```

Usage

This subcommand is not valid when there is no active security mechanism.
PROXy subcommand—Execute FTP subcommand on secondary control connections

Purpose

Use the PROXy subcommand to execute an FTP subcommand on secondary control connections. PROXy enables the FTP command to connect simultaneously to two remote FTP servers, and transfer files between those servers.

Format

```
PROXy subcommand
```

Parameters

subcommand

The name of any FTP subcommand except those listed in the first note in "Usage" on page 223. The first PROXy subcommand should be Open, which establishes the secondary server connection.

The following subcommands behave differently when prefaced by the PROXy subcommand:

- Open establishes the secondary server connection.
- Close closes the secondary server connection.
- Get and MGet transfer files from the host on the primary connection to the host on the secondary connection.
- PUT, MPut, and APpend transfer files from the host on the secondary connection to the host on the primary connection.

Examples

The following example shows a proxy open to establish connection to a secondary server.

```
Command:
proxy open 9.67.113.57 6321
Connecting to: 9.67.113.57 port: 6321.
220 Connection will not timeout.
NAME (9.67.113.57:USER33): user34
>>USER user34
331 Send password please.
PASSWORD:
>>PASS
230 USER34 is logged on. Working directory is "USER34."
```

The following example shows the commands for a proxy get between IPv4 nodes:

- PASV to the secondary server
- PORT to the primary server
- RETR to the primary server
- STOR to the secondary server

Two 250 replies are received by the client, one from each server.
The following shows a proxy open to establish connection to a secondary server with an IPv6 address:

Command:
proxy open local167v6
220-Welcome to my test system.
220-You are logged on from fec9:c2d4::9:67:115:13
220 Connection will not timeout.

Command:
user user2
>>> USER user2
331 Send password please.
PASSWORD:

>>> PASS
230 USER2 is logged on. Working directory is "/".

This example shows the commands for a proxy PUT between IPv6 nodes:

- EPSV to the secondary server
- EPRT to the primary server
- STOR to the primary server
- RETR to the secondary server

As in the first example, two 250 replies are received by the client.

Command:
proxy put bob testfile
Load module transfer does not support load module rename

>>> EPSV
229 Entering Extended Passive Mode (|||1027||)
>>> EPRT |2|fec9:c2d4::9:67:115:12|1027|
200 EPRT request OK
>>> STOR testfile
125 Storing data set /tmp/myTest/testfile
>>> RETR bob
125 Sending data set /tmp/myTest/bob
250 Transfer completed successfully.
250 Transfer completed successfully.

Command:

Usage

- The following subcommands are not valid proxy subcommands:
  - DEBug
  - DUMP
  - DELImit
  - GLob
- LANGUAGE
- LCd
- LMkdir
- LOC$Site
- LOCStat
- LPwd
- PROMpt
- QUIT
- REStart
- SENDPort
- SENDSite
- SRestart
- SUnique
- TSO
- Verbose

- To receive help from a server on a secondary control connection, enter PROXY HELP SERVER.
- Data transfer in PROXY mode can be restricted if the server is set up to reject PORT and EPRT commands with certain parameters. Refer to the z/OS Communications Server: IP Configuration Guide for more details.
- If an open subcommand is entered as a proxy subcommand by the client and the session is currently protected by a security mechanism (for example, TLS), then the subcommand is rejected with the following message:
  Proxy open is not supported with security mechanisms
- If the connection to one server is IPv4 and the connection to the other is IPv6, proxy transfers may not be possible. The two servers must have the ability to connect to each other as well as to the client. A z/OS FTP server cannot be the primary server in a proxy transfer unless the connection from the client to each of the servers is of the same protocol.
PUt subcommand—Copy data sets to the remote host

Purpose

Use the PUt subcommand to copy data sets from your local host to the remote host.

Format

\[ \text{PUt} \text{ local\_file} \rightarrow \text{foreign\_file} \]

Parameters

\textit{local\_file}

Specifies the name of the file on your local host being sent to the remote host.

\textit{foreign\_file}

Specifies the name that the delivered data set is given on the remote host. If the \textit{foreign\_file} name is not specified, the \textit{foreign\_file} name is the same as the \textit{local\_file} name.

Usage

- FTP maintains the attributes of a data set that is transmitted between a client and a server. However, when you use the PUt subcommand, FTP may truncate data records and you might lose data, if one of the following occurs:
  - If you are creating a new file at the server and the value of LRecl, as shown by the STATus subcommand, is a value less than the LRecl of the transmitted data set and SENDSite subcommand has been set to OFF, then FTP truncates the transmitted data set.
  - If the data set name already exists at the receiving site and the logical record length (LRecl) of the data set at the receiving site is less than the LRecl of the transmitted data set, then FTP truncates the transmitted data set.

- If FTP does not support directory content transfers in partitioned data sets, it is not possible to FTP load modules.

- When a PUt is issued, FTP automatically sends a Site subcommand containing record format information for the file or data set. To toggle this off, you must first issue a SENDSite subcommand. See "SENDSite subcommand—Toggle the sending of site information" on page 239 for more detailed information.

- If the remote host already has a file with the name specified by \textit{foreign\_file}, the remote host overwrites the existing file. If the remote host does not have a file with the same name specified by \textit{foreign\_file}, the remote host creates a new file.

- To put files on the remote host with unique file names, you must have set unique storage on before issuing the PUt command. Use the SUnique subcommand to change the storage method.

- To send a data set to the remote host, you must have a defined working directory on the remote host and write privileges to the files in this working directory. UPDATE authority is the minimum required for write access. To ensure that the newly created data set can be written to before the data set has been allocated and opened, FTP will validate that the user ID has at least UPDATE authority. If this fails, then FTP will be able to issue a reply to the client which is indicative of the failure. If open is allowed to continue and it fails due to lack of authority, then the reply will not be as definitive.
Context

- See "SUnique subcommand—Changes the storage method" on page 271 for information about changing the storage method on the remote host.
- See Appendix A, “Specifying data sets and files” on page 383 for more information about naming conventions.
- PUT can be used with the PROXY subcommand to transfer files from a host on a secondary connection to a host on a primary connection. See "PROXY subcommand—Execute FTP subcommand on secondary control connections" on page 222 for more information.
**PWd subcommand—Display the current working directory**

**Purpose**

Use the PWd subcommand to display the name of the current working directory on the remote host.

**Format**

```
>>> PWd
```

**Parameters**

There are no parameters for this subcommand.

**Examples**

Display the name of the current working directory:

```
pwd
>>> PWd
  257 "'USER17.HSMTEST.'" is working directory
Command:
```

Display the name of the current HFS working directory:

```
pwd
>>> PWd
  257 "'/u/user121/example'" is the HFS working directory.
Command:
```
QUIt subcommand—Leave the FTP environment

**Purpose**

Use the QUIt subcommand to disconnect from the foreign host and end the FTP session.

**Format**

```
> [Z(qOOOO)
> QUIt
```

**Parameters**

There are no parameters for this subcommand.

**Usage**

- The QUIt subcommand ends the FTP session with the remote host and exits FTP on the local host. To establish a new session, use the FTP command.
- In a z/OS UNIX environment, you can also press Ctrl-C to end an FTP session.
- When running with both a primary and a secondary server (by using the PROXy subcommand), the QUIt subcommand disconnects both sessions.

**Context**

See ["Using the FTP command" on page 29](#) for information about the FTP command.
QUOte subcommand—Send an uninterpreted string of data

Purpose

Use the QUOte subcommand to send an uninterpreted string of data to the server port on the foreign host.

The QUOte subcommand bypasses the FTP interface of your local host. You can use the QUOte subcommand to send commands that the remote server understands, but that the local host does not understand.

Format

\[ \text{QUOte}\ string \]

Parameters

\text{string} specifies the data to be sent verbatim to the remote host FTP server.

Examples

- For example, QUOTE TYPE B 1 causes the FTP server to change its transfer type to Shift JIS kanji, without changing the transfer type in the FTP client. The client in this example should be set to the ASCII transfer type before the QUOte subcommand is issued.
- The following example shows the screen display when setting the DBCS transfer type to JIS78KJ, shift-in JISROMAN, and then setting it to HANGEUL using EBCDIC SO/SI characters. The example shows an MVS TCP/IP FTP client connected to an MVS TCP/IP FTP server. All three methods of setting the DBCS transfer type are shown.
Usage

- No parsing or validity checking is performed on the character string you enter by FTP on your local host. If the character string you send to the FTP server is part of a required sequence of commands, you are required to provide this sequence correctly, or the results are unpredictable.

- The QUOte subcommand can be used to generate any of the DBCS TYpe commands supported by the server. This subcommand is used when the FTP server supports the DBCS TYpe command, but the FTP client does not.
RECord subcommand—Set the file structure to record

Purpose

Use the RECord subcommand to set the file structure to record. This is equivalent to specifying the STRucture R subcommand. See “STRucture subcommand—Set the file structure” on page 270 for more information.

Format

```
RECord
```

Parameters

There are no parameters for this subcommand.
REName subcommand—Rename files

Purpose

Use the REName subcommand to rename a file on the remote host.

Format

```
REName original_name new_name
```

Parameters

- `original_name`
  - Specifies the current name of the file.
- `new_name`
  - Specifies the new name of the file.

Usage

- For MVS data sets, if the file specified by the `new_name` already exists, the existing file is not replaced by the new file.
- For HFS files, if the file specified by `new_name` already exists, the existing file is replaced.
REStart subcommand—Restart a checkpointed data transfer

Purpose

Use the REStart subcommand to restart a checkpointed data transfer.

Format

```
REStart
```

Parameters

There are no parameters for this subcommand.

Usage

- The REStart subcommand restarts the last checkpoint file transfer request at the point of the last valid checkpoint stored in the checkpoint data set. See FTPDATA statement CHKPTPREFIX to determine the check point data set name. The file transfer environment (such as file transfer mode, type, and the hlq of the checkpoint data set name) which was in effect at the time of the last checkpoint file transfer request must be reestablished before issuing the REStart subcommand.
- The REStart subcommand can be used when a checkpointed file transfer request fails because of some temporary condition such as the loss of the connection between the client and the server.
- The checkpoint file name is formed using the CHKPTPrefix value and the current working directory. Refer to the description of CHKPTPrefix under the "LOCSIte subcommand—Specify site information to the local host" on page 183 or "Changing local site defaults using FTP.DATA" on page 63.

Context

For more information about checkpointing a file transfer request, see the description of the CHKptint parameter on page 244 or page 186.
RMdir subcommand—Remove a directory on the remote host

Purpose

Use the RMdir subcommand to remove a directory on the remote host.

Format

```
RMdir directory
```

Parameters

- **directory**
  
  Specifies the name of the directory to be removed.

Usage

- The RMdir subcommand sends a request to the remote host FTP server to remove a directory with name `directory` from the current remote directory.
- The RMdir subcommand can be used to delete a PDS.
SAfe subcommand—Set the protection level to safe

Purpose
Set the protection level on data transfers to “safe”. Data transmissions are integrity-protected by cryptographic checksum.

Format

Parameters
There are no parameters for this subcommand.
SChinese subcommand—Change the data transfer type to SCHINESE

Purpose

Use the SChinese subcommand to change the data transfer type to Simplified Chinese.

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the SItel/LOCStel SBDatalon or SItel XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

```
SChinese
```

Parameters

Sosi
Transferred data contains the shift-out and shift-in characters specified by the one of the following parameters — Ascii, Ebcdic, or Space. If no parameter is specified, ASCII is used as the default.

If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

Ascii
When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

Ebcdic
When combined with the Sosi parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

Space
When combined with the Sosi parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSo
Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

NOType
Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.
Usage

The SChinese client subcommand is equivalent to the TYPE B 9 server command.

Context

See “FTP with DBCS support” on page 123 for more information.
SENDPort subcommand—Toggle the sending of port information

Purpose
Use the SENDPort subcommand to toggle the automatic sending of the PORT command.

Format
```
SENDPort
```  

Parameters
There are no parameters for this subcommand.

Usage
- By default, the SENDPort subcommand is turned on when you start an FTP session. Each time you use the SENDPort subcommand, it is turned alternately on and off.
- FTP does not send PORT commands for data transfer when you disable PORT commands by toggling the function off.
- SENDPort is useful for communication with those FTP implementations that ignore PORT commands, but show (incorrectly) that the PORT command has been accepted.
- To determine if the sending of port information is enabled or disabled on your local host, use the LOCSTat subcommand.
- The sendport setting is ignored during proxy transfer.

Context
See “LOCSTat subcommand—Display local status information” on page 201 for more information about LOCSTat subcommand.
SENDSite subcommand—Toggle the sending of site information

Purpose

Use the SENDSite subcommand to toggle the automatic sending of the SITE subcommands when sending a data set to a foreign host.

Format

```
SENDSite
```

Parameters

There are no parameters for this subcommand.

Usage

- By default, the SENDSite subcommand is turned on when you start an FTP session. Each time you use the SENDSite subcommand, it is turned alternately on and off.
  
  When turned on, FTP sends a SITE subcommand containing record format information for the file or data set when you issue the PUT or MPut subcommand.

- SENDSite is useful when you want to PUT a file to the remote host and have the file created with the same characteristics as defined at the local host.

- If you are using either an SMS data class or a model DCB at your MVS server to provide the logical record length or record format, you must toggle the SENDSite setting off at the client. Otherwise, the SITE information that is sent automatically by the client overrides the values provided by the SMS dataclass or model DCB.

- To determine if the sending of site information is enabled or disabled on your local host, use the LOCSTat subcommand.

Context

See "LOCSTat subcommand—Display local status information" on page 201 for information about the LOCSTat subcommand.
Site subcommand—Send site-specific information to a host

Purpose

Use the Site subcommand to send information that is used by the remote host to provide services specific to that host system.

Format

```
Site options
```

options:

- ASATrans
- AUTOMount
- AUTORecall
- BLKsize = size
- Blocks
- BLOCKSSize = size
- BUfno = number
- CHKptint = number
- CHMod ooo filename
- symbolic filename
- CONDdisp = Catalog
- Delete
- CTRLConn 7BIT
- conv_ascii
- FTP_STANDARD_TABLE
- *
- CYlinders
- DATAClass = data_class
- DATASetmode

**Chapter 5. FTP subcommands**

- **DB2** = `db2_name`
- **DCbdsn** = `data_set_name`
- **DEBug** = `?`, `ACC`, `ALL`, `BAS`, `CMD`, `FLO`, `FSC(n)`, `INT`, `JES`, `NONE`, `PAR`, `SEC`, `SOC(n)`, `SQL`, `UTL`, `Xyy`
- **DESt** = `destination`
- **Directory** = `size`
- **DIRECTORYMode**
- **DUmp** = `?`, `n`, `ALL`, `FSC`, `JES`, `NONE`, `SOC`, `SQL`, `Xyy`
- **ENCODING** = `SBCS`, `MBCS`
Parameters

ASAtrans
Permits the FTP server to interpret the characters in the first column of ASA files being transferred as print control characters.

AUTOMount
Permits automatic mounting of volumes for data sets on volumes that are not mounted. If AUTOMount is specified and an unmounted volume is needed, a message is automatically issued to the MVS operator console requesting that
the volume be mounted. The MVS operator must then mount the volume and reply to the message before FTP can proceed.

**AUTORecall**  
Permits automatic recall of migrated data sets.

**BLKsize**  
Specifies the block size of a newly allocated data set. BLKsize is functionally equivalent to BLOCKSize. The BLOCKSize parameter is obsolete, but it is accepted to provide compatibility with previous releases of z/OS TCP/IP.  
When specified without a size, no block size is used when allocating the new data set. When specified without a size, the equal sign (=) is optional.  
Specify BLKsize with no value if you are also specifying DATAClass=data_class and you want the SMS data class to provide the BLKsize value, or if you are specifying DCbdsn=data_set_name and you want to use the block size from the DCBDSN data set. If BLKsize=size is specified with either the DATAClass or DCbdsn parameters, the value specified by the SITE BLKsize parameter overrides the DATAClass or DCbdsn block size.

**Notes:**  
1. If you specify BLKsize without a size or with a size of 0, FTP does not specify the block size when allocating new data sets.  
2. Be especially careful specifying both BLKsize= and BLOCKs. While there are conditions where this is tolerated, if a valid BLKsize cannot be determined, the data set will not be created when the allocation is attempted.

**size**  
Specifies the block size of a newly allocated data set. The valid range is 0–32 760.  
BLKsize=0 is a special case. When BLKsize=0 is specified, the operating system attempts to determine a block size for the new data set. FTP does not create the new data set unless the system is able to establish a nonzero block size.

**BLOCKS**  
Specifies that primary and secondary space allocations are in blocks.  
If both PRimary and SECondary are unspecified, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the BLOCKS parameter is ignored.

**BLOCKSize**  
Specifies the block size of a newly allocated data set. BLOCKSize is functionally equivalent to BLKsize. BLOCKSize is obsolete but it is accepted to provide compatibility with previous releases of z/OS TCP/IP. See the BLKsize parameter for more information.

**BUfno**  
Specifies the number of access method buffers that is used when data is read from or written to a data set. The default value is 5.  
**number**  
The number of buffers used. The valid range is 1–35.

**CHKpint**  
Specifies the checkpoint interval for the sending site in a file transfer request. If the checkpoint interval is 0, no checkpointing occurs and no marker blocks are transmitted. The default value is 0.
Use Site to set CHKptint when the FTP server is the sending site (Get command). It will not be useful to change the checkpoint interval to more than 0 in a Get unless the receiving site is able to process checkpoint markers in the file transfer data stream.

CHKptint is valid only for block and compressed modes of data transfer.

number

The checkpoint interval for the sending site in a file transfer request. This value is used to determine when checkpoint marker blocks are to be transmitted so that transmission can be restarted based on the information in the last marker.

A large checkpoint interval means that a large amount of data is sent in between markers and therefore few markers are sent. A smaller checkpoint interval means that less data is sent between markers and therefore more markers are sent.

The cost involved with using a nonzero checkpoint interval is the markers themselves are transmitted, which means more bytes are being sent across the network (approximately 44 bytes per marker).

To estimate the appropriate checkpoint interval, use the following formula. You need to know the record length of the file you are transferring and how much data you think can be transmitted reliably.

\[
\text{CHKPTINT} = \frac{\text{amount of data in interval}}{\text{record length of the file}}
\]

Do not execute a CHKptint more often than once every 200KB. The following are examples of a file you are transferring with 80-byte records.

\[
\text{CHKPTINT} = \frac{200\text{KB}}{80\text{ bytes}}
\]

\[
= \frac{200 \times 1024\text{ bytes}}{80\text{ bytes}}
\]

\[
= 2560
\]

CHMod

Changes the permission bits for a file.

\(\text{ooo filename}\)

\(\text{ooo}\) is the octal mask and \text{filename} is the name of the file whose mask is to be changed.

\(\text{symbolic filename}\)

\text{symbolic} is a value in the format accepted by the z/OS UNIX CHMOD command (for example, a+r).

\textbf{Note:} Only \textit{r}, \textit{w}, and \textit{x} permission bits are accepted in \textit{symbolic}.

If the file name does not begin with a slash (/), it is appended to the current working directory. If the file name does begin with a slash (/), it is interpreted as a complete directory name.

The file name specified must be an HFS file name. The setting of QUOtesoverride is ignored and all quotation marks are treated as part of the file name.

The CHMod keyword must be the only or last keyword on a Site subcommand.

CONDdisp

Specifies the disposition of the data set if a store operation for a new data set ends before all of the data is written.
Catlg
ертвает that a data set is kept and cataloged when an FTP file transfer ends prematurely.

Delete
Specifies that a data set is deleted when an FTP file transfer ends prematurely.
Delete is ignored if the file transfer failed as a result of the FTP server being terminated or if the server has received checkpoint information during data transfer.

CTRLConn
Specifies the ASCII code page to be used for control connections. The valid subcommands are:
SITE CTRLConn=7BIT
SITE CTRLConn=iconv_ascii
SITE CTRLConn=* 7BIT
Indicates 7-bit ASCII is to be used.
iconv_ascii
A name recognized by iconv to indicate an ASCII code page. For a list of code pages supported by iconv, refer to code set converters in the C/C++ Programming Guide.

FTP_STANDARD_TABLE
Indicates that the FTP internal tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), are to be used on the control connection.
*  Indicates that the ASCII used at initialization is to be used.

Note: Setting the control connection code page with SITE CTRLCONN disables UTF-8 encoding. You must start a new session to restore UTF-8 encoding.

CYlinders
Specifies that primary and secondary space allocations are in cylinders.
If both PRIMARY and SECONDARY are unspecified, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the CYlinders parameter is ignored.

DATAClass
Used to specify the SMS data class, as defined by your organization, for the target host. Specifying DATAClass with no parameter value cancels the data class specification. The equal sign (=) is optional in this case.
See “Specifying values for new data sets” on page 86 for more information about specifying attributes when allocating new data sets.

data_class
Specifies the SMS data class, as defined by your organization, for the target host. If values are specified for any of the following SITE parameters, the values specified by the SITE parameter overrides the value specified in the SMS data class:
• BLKsize
• Directory
• LRecl
If the DCbdsn Slte parameter is specified, the LRecl, RECfm, BLOCKSsize, and RETpd (if specified) of the DCBDSN data set overrides the values specified in the data class.

If the MGmtclass site parameter is specified, and the requested management class specifies a retention period, the retention period value of the management class may override the retention period value of the data class.

A Slte DATACLASS command pointing to an SMS data class must be used to create a PDSE. A load module loading from a temporary data set will always be a REPLACE operation, in that existing members will be overwritten in case of name conflicts. LMTR will not be performed in STOU mode (when SUnique is turned on).

**DATASETmode**

Specifies that all the data set qualifiers below the current directory are treated as entries in the directory (disables DIRECTORyMode).

**DB2**

Specifies the name of the DB2 subsystem.

```plaintext
   db2_name
```

The name of the DB2 subsystem.

**DCbdsn**

Specifies the name of the MVS data set to be used as a model for allocation of new data sets. Specifying DCbdsn with no parameter value cancels the DCbdsn specification.

```plaintext
   data_set_name
```

Specifies the name of the data set. The file name must be an MVS data set name. HFS file names are not allowed on the DCbdsn parameter. The setting of QUOtesoverride is ignored. If the file name is enclosed in single quotation marks, it overrides the current working directory; otherwise it is appended to the current working directory.

**Notes:**

1. Specify Slte RECfm, LRecl, and BLKsize parameters with no values to allow characteristics from the model DCB to be used.
2. To override the model characteristics of RECfm, LRecL, BLKsize, or RETpd, specify a value on the Slte command.
3. Ensure that SENDSite subcommand is toggled off. Otherwise, the Slte information that is sent automatically by the client overrides the values provided by the model DCB.
4. If MGmtclass is specified, the RETpd value of the MGmtclass may override the RETpd value.

Specifying a GDG data set with a relative index produces an error message. The following examples are unsupported specifications:

- PRImary
- RECfm
- RETpd
- SECondary
SITE DCBDSN=MYGDG(0)
SITE DCBDSN=MYGDG(–nnn) or
SITE DCBDSN=MYGDG(+nnn)

See "Steps for using a DCBDSN model to create a new data set" on page 87 for more information about DCBdsn.

DEBug
Activates or disables general tracing at the FTP server. One or more traces can be activated with a single debug parameter with the following options:

? Displays the status of the traces.

ACC
The ACC trace shows the details of the login process.

ALL
This parameter is used to set all of the trace points.

Note: Both the FSC and the SOC traces will be set to level 1 when the ALL parameter is processed.

BAS
This parameter is used to set a select group of traces that offer the best overall details without the intense tracing of some of the traces. Specifying this parameter is the same as SITE DEBUG=(CMD, INT, FSC, SOC).

CMD
The CMD trace shows each command and the parsing of the parameters for the command.

FLO
The FLO trace shows the flow of control within FTP. It is useful to show which services of FTP are used for an FTP request.

FSC(n)
The FSC trace shows details of the processing of the file services subcommands APPE, STOR, STOU, RETR, DELE, RNFR, and RNTO.

This trace can generate very detailed information and therefore allows you to specify levels of detail for the trace points. The level (n) can be a number from 1 to 5.

INT
The INT trace shows the details of the initialization and termination of the FTP session with the server.

JES
The JES trace shows details of the processing for JES requests (that is, requests when SITE FILETYPE=JES is in effect).

NONE
This parameter is used to turn off all of the traces.

PAR
The PAR trace shows details of the FTP command parser. It is useful in debugging problems in the handling of the command parameters.

SEC
The SEC trace shows the processing of security functions such as TLS and GSSAPI negotiations.

SOC(n)
The SOC trace shows details of the processing during the setup of the
interface between the FTP application and the network as well as details of the actual amounts of data that is processed. This trace can generate very detailed information and therefore allows you to specify levels of detail for the trace points.

The level 1 tracing that is specified by entering SOC or SOC(1) is the level normally used unless more data is requested by the TCP/IP service group. The level \( (n) \) can be a number from 1 to 8.

**SQL**

The SQL trace shows details of the processing for SQL requests, such as requests when SITE FILETYPE=SQL is in effect.

**UTL**

The UTL trace shows the processing of utility functions such as CD and Site.

**Xyy**

This syntax is used to turn off (reset) a trace that is named by yyy. For example, SITE DEBUG=(XPAR, XACC) will turn off the PAR and the ACC traces.

Usage notes for the DEBug parameter:

- The client is allowed to change general tracing at the server if the FTP.DATA file for the server specified DEBUGONSITE TRUE.

- The state of the debug traces points is displayed as a response to the Site subcommand. To see the state without making a change, enter SITE DEBUG=(?).

- The setting of the traces is additive. This is demonstrated by the following example:

  ```
  SITE DEBUG=(BAS)
  200-Active traces: CMD INT FSC(1) SOC(1)
  200 Site command was accepted
  SITE DEBUG=(ACC)
  200-Active traces: CMD INT ACC FSC(1) SOC(1)
  200 Site command was accepted
  ```

- To ensure that only the needed traces are active, use the NONE value to clear all traces before setting the requested ones.

  ```
  SITE DEBUG=(?)
  EZA1701I >>> SITE DEBUG=(?)
  200-Active traces: CMD INT FSC(1) SOC(1)
  200 Site command was accepted
  SITE DEBUG=(NONE,FSC(2))
  EZA1701I >>> SITE DEBUG=(NONE,FSC(2))
  200-Active traces: FSC(2)
  200 Site command was accepted
  ```

- For the FSC and SOC trace options only one level of tracing can be defined at any time. However, when level 2 is defined, levels 1 and 2 are active. When level 3 is defined, levels 1, 2 and 3 are active. This progression also applies to levels 4 and 5.

  ```
  site debug=(fsc(2),soc(1))
  >>> SITE debug=(fsc(2),soc(1))
  200-Active traces: FSC(2) SOC(1)
  200 Site command was accepted
  site debug=(fsc(1),soc(2))
  >>> SITE debug=(fsc(1),soc(2))
  200-Active traces: FSC(1) SOC(2)
  200 Site command was accepted
  ```
**Note:** The FSC command will accept level values 6–8, but will only provide level 5 trace data. Likewise, SOC will accept level values 4–8, but will only provide level 3 trace data.

**DESt**

Indicates the Network Job Entry (NJE) destination to which the files are routed when you enter a PUT command. If specified without a *destination*, the destination resets and files are stored at the host system rather than sent to a remote network.

The Site DESt subcommand enables you to send data sets (rather than storing them at the server) to other users on machines that are connected on an NJE network.

*destination*

Indicates the NJE destination to which the files are routed when you enter a PUT command. The value specified for destination can be:

- userID@nodeID
- nodeID.userID
- nodeID
- DestID

The file is sent over the NJE network to the specified destination.

This parameter is ignored if FILEtype=JES is set.

**Directory**

Specifies the number of directory blocks to be allocated for the directory of a PDS. When specified without the *size*, no directory value is used when allocating the data set. The equal sign (=) is optional in this case.

Specify Directory without a *size* when you are also specifying DATAClass=dataclass and you want the SMS data class to provide the Directory *size*. If Directory=*size* is specified with DATAClass, the value specified by the Site Directory parameter overrides the DATAClass directory specification.

*size*

Specifies the number of directory blocks to be allocated for the directory of a PDS. The valid range is 1—16 777 215.

**DIRECTORYMode**

Specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory. In directory mode, this data set qualifier is the only one used by the MGet, LS, and Dir subcommands.

DIRECTORYMode has no effect on files residing in an HFS.

**DUMP**

Activates or disables extended tracing at the FTP server.

**Note:** Extended tracing has the potential to generate a large amount of trace data and should not be set unless requested to debug a specific problem in the code.

One or more traces can be activated with a single dump parameter with the following options:

- ? Displays the status of the traces.
- n Specifies the ID number of a specific extended trace point that is to be activated in the FTP code. The number has a range of 1–99.
ALL
This parameter is used to set all of the trace points.

FSC
Activates all of the extended trace points in the file services code. The ID numbers for FSC are 20 to 49.

JES
Activates all of the extended trace points in the JES services code. The ID numbers for JES are 60 to 69.

NONE
This parameter is used to turn off all of the traces.

SOC
Activates all of the extended trace points in the network services code. The ID numbers for SOC are 50 to 59.

SQL
Activates all of the extended trace points in the SQL services code. The ID numbers for SQL are 70 to 79.

Xxyy
This syntax is used to turn off (reset) a trace that is named by yyy. For example, SITE DUMP=(X21,X22,XSQL) will reset the extended trace points 21 and 22 and all of the SQL trace points.

Usage notes for the DUMP parameter:
- The client is allowed to change extended tracing at the server if the FTP.DATA file for the server specified DUMPONSITE TRUE.
- The setting of the traces is additive. This is demonstrated by the following example:

  SITE DUMP=(NONE,21)
  EZA1701I >>> SITE DUMP=(NONE,21)
  200-Active dumpIDs: 21
  200 Site command was accepted
  SITE DUMP=(22)
  EZA1701I >>> SITE DUMP=(22)
  200-Active dumpIDs: 21 22
  200 Site command was accepted

- The range of 99 extended trace points is defined to allow easy extension of the trace points by the TCP/IP service team. Additional trace points can be added to the code without any changes to the external mechanism to control the traces.
- Refer to the z/OS Communications Server: IP Diagnosis for the definition of the extended trace points.

ENCODING
Indicates the kind of encoding that is used for conversions between codepages for data transfers.

SBCS
Single Byte encoding. Code pages are specified using the SBDATACONN statement. This is the default value.

MBCS
Multibyte encoding. Code pages are specified using the MBDATACONN statement.

FILEtype
 Specifies the file type of the data set.
The file type of the data set can be:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQ</td>
<td>Sequential or partitioned data sets</td>
</tr>
<tr>
<td>SQL</td>
<td>SQL query function</td>
</tr>
<tr>
<td>JES</td>
<td>Remote job submission</td>
</tr>
</tbody>
</table>

**ISPFSTATS**

Allows FTP to create or update ISPF Member statistics when PUT, MPut or APPend subcommands are issued.

**JESENTRYLimit**

JESENTRYLimit specifies how many entries can be displayed at once via a LIST or NLST command. JESENTRYLIMIT is only valid if JESINTERFACELEVEL=2.

- **number**
  - Fixed number of entries to display. The default for JESENTRYLimit is 200, if not specified in the server FTP.DATA file.

**JESJOBName**

Specifies that any command (Get, LIST, DIR or MGet) should be limited to those jobs, started tasks, APPC/MVS, or TSO output that match the specified value. JESJOBName is only accepted if JESINTERFACELevel is set to 2.

- **jobname**
  - Specified job name. Can be or contain a wildcard (* or ?).

- **loginuserid**
  - The logged in user ID appended with an asterisk (*). Default value.

**Note:** JESJOBName matches the first job name that a job is assigned. Jobs that change job names during execution time are matched only by their initial job name.

**JESLrecm**

Specifies the logical record length (LRecl) for the Job Entry System (JES) internal reader at the foreign host.

- **length**
  - The logical record length for the JES internal reader at the foreign host. The valid range is 1–254.
  - * Indicates that the logical record length should be taken from the site LRecl parameter setting.

**JESOwner**

Specifies that any command (Get, LIST, DIR or MGet) should be limited to those jobs, started tasks, APPC/MVS, or TSO output which are owned by the user ID specified. JESOWNER cannot be modified unless JESINTERFACELEVEL is set to 2.

- **userid**
  - Specified user ID. The userid can be or contain a wild card (* or ?).

- **loginuserid**
  - The logged in user ID. Default value.

**JESRecfm**

Specifies the record format for the JES internal reader at the foreign host.
Fixed record format

Variable record format

* Indicates that the record format should be taken from the Site RECfm parameter setting.

**JES**Status

Specifies what type of information should be returned on LIST and NLST commands. Acceptable values are INPUT, ACTIVE, OUTPUT or ALL. The default value for JESSTATUS is ALL. JESSTATUS can only be modified if JESINTERFACELEVEL=2.

**LRecl**

Specifies the logical record length (LRecl) of a newly allocated data set. When specified without a length, no LRecl is used when allocating the data set. The equal sign (=) is optional in this case.

Specify LRecl with no value when you are also specifying DATAClass=data_class and you want the SMS data class to provide the LRecl value, or when you are specifying DCbdsn=data_set_name and you want to use the LRecl from the DCBDSN data set. If LRecl=length is specified with either DATAClass or DCbdsn, the length specified by the Site LRecl parameter overrides the DATAClass or DCbdsn LRecl.

length

Specifies the logical record length of a newly allocated data set. The valid range is 0—32760. A special value of x (LRecl=x) is also supported to indicate that a logical record length can exceed 32760 for variable-length spanned records.

Specifying LRecl=0 has the same effect as specifying LRecl with no parameters.

**MBDATACONN=(file_system_codepage,network_transfer_codepage)**

Specifies the codepages for the file system and for the network transfer used when data is translated during a data transfer.

file_system_codepage

Specifies the name of the file system codepage. The supported codepages are IBM-1388 and UTF-8.

network_transfer_codepage

Specifies the network transfer code page. The supported codepage is IBM-5488.

**MGmtclass**

Specifies the SMS management class as defined by your organization for the target host. Specifying MGmtclass with no mgmtclass cancels the management class specification. The equal sign (=) is optional in this case.

mgmtclass

Specifies the SMS management class as defined by your organization for the target host. If the mgmtclass specified has a setting for RETpd, the value specified by the mgmtclass may override the setting of the RETpd site parameter, the RETpd value of a model data set if the DCbdsn parameter is specified, and the RETpd value defined in an SMS data class if DATAClass is specified. See “Specifying values for new data sets” on page 86 for more information about specifying attributes when allocating new data sets.
MIGratevol
Specifies the volume ID for migrated data sets if they do not use IBM Storage Management Subsystems. If you do not specify MIGratevol, the default volume_serial is MIGRAT.

volume_ID
The volume ID for migrated data.

NOASAtrans
Treats ASA file transfers as regular file transfers; that is, the ASA characters are treated as part of the data and are not converted to print control characters.

NOAUTOMount
Prevents automatic mounting of volumes for data sets on volumes that are not mounted.

NOAUTORecall
Prevents automatic recall of migrated data sets.

Note: A migrated data set can be deleted even though NOAUTORecall is specified, because migrated data sets are not recalled for deletion.

NOISPFSTATS
Does not allow FTP to create or update ISPF Member statistics when PUT, MPUT or APPEND subcommands are issued.

NOQUOtesoverride
Treats a single quotation mark appearing at the beginning of the file name, as well as all other single quotation marks contained in the file name, as part of the actual file name. The entire file name, including the leading single quotation mark, is appended to the current working directory.

NORDW
Specifies that Record Descriptor Words (RDWs) are not treated as if they were part of the record and are discarded during FTP transmission of variable format data sets.

NOSBSUB
Specifies that substitution is not allowed for data bytes that cannot be translated in a single-byte character translation. This causes a data transfer failure if a character cannot be translated during the transfer.

NOSPRead
Specifies that the output is in report format rather than spreadsheet format when the file type is SQL.

NOTRAILingblanks
Specifies that the FTP server does not preserve the trailing blanks that are in a fixed format data set when the data is sent to the foreign host.

NOTRUNcate
Specifies that truncation is not permitted. The FTP server will set an error and fail file transfer if a record that is longer than the LRECL of the new file is detected.

Note: If WRAPRECORD is set then the data is wrapped, not truncated, no error will be set and the file transfer will continue.

NOUCSSUB
In Unicode-to-EBCDIC conversion, the data transfer is terminated if any Unicode character cannot be converted into the EBCDIC code set.
**NOUCSTRUNC**
In Unicode-to-EBCDIC conversion, truncation of EBCDIC data is not allowed. The data transfer is aborted if the logical record length of the receiving data set is too small to contain the data after conversion to EBCDIC.

**Note:** The setting of the CONDdisp parameter determines what happens to the target data set if the transfer is aborted.

**NOWRAPrecord**
Indicates that data is truncated if no new line character is encountered before the logical record length of the receiving file is reached.

**Note:** If NOWRAPrecord and NOTRUNcate are set, then an error will be set and the file transfer will fail.

**NOWRTAPEFastio**
Indicates that ASCII Stream data that is being written to tape must be written using the LE Run-Time Library.

**PRLimary**
Specifies the amount of tracks, blocks, or cylinders for primary allocation. When specified without an *amount*, no primary value is used when allocating the data set. The equal sign (=) is optional in this case.

Specify PRLimary with no value when you are also specifying DATAClass=**data_class** and you want the SMS data class to provide the PRLimary *amount*.

To allow the SMS data class to determine the space allocation, both PRLimary and SECondyary must be specified with no value. The tracks, blocks, and cylinders setting is ignored in this case. If PRLimary=**amount** is specified with DATAClass, the value specified by the Site PRLimary parameter overrides the DATAClass space allocation.

*amount*
Specifies the amount of tracks, blocks, or cylinders for primary allocation. For allocating partitioned data sets, this is the amount that is allocated for the primary extent.

For allocating sequential data sets this is the maximum amount that is allocated for the primary extent. If a lesser amount is needed to hold the data being transferred, only the amount actually needed to hold the data is allocated. The valid range is 1—16,777,215.

**Qdisk**
Used to display statistics about available space on a volume. If Qdisk is entered without a specific *volume_serial*, statistics about available space are displayed for each volume that is defined with “Use Attribute=storage.”

*volume_serial*
Displays statistics about available space on a specific volume.

**QUOtesoverride**
Specifies single quotation mark at the beginning and end of a file name should override the current working directory instead of being appended to the current working directory. This is the way single quotation marks are used in all previous MVS FTP servers, and is the default. Any single quotation mark inside the beginning and ending quotation mark are treated as part of the file name.
QUOtesoverride indicates the usage of single quotation mark appearing at the beginning of, or surrounding, a file name. The setting of this keyword affects all FTP subcommands that have a path name as a parameter except keywords on the SItc subcommand.

READTAPEFormat
Used to provide information about an input data set on tape. If specified without the tape_format (which is the default), processing of input tapes will not take advantage of the record format information prior to open. The equal sign (=) is optional in this case.

READTAPEFormat has no effect on, and is not affected by DATAClass, DCbdsn, JESLrecl, JESRecfm, LRecl, RECfm, or any other parameters associated with creating a data set.

tape_format
Specifies the format of the records on the input tape data set. Valid formats are:

- F Fixed record length
- V Variable record length
- S Spanned records
- X Logical record length is undefined (Lrecl X)
- blank Unspecified

The formats are mutually exclusive. Spanned implies variable, and Lrecl X implies spanned. If specified, the tape_format must be the most inclusive identifier in the list that matches the tape. If it is not the most inclusive identifier, an error message will be issued. For example, if tape_format is S (spanned), and the tape contains records with undefined length (Lrecl X), the request will fail. An unspecified format will avoid this type of error. However, the following should be considered:

- Specify a value for READTAPEFormat in all the following cases. Failure to specify a format will likely cause errors in processing the tape.
  - The record length is undefined (Lrecl X).
  - The records are spanned (Recfm is VBS, VS).
  - The records are variable (Recfm is V, VB, VBA) and RDW is specified.
- Specify a value for READTAPEFormat for all input tapes with one of the listed formats to ensure best results.

RDW
Specifies that Record Descriptor Words (RDWs) are treated as if they were part of the record and not discarded during FTP transmission of variable format data sets in stream mode.

Note: RDW information is stored in binary. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.

RECfm
Specifies the record format of a data set. When specified without the format, no record format is used when allocating the data set. The equal sign (=) is optional in this case.
Specify RECfm with no value when you are also specifying DATAClass=\texttt{data\_class} and you want the SMS data class to provide the RECfm format, or when you are specifying DCbdsn=\texttt{data\_set\_name} and you want to use the record format from the DCBDSN data set.

If RECfm=\texttt{format} is specified with either DATAClass or DCbdsn, the value specified by the Site RECfm parameter overrides the DATAClass or DCbdsn record format.

\textbf{format}

Specifies the record format of a data set. Valid record formats are: F, FA, FB, FBA, FBM, FBS, FBSA, FBSM, FM, FS, FSA, FSM, U, UA, UM, V, VA, VB, VBA, VBM, VBSA, VBSM, VBS, VM, VS, VSA, and VSM. The characters used to specify these record formats have the following meanings:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Fixed record length</td>
</tr>
<tr>
<td>V</td>
<td>Variable record length</td>
</tr>
<tr>
<td>U</td>
<td>Undefined record length</td>
</tr>
<tr>
<td>B</td>
<td>Blocked records</td>
</tr>
<tr>
<td>S</td>
<td>Spanned records (if variable) / standard records (if fixed)</td>
</tr>
<tr>
<td>A</td>
<td>Records contain ISO/ANSI control characters</td>
</tr>
<tr>
<td>M</td>
<td>Records contain machine code control characters</td>
</tr>
</tbody>
</table>

\textbf{RETpd}

Specifies the number of days that a newly allocated data set should be retained. When specified without the number of \textit{days}, a retention period will not be specified when allocating new data sets. The equal sign (=) is optional in this case.

Specify RETpd with no value when you are also specifying DATAClass=\texttt{data\_class} or MGmtclass=\texttt{mgmtclass} and you want SMS to provide the RETpd value, or when you are specifying DCbdsn=\texttt{data\_set\_name} and you want to use the RETpd from the DCBDSN data set. If more than one of the Site parameters (RETpd, MGmtclass, DATAClass, or DCbdsn) are specified, the order of precedence (highest to lowest) is:

1. MGmtclass
2. RETpd
3. DCbdsn
4. DATAClass

If a retention period is associated with an SMS management or data class, or with a model DCBDSN data set, the value of the retention period may be overridden to another retention period, but it cannot be overridden to have no retention period specified for the newly created data sets.

\textbf{days}

Specifies the number of days that a newly allocated data set should be retained. The valid range is 0—9999. A value of 0 indicates a retention period of 0 days so that the data set expires the same day it was created.

\textbf{Note:} An attempt to either append or replace an existing data set with a retention period requires operator interaction to take place for permission to alter the data set. This is normal MVS behavior.
**SBDataconn**

Specifies the conversions between file system and network code pages to be used for data transfers. Valid subcommands are:

`SITE SBDataconn=dsname`

`SITE SBDataconn=(file_system_cp, network_transfer_cp)`

`SITE SBDataconn=*

*dsname*

Specifies the fully qualified name of an MVS data set or HFS file that contains the EBCDIC-to-ASCII and ASCII-to-EBCDIC translate tables generated by the CONVXLAT utility.

**Notes:**

1. The name must not be enclosed in quotation marks. If quotation marks appear, they are treated as part of the name. (QUOtesoverride is ignored.)
2. The HFS name is case sensitive. The MVS name is not.
3. The name cannot begin with a left parenthesis `[(`.
4. The SBDataconn keyword must be the only keyword or the last keyword on a SITE subcommand.
5. The translate tables being used for the data connection can also be changed by a SITE XLate subcommand.
6. SITE XLate and SITE SBDataconn are mutually exclusive.

*file_system_cp*

Specifies the name of a code page recognized by iconv. For a list of code pages supported by iconv, refer to code set converters in the [z/OS C/C++ Programming Guide](#).

*network_transfer_cp*

Specifies the name of a code page recognized by iconv. For a list of code pages supported by iconv, refer to code set converters in the [z/OS C/C++ Programming Guide](#).

*FTP_STANDARD_TABLE*

Indicates that the FTP internal tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), are to be used on the data connection.

* * Indicates the translate tables set up at initialization for the data connection must be used.

*SBSUB*

Specifies that substitution is allowed for data bytes that cannot be translated in a single byte character translation. The substitution character is specified by the SBSUBCHAR parameter.

*SBSUBCHAR nn*

Specifies the value that is used for substitution when SBSUB is also specified. The value is one of the following:

**SPACE**

When the target codeset is ASCII, replace untranslatable characters with X'20' during SBCS data transfers. When the target codeset is EBCDIC, replace untranslatable characters with X'40' during SBCS data transfers.

*nn* Replace untranslatable characters with *nn* during SBCS data transfers where *nn* is a hexadecimal value from 00 to FF.
SECondy
Specifies the amount of tracks, blocks, or cylinders for secondary allocation. When specified without the amount for the C server, no secondary value is used when allocating the data set. The equal sign (=) is optional in this case.

Specify SECondy with no value when you are also specifying DATAClass=dataclass and you want the SMS data class to provide the SECondy value. To allow the SMS data class to determine the space allocation, both PRImary and SECondy must be specified with no value. The tracks, blocks, or cylinders setting is ignored in this case. If SECondy=amount is specified with DATAClass, the value specified by the Site SECondy parameter overrides the DATAClass space allocation.

amount
Specifies the amount of tracks, blocks, or cylinders for secondary allocation. The valid range is 0—16 777 215.

SPRead
Specifies that the output is in spreadsheet format when the file type is SQL.

SQLCol
Specifies the column headings of the SQL output file.

any
The label of the DB2 SQL table column heading is the first choice for column heading, but if there is no label, the name becomes the column heading.

labels
Labels are the DB2 SQL table column headings. If any of the columns do not have labels, FTP supplies a column heading in the form of COLnnn.

names
Uses the names of the DB2 SQL table column headings. The labels are ignored.

STOrclass
Specifies the SMS storage class as defined by your organization for the target host. Cancels the storage class specification when specified without a storage_class parameter value. The equal sign (=) is optional in this case.

See “Specifying values for new data sets” on page 86 for more information about specifying attributes when allocating new data sets.

storage_class
Specifies the SMS storage class as defined by your organization for the target host. The SMS storage class may override settings for the VOLume or Unit site parameters.

TRacks
Specifies that primary and secondary space allocations are in tracks. If both PRImary and SECondy are unspecified, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the TRacks parameter is ignored.

TRAILingblanks
Specifies that the FTP server preserves the trailing blanks that are in a fixed format data set when the data is retrieved from a foreign host.

TRUNcate
Specifies that truncation is permitted. The FTP server does not set an error when a truncated record is detected and file transfer continues.
UCOUNT
Specifies how many devices to allocate concurrently to support the allocation request.

unit_count
Specifies number of devices to allocate. Valid value is 1–59. When specified without a value, the FTP server does not specify a unit count when allocating data sets.

P Parallel mount request.

The setting for UCOUNT may be overridden by the SMS storage class, if one is specified. Therefore, it is recommended that UCOUNT not be coded if an SMS storage class is in use.

UCSHOSTCS
Specifies the EBCDIC code set to be used when converting to and from Unicode. If you do not specify a code_set, the current code set is used.

code_set
Name of the EBCDIC code set to be used when converting to and from Unicode.

UCSSUB
In Unicode-to-EBCDIC conversion, the EBCDIC substitution character is used to replace any Unicode character that cannot successfully be converted. Data transfer continues.

UCSTRUNC
In Unicode-to-EBCDIC conversion, truncation of EBCDIC data is allowed. The data transfer continues even if EBCDIC data is truncated.

Note: If the EBCDIC data contains any double-byte data, truncation may not honor character boundaries and EBCDIC records may not end in Shift-in state.

UMask
Defines the file mode creation mask. The file mode creation mask defines which permission bits are not to be set on when a file is created. When a file is created, the permission bits requested by the file creation are compared to the file mode creation mask, and any bits requested by the file creation which are disallowed by the file mode creation mask are turned off.

The format of the UMask keyword is UMASK ooo.

When a file is created, the specified permission bits for the file are 666 (-rw-rw-rw-). If the file mode creation mask is 027, the requested permissions and the file mode creation mask are compared:

```
110110110 - 666
000010111 - 027
---------
11010000 - 640
```

The actual permission bits set for the file when it is created is 640 (-rw-r-----).

Notes:
1. The default value for UMask is 027.
2. You cannot use FTP to create HFS files with execute permissions. If you require execute permissions, use the SITE CHMod command to change permissions after the file has been created.
**Unit**

Specifies the unit type for allocation of new data sets.

The setting for Unit may be overridden by the SMS storage class, if one is
specified. Therefore, Unit should not be coded if an SMS storage class is in
use.

*unit_type*

The unit type (for example, 3380) for the allocation of new data sets on
direct access devices. If *unit_type* is not specified, the unit type used for
allocation is set back to the system default.

**VCOUNT**

Specifies the number of volumes an allocated data set may span. When
specified without a *volume_count* value, the FTP server uses a volume count of
50 when allocating data sets.

*volume_count*

Valid values are from 1–255.

**VOLUME**

Specifies the volume serial number for allocation of new data sets.

*volume_serial*

The serial number of the volume to use for allocation.

*volume_serial_list*

A list of one or more volume serial numbers for allocation. Delimit each
volume_serial from the prior one with a comma.

If *volume_serial_list* or *volume_serial* is not specified, the volume serial number
used for allocation is set back to the system default.

The setting for VOLUME may be overridden by the SMS storage class, if one is
specified. Therefore, VOLUME should not be coded if an SMS storage class is
in use.

The MVS FTP server identifies multiple commands issued with a single-site
command by the white space. For example (note the white space in the two
commands): *site vol=fffff* is a single-site command; however, *site vol =
fffff* is treated by the server as three different commands.

**WRAPrecord**

Indicates that data is wrapped to the next record if no new line character is
encountered before the logical record length of the receiving file is reached.

**WRTAPEFastio**

Indicates that ASCII Stream data that is being written to tape is allowed to be
written using BSAM I/O.

**XLate**

Indicates the desired translate table to be used for the data connection. Valid
subcommands are:

SITE XLate=name
SITE XLate**

*name*

Specifies the name that corresponds to the desired translate table data set. The
corresponding data set name is hlq.name.TCPXLBIN unless
environment variable _FTPXLATE_ = *dsn* was defined for the FTP
server to override the data set name. In that case, *dsn* is the data set used.
* Indicates the translate tables set up at initialization for the data connection are to be used.

Notes:
1. The translate tables being used for the data connection can also be changed by an SBDdataconn subcommand.
2. Site XLate and Site SBDdataconn are mutually exclusive.

Usage

- Because more than one parameter can be specified with the Site subcommand, parameter can be repeated many times, with each parameter separated by a blank space.
- If the remote host is using TCP/IP, use the Site subcommand to set data set routing and send data set allocation attributes to the host. The site-dependent information sent with the Site subcommand remains active until you issue a new Site subcommand. The new Site subcommand adds to, or changes, the attributes established by previous Site subcommands.
- If you specify one or more incorrect parameters with the Site subcommand, an error message specifying the incorrect parameter is displayed. All correct parameters are set, regardless of any incorrect parameters, and do not need to be reissued.
- If the data sets are sent from an MVS host, use the SENDSite subcommand to automatically invoke the Site subcommand.

If the local host does not support the Site subcommand, site information can be sent to the MVS FTP server from the client using the QUOte subcommand.
- Use the HELP SERVER Site command to get information on the Site parameters supported by a server.

Context

- To check the effect of the Site subcommand on the attributes at the foreign host, see "STATus subcommand—Retrieve status information from a remote host" on page 267.
- See "SENDSite subcommand—Toggle the sending of site information" on page 239 for more information about the SENDSite subcommand.
- See "QUOte subcommand—Send an uninterpreted string of data" on page 229 for more information about the QUOte subcommand.
- Refer to the z/OS MVS JCL Reference for more information about some of the Site and LOC Site parameters.
SJiskanji subcommand—Change the data transfer type to SJISKANJI

Purpose

Use the SJiskanji subcommand to change the data transfer type to SJISKANJI.

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the Site/LOCSSite SBDdataconn or Site XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

Format

```
SJiskanji
```

Parameters

Sosi
Transferred data contains the shift-out and shift-in characters specified by one of the following parameters — Ascii, Ebcdic or Space. If no parameter is specified, ASCII is used as the default.

If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

Ascii
When combined with the Sosi parameter, causes shift-out and shift-in characters X’1E’ and X’1F’ to be used to delimit DBCS strings in ASCII data.

Ebcdic
When combined with the Sosi parameter, causes shift-out and shift-in characters X’0E’ and X’0F’ to be used to delimit DBCS strings in ASCII data.

Space
When combined with the Sosi parameter, causes shift-out and shift-in characters X’20’ and X’20’ (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSo
Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

NOType
Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.
Examples

To cause the FTP client to change its transfer type to Shift JIS kanji, without
sending a TYPE command to the FTP server, use:
SJISKANJI (NOTYPE

The server in this example should be set to the ASCII transfer type before the
(NOTYPE subcommand is issued.

Usage

• The SJis kanji client subcommand is equivalent to the TYPE B 1 server
  command.
• The SJis kanji (Sosi or SJis kanji (Sosi ASCII client subcommands are equivalent
to the TYPE B 1 S A server command.
• The SJis kanji (Sosi EBcdic client subcommand is equivalent to the TYPE B 1 S
  E server command.
• The SJis kanji (Sosi SPACE client subcommand is equivalent to the TYPE B 1 S
  S server command.
• The SJis kanji (NOSO client subcommand is equivalent to the TYPE B 1 N server
  command.

Context

See [FTP with DBCS support on page 123](#) for more information.
SRestart subcommand—Restart a stream data transfer

Purpose
Use the SRestart subcommand to restart an interrupted stream mode data transfer.

Format

```
SRestart
Get  foreign_file  local_file
PuT  local_file  foreign_file
```

Parameters

Get
Resume a get subcommand.

- **foreign_file**
  Specifies the name of the file to be retrieved from the remote host.

- **local_file**
  Specifies the name of the local file created as a result of the Get subcommand.

  Enter the same names for `foreign_file` and `local_file` as you used in the interrupted get subcommand.

  You can override the usage of the current local working directory in the local file name by specifying `local_file` as a complete data set name enclosed in single quotation marks (`'`). If `local_file` is not specified, the `local_file` name is the same as the `foreign_file` name.

PuT
Resume a put subcommand.

- **local_file**
  Specifies the name of the file on your local host being sent to the remote host.

- **foreign_file**
  Specifies the name that the delivered file is given on the remote host. If the `foreign_file` name is not specified, the `foreign_file` name is the same as the `local_file` name.

  Enter the same names for `foreign_file` and `local_file` as you used in the interrupted put subcommand.

  You can override the usage of the current local working directory in the local file name by specifying `local_file` as a complete data set name enclosed in single quotation marks (`'`). If `local_file` is not specified, the `local_file` name is the same as the `foreign_file` name.

Usage

- Do not use SRestart to resume an interrupted file transfer unless you can recreate the start options, FTP.DATA statements, S1te, and LOCSIte options in effect at the time of the failed file transfer. Unpredictable results may occur if the
file transfer environment cannot be recreated. If you cannot recreate the file transfer environment, issue Get or PUT to transfer the file again.

- Not all file transfers can be resumed in STREAM mode. Observe these restrictions during the original file transfer as well as the srestart transfer.
  - Mode must be STREAM.
  - Filetype must be SEQ.
  - Structure must be FILE.
  - SUnique option must be disabled.
  - Data type must be ASCII, EBCDIC, or Image.
  - The local file must reside in the HFS.
  - For SRestart PUT, a z/OS CS V1R4 or later server will reject the restart if the foreign_file is not an HFS file.

- For SRestart PUT, if the server cannot calculate an appropriate point to restart the file transfer, SRestart will fail.
- For SRestart GET, if the client cannot calculate an appropriate point to restart the file transfer, SRestart will fail.
- If SRestart fails, use Get or PUT and start the file transfer again.

**Context**

- See “SUnique subcommand—Changes the storage method” on page 271 for information about changing the storage method on the remote host.
- See “STRucture subcommand—Set the file structure” on page 270 for more information.
**STAtus subcommand—Retrieve status information from a remote host**

**Purpose**

Use the STAtus subcommand to retrieve current configuration information from the FTP server. This information includes the current settings of the configuration variables, which can be initialized in the FTP.DATA data set or changed using various FTP subcommands. For information on the parameters of the FTP.DATA data set, refer to the *z/OS Communications Server: IP Configuration Reference*.

**Format**

```
STAtus name
```

**Parameters**

`name`

Specifies the file or directory for which status information is requested. The `name` parameter is not supported by the z/OS FTP server.
Examples

The following is an example of retrieving the status information from an FTP server:

```
status
>>>STAT
211-Server FTP talking to host 9.117.222.59, port 23467
211-User: USER33 Working directory: /u/user33
211-The control connection has transferred 395 bytes
211-There is no current data connection.
211-The next data connection will be actively opened
to host 9.117.222.59, port 23467,
using Mode Stream, Structure File, type EBCDIC, byte-size 8
211-Automatic recall of migrated data sets.
211-Automatic mount of direct access volumes.
211-Auto tape mount is allowed.
211-Inactivity timer is disabled
211-VCOUNT is 59
211-ASA control characters in ASA files opened for text processing
will be transferred as ASA control characters.
211-Trailing blanks are removed from a fixed format
211-data set when it is retrieved.
211-Data set mode. (Do not treat each qualifier as a directory.)
211-ISPSTATS is set to FALSE
211-Primary allocation 2 tracks. Secondary allocation 1 track.
211-Partitioned data sets will be created with 27 directory blocks.
211-FileType SEQ (Sequential - default).
211-Number of access method buffers is 5
211-RDWs from variable format data sets are discarded.
211-Records on input tape are unspecified format
211-SITE DB2 subsystem name is D7A
211-Data not wrapped into next record.
211-Tape write is not allowed to use BSAM I/O
211-Truncated records will not be treated as an error
211-JESRECL is 80
211-JESRECFM is Fixed
211-JESINTERFACELEVEL is 1
211-ENCoding is set to SBCS
211-SBSUB is set to FALSE
211-SBSUBCHAR is set to SPACE
211-SMS is active.
211-New data sets will be catalogued if a store operation ends abnormally
211-Single quotes will override the current working directory.
211-UMASK value is 027
211-Process id is 50331660
211-Checkpoint interval is 0
211-Authentication type: None
211-Record format FB, Lrecl: 80, Blocksize: 3120
211 *** end of status *** Command:
```

Usage

The retrieved status information can be a directory, a file, or general status information, such as a summary of activity. If name is omitted, general status information is retrieved.

For further information on setting values for server initialization in the FTP.DATA data set, refer to the [z/OS Communications Server: IP Configuration Reference](z/OS V1R4.0 CS: IP User's Guide and Commands).
STREam subcommand—Set the stream data transfer mode

Purpose

Use the STREam subcommand to set the data transfer mode to stream mode. This is equivalent to specifying the MOde S subcommand. See “MOde subcommand—Set the data transfer mode” on page 213 for more information.

Format

```
STREam
```

Parameters

There are no parameters for this subcommand.
STRucture subcommand—Set the file structure

Purpose
Use the STRucture subcommand to set the file structure.

Format

```
> [Z(qOOOO> [Z(qOOOO

STRucture

File

Record
```

Parameters

- **File**
  Sets the file structure to file. The file structure affects both the transfer mode and the interpretation and storage of the data set or file. With a file structure of F, the data being transferred is considered to be a continuous sequence of data bytes.

- **Record**
  Sets the file structure to record.
**SUnique subcommand—Changes the storage method**

**Purpose**

The SUnique subcommand changes the method of storing files on the foreign host.

**Format**

```
$Unique
- On Off
- NAME NONAME
```

**Parameters**

- **On**
  - Turns on store unique.

- **Off**
  - Turns off store unique.

- **NAME**
  - When specified with ON or OFF, instructs the client to include a name when sending a store-unique command to the server.

- **NONAME**
  - When specified with ON or OFF, instructs the client to omit a name when sending the store-unique command to the server.

**Usage**

- By default, the SUnique setting is OFF NAME and FTP uses a store command (STOR) with the PUt and MPut subcommands. If the foreign host already has a data set or file with the name specified by `foreign_file`, the foreign host overwrites the existing data set or file.

- If SUnique is set to ON, FTP uses a store-unique command (STOU) with the PUt and MPut subcommands and prevents you from overwriting or erasing the existing data set or file on the foreign host. If the default setting of NAME is in effect, a name string will be sent to the server with the store-unique command. The created foreign data set or file is stored with a unique name. FTP sends the unique name of the created foreign data set or file to the local host, where the data set or file name is displayed on your terminal.

- SUnique with no parameters toggles the ON/OFF setting. If ON or OFF is specified, SUnique will be set to that value regardless of its current setting. The NAME/NONAME setting can be changed as SUnique is turned ON or OFF. It is in effect when SUnique is ON and does not change for the session until another NAME or NONAME setting is specified.
SYstem subcommand—Display the operating system name

Purpose

Use the SYstem subcommand to display the name of the remote host operating system. The remote host must have also implemented the SYST subcommand.

Format

```
> [Z(qOOOO> [Z(qOOOO
SYstem
``` 

Parameters

There are no parameters for this subcommand.

Usage

Use this subcommand to determine the operating system at the server. The reply from the server has as its first word one of the system names from the Protocol Numbers and Assignment Services list for Operating System Names. The names are maintained by the Internet Assigned Numbers Authority [http://www.iana.org](http://www.iana.org).

For the z/OS server, the reply is determined by the current working directory at the server.

- If the current working directory is an hfs directory, the reply is:
  
  `215 UNIX is the operating system of this server. FTP Server is running on z/OS.`

- Otherwise, the reply is:
  
  `215 MVS is the operating system of this server. FTP Server is running on z/OS.`
**TChinese subcommand—Change the data transfer type to TCHINESE**

**Purpose**

Use the TChinese subcommand to change the data transfer type to Traditional Chinese (5550).

MVS FTP uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, use the Site/LOCSIte SBDataconn or Site XLate subcommand to have the server (or client) change the SBCS translation for the data connection.

**Format**

```
TChinese
  (Ascii NOSo NOType Sosi Ebcdic Space)
```

**Parameters**

**Sosi**  
Transferred data contains the shift-out and shift-in characters specified by one of the following parameters — Ascii, Ebcdic or Space. If no parameter is specified, ASCII is used as the default.

If the S parameter is not specified at all, shift-out or shift-in characters are not used in the transferred data.

**Ascii**  
When combined with the Sosi parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

**Ebcdic**  
When combined with the Sosi parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

**Space**  
When combined with the Sosi parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

**NOSo**  
Specifies that the data transferred is pure DBCS (data with no SBCS characters) and is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

**NOType**  
Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when translation is to be done by the FTP client only.

**Usage**

The TChinese client subcommand is equivalent to the TYPE B 7 server command.
See "FTP with DBCS support" on page 123 for more information.
TSO subcommand—Use TSO commands

Purpose

Use the TSO subcommand to pass a Time Sharing Option (TSO) command to a local host TSO environment.

Format

```
[ ] Z(qOOOO ] [ ] Z(u3000)

TSO command_line
```

Parameters

`command_line`

Specifies a TSO command. Do not use synonyms.

Usage

The TSO subcommand is not available from batch.
TYpe subcommand—Set the data transfer type

Purpose

Use the TYpe subcommand to set the data transfer type for the client and server at the same time with one command. FTP supports the ASCII, EBCDIC, image (binary), Unicode, and two DBCS data transfer types.

Format

```
> [Z(qOOOO > [Z(qOOOO
TYpe
```

Parameters

- **A**  Sets the transfer type as ASCII. Specifying the ASCII transfer type has the same effect as using the ASScii subcommand. When the data transfer type is ASCII, FTP will translate outgoing files to ASCII before sending them, and translate incoming files from ASCII to the file system code page before storing them. ASCII is the default transfer type.

- **B**  Sets the transfer type as DBCS. Specifying the B transfer type with the appropriate options has the same effect as using the BIG5, EUckanji, HAngeul, JIS78kj, JIS83kj, Ksc5601, SJiskanji, SChinese, or TChinese subcommands. If B is specified alone, the second type parameter defaults to 1 and current transfer type is changed to Shift JIS kanji.

When you transfer double-byte data, the currently active SBCS translation table is used for SBCS characters in the data set. If necessary, use the SIte/LOCSIte SBDataconn or SIte XLate FTP subcommand to select an alternate SBCS translation table that is appropriate for your data before transferring your double-byte data.

- **B 1**  Changes current transfer type to Shift JIS kanji.

- **B 2**  Changes current transfer type to Extended UNIX Code kanji.
B 3
Changes current transfer type to JIS 1983 kanji.

B 4
Changes current transfer type to JIS 1978 kanji.

B 5
Changes current transfer type to Hangeul.

B 6

B 7
Changes current transfer type to Traditional Chinese (5550).

B 8
Changes current transfer type to Big-5.

B 9
Changes current transfer type to Simplified Chinese.

S
Transferred data contains shift-out and shift-in delimeters.

If S is specified alone, the second parameter defaults to A. Shift-out and shift-in characters X'1E' and X'1F' are used.

The S parameter can be used to control the use of shift-out (SO) and shift-in (SI) characters during DBCS data transfer for Big5, SChinese, Shift-JIS kanji, EUC kanji, Hangeul, KSC-5601, and TChinese.

If Sosi is not specified at all, shift-out or shift-in characters are not used in the transferred data.

S A
Use shift-out and shift-in characters X'1E' and X'1F' to delimit DBCS strings in the transferred data.

S E
Use shift-out and shift-in characters X'0E' and X'0F' to delimit DBCS strings in the transferred data.

S S
Use ASCII spaces (X'20') as shift-out and shift-in characters to delimit DBCS strings in the transferred data.

A
Use ASCII shift-in escape sequence E\textbackslash s C ( B. This is the default. (Used for DBCS data types JIS 1983 kanji and JIS 1978 kanji only.)

R
Use JISROMAN shift-in escape sequence E\textbackslash s C ( J. (Used for DBCS data types JIS 1983 kanji and JIS 1978 kanji only.)

N
Indicates the transfer is to be pure DBCS data (data with no SBCS characters) and that the data is to be transferred to or from EBCDIC DBCS data that contains no shift-out or shift-in delimiters.

When data is transferred from the EBCDIC host, the entire data set is assumed to be EBCDIC DBCS with no SO/SI characters in the data. The data is then converted to the required ASCII type and if any SO/SI option has been specified for the transferred data then the corresponding SO/SI characters are used to delimit the ASCII DBCS strings.

When transferring data to the EBCDIC host, no SO/SI characters are inserted, and if any SO/SI option is specified for the transferred data, the corresponding SO/SI characters are removed from the ASCII data and not replaced at the
host. The length of data may change during transfer to and from the EBCDIC host when pure DBCS is specified with any SO/SI option. When pure DBCS is specified by itself, the length of data does not change. If N is not specified, the shift-out or/shift-in characters X'0E' and X'0F' are used at the host.

**E** Sets the transfer type as EBCDIC. Specifying the EBCDIC transfer type has the same effect as using the EBcdic subcommand. The EBCDIC transfer type is intended for efficient transfer between hosts that use EBCDIC for their internal character representation.

**F** Sets the transfer type as EBCDIC IBM kanji. Specifying the IBM kanji transfer type has the same effect as using the Ibmkanji subcommand.

**F 1**

Change current transfer type to IBM (EBCDIC) kanji.

**I** Sets the transfer type as image (binary). Specifying the image transfer type has the same effect as using the BINary subcommand. With the image transfer type, data is sent as contiguous bits, packed into 8-bit bytes. The image transfer type is used for the efficient storage and retrieval of data sets or files, and for the transfer of binary data.

**U 2**

Sets the transfer type to Unicode UCS-2. Type U 2 has optional parameters:

**B** Specifies big-endian byte order for Unicode encoding. This is the default.

**L** Specifies little-endian byte order for Unicode encoding.

### Examples

- **Transfer text data to another host:**

  User: ascii
  System: >>>TYPE A
  200 Representation type is ASCII.
  Command:

- **Transfer binary data to another host:**

  User: type i
  System: >>>TYPE I
  200 Representation type is IMAGE.
  Command:

- **Transfer text data from an EBCDIC host to an EBCDIC host:**

  User: type e
  System: >>>TYPE E
  200 Representation type is Ebcdic NonPrint
  Command:

- **Transfer binary data from an EBCDIC host to an EBCDIC host:**

  User: type i
  System: >>>TYPE I
  200 Representation type is Image.
  Command:

- Set the transfer type to JIS 1983 kanji using the JISROMAN shift-in escape sequence ESC ( J:

  TYPE B 3 R
• Set the transfer type to Shift-JIS kanji using the EBCDIC SO/SI characters X'0E'/X'0F' in the transferred date:
  TYPE B 1 S E

Usage

• If no Sosi option is specified by the TYpe command for BIG5, SChinese, EUckanji, HAngeul, Ksc5601, SJiskanji, or TChinese, standard DBCS control is used for the data transfer. This means that no SO/SI characters are placed in the ASCII data when transferring from the (EBCDIC) host to ASCII and the value of each ASCII character is used to determine if it is a single-byte character or part of a double-byte character when transferring to the host. For JIS 1983 kanji and JIS 1978 kanji, three-character escape sequences are always used to delimit DBCS strings in mixed SBCS/DBCS ASCII data. These escape sequences cannot be altered by using the S, S A, S E, or S S parameters.

• If no Sosi option is specified, the length of data may change as it is transferred to or from the EBCDIC host since EBCDIC DBCS types on the host contain SO/SI characters in mixed SBCS/DBCS data to determine which characters are part of a DBCS string. Any of the above SO/SI options (S, S A, S E or S S) may be used for mixed SBCS/DBCS data so that the length of data does not change when transferred to or from the EBCDIC host. Use of three-character escape sequences for JIS 1983 kanji and JIS 1978 kanji means that the length of data for these types always change when transferring mixed SBCS/DBCS data to or from the EBCDIC host.

• Use ASCII spaces as SO/SI characters in the transferred data only for transfer from the EBCDIC host. Data can be transferred to the host when using this option but care must be taken as each ASCII space is interpreted as a shift-out or shift-in character and is replaced with the corresponding SO/SI character on the host.

Context

For more information about transfer methods, see Table 11 on page 51.
UCs2 subcommand—Change data transfer type to Unicode UCS-2

Purpose

Use the UCs2 subcommand to change the data transfer type to Unicode UCS-2.

Format

```
UCs2
```

Parameters

BIG

Specifies big-endian byte order for the Unicode encoding. This is the default.

LITTLE

Specifies little-endian byte order for the Unicode encoding.

NOTYPE

Suppresses the sending of the corresponding TYpe command to the server. Use this parameter when you want translation to be done by the FTP client only.

Usage

The UCs2 client subcommand is equivalent to the TYpe U 2 subcommand.
User subcommand—Identify yourself to a host or change your TSO user ID password

Purpose

Use the User subcommand to identify yourself to the host after opening a connection or, if the foreign host is a z/OS FTP server, you can change your TSO user ID password.

Format

```
User user_id
Password:
password/new_password/new_password :userdata
```

Parameters

```
user_id
Specifies your logon name on the host.

password
Specifies your current password on the host. If you do not supply a password when invoking the User subcommand, you are prompted to enter a password if the host requires a logon password.

new_password/new_password
Specifies your new password on the host. You must enter the password twice.

:userdata
The optional user data must be separated from the password information by a colon (:) and can be any combination of up to 200 nonblank characters and numbers—except the colon. Care should be taken when using the backslash character (\) in combination with other characters, which might be interpreted as an escape sequence by the C compiler.

account_information
An optional parameter that will be supplied to the remote FTP server if the server requests account information after receiving the password.
```

Usage

- Not all FTP servers support the userdata parameter. The z/OS FTP server interprets userdata as a character string and passes it to the server FTCHKPWD user exit routine.
- If you enter your password incorrectly, you are not prompted to enter the password again. You must reissue the User subcommand to enter the correct password.
- If you do not specify password/new_password/new_password on the User subcommand, you can specify it when you are prompted for the password after
entering the User subcommand. You can specify the User subcommand to change your TSO user_id password at any time during the FTP session.

- To avoid having your password print when issuing your user ID and password as part of a CLIST or batch job, enter your user ID and password on separate lines.
- Do not place any spaces between the passwords and the slashes (/), and the user data.
- You can use the NETRC data set to automatically log on to a remote host.

Context

For information about using NETRC, see "NETRC data set" on page 37.
Verbose subcommand—Toggle verbose mode

Purpose

Use the Verbose subcommand to toggle verbose mode. When in verbose mode the client displays message IDs. This subcommand is effective only when the client is running in a z/OS UNIX environment.

Format

```
Verbose
```

Parameters

There are no parameters for this subcommand.

Examples

verbose
Message IDs are not displayed when running in OS/390 UNIX
Command:
lpw
Local directory name set to hierarchical file /u/user33
Command:
verbose
EZA2859I Message IDs are displayed when running in OS/390 UNIX
EZA1460I Command:
lpw
EZA2578I Local directory name set to hierarchical file /u/user33
EZA1460I Command:

Usage

Verbose is normally toggled off when the client starts. If you want it toggled on when the client starts, use the -v parameter on the FTP command. If the FTP client is running in a TSO environment, the display of message IDs is controlled by the profile options MSGID and NOMSGID.
Chapter 6. Sending electronic mail using SMTP commands

This chapter describes how to use the SMTPNOTE command, provided with z/OS CS, to prepare and send electronic mail. Recipients of the mail can be users on your local host, users on network job entry (NJE), or users on TCP hosts. The SMTPNOTE command uses the Simple Mail Transfer Protocol (SMTP) to send the mail.

This chapter describes information about:
- Interfaces to the SMTP address space
- "Using the SMTPNOTE command from your terminal" on page 286
- "Monitoring the status of SMTP using the SMSG command" on page 292
- "SMTP commands" on page 298
- "SMTP responses" on page 314
- "Using batch SMTP command in TSO utilities" on page 315
- "SMTP with DBCS support" on page 316

Interfaces to the SMTP address space

Interfaces to the SMTP address space are:
- SMTP mail can be sent and received interactively over a TCP/IP network. Mail from TCP/IP network sites destined for local MVS users (or users on a NJE network attached to the local MVS system) arrives over this interface. All commands and data received and transmitted through this interface use ASCII characters.
- Interface from the Job Entry Subsystem (JES) spool, including any connected NJE nodes. SMTP commands can be written into a SYSOUT data set, with an external writer name of the SMTP address space. SMTP processes each of the commands in the data set in sequence, exactly as if it had been transmitted over a TCP/IP connection. This is how mail is sent from local MVS users to recipients on the TCP network. Batch SMTP data sets must contain commands and data in EBCDIC characters.

For a description of batch SMTP in TSO utilities, see "Using batch SMTP command in TSO utilities" on page 315. For examples of batch SMTP, see "Batch SMTP examples" on page 315.

Notes:
1. This interface can be used to send mail only. Using the VERB command with this interface causes spool problems. For a description of the VERB command, refer to "VERB command—Enable or disable verbose mode" on page 312.
2. With this interface, you might not be able to use certain SMTP options that require the NETDATA format. For details, refer to the z/OS Communications Server: IP Configuration Reference.

IBM z/OS CS TCP/IP provides an SMTP gateway function, which can be used to transfer electronic mail between an NJE network and a TCP/IP network, as shown in Figure 10 on page 286.
Using the SMTPNOTE command from your terminal

The SMTPNOTE command enables you to prepare the mail using the facilities of the Time Sharing Option (TSO) EDIT command, or to send mail prepared with a system editor of your choice.
SMTPNOTE command—Send electronic mail to one or more recipients on NJE or TCP networks

Purpose
Use the SMTPNOTE command to send electronic mail to one or more recipients on NJE or TCP networks.

Format

```
>SMTPNOTE
  To  (--recipient--)  Cc  (--recipient--)
    NOCc

Subject  (--subject--)
  Dataset  (--data_set_name--)
    Batch
    Reuse

>SMTPNOTE
```

Note: The minimum abbreviation for each parameter is shown in uppercase letters.

Parameters

**To** *(recipient)*
Specifies a single recipient for the mail. If you do not specify the To parameter, you are prompted to enter a list of recipients. Enter the name of each recipient on a separate line and end the list with a blank line. You must specify at least one mail recipient.

**Cc** *(recipient)*
Specifies a single copy recipient for the mail. If you do not specify the Cc parameter and you do not specify the NOCc or Batch parameters, you are prompted to enter a list of copy recipients.

If there are no recipients, press Enter. Otherwise, enter the name of each recipient on a separate line and end the list with a blank line.

**NOCc**
Specifies that no prompting for Cc takes place.

**recipient**
Specifies the path address of the mail recipient. The format of *recipient* is equivalent to the path syntax, as described in RFC 821, without the (<) and (>) delimiters. See Appendix F, “Related protocol specifications (RFCs)” on page 409 for information about obtaining RFCs. This *recipient* parameter has one of the following formats:

- `user_id@host_name`
  User on a host in your domain, possibly on your local node.

- `user_id@host_name .domain`
  User on a host in a specified domain.

- `user_id%nje_host_name@ gateway_name .domain`
  User on an NJE or RSCS node connected to a TCP network at gateway_name. See “Electronic mail gateway on page 297” for more information.

- `@host1,@host2,...,@hostn :user_id @host_name`
  User on a host that is not known by the local Domain Name Server, but that can be reached by following the path from host1 to hostn.
Subject *(subject)*
Specifies the subject of the note. If you do not specify the Subject parameter, you are prompted for the subject.

If there is no subject for the note, press Enter. If Subject is specified as a keyword, you cannot have embedded blanks. Otherwise, the subject can be any arbitrary string of characters, but is limited to a total length of 233 characters.

Batch
Specifies that no prompting take place. You must also specify the To and Dataset parameters.

Dataset *(data_set_name)*
Indicates that the text of the mail is contained in a sequential data set. The data set can have any record format, be blocked or unblocked, and have records of up to 243 characters in length. The *data_set_name* is the name of the sequential data set containing the text of the note. It must be a valid data set name, and is fully qualified if it is contained within single quotation marks (*'*).

Reuse
Causes SMTP to reuse the contents of a note that was previously canceled. If a note was not canceled, the Reuse parameter is ignored.

Usage
- When the To(), Cc(), Subject(), or Dataset() parameters are specified more than once, the last value specified is the one used by SMTPNOTE. No error messages are generated for duplicate parameters on the command line.
- SMTPNOTE no longer requires quotation marks around blanks, single quotation marks, semicolons, or commas, or triple quotation marks around data set names.
Preparing and sending mail

After you enter the SMTPNOTE command, you are prompted for the mail recipients (To), the copy recipients (Cc:), and the subject of the note (Subject:), if they were not specified with the SMTPNOTE command. If you enter a list of recipients, enter the name of each recipient on a separate line and indicate the end of the list by entering a blank line.

After you answer the prompts, SMTPNOTE invokes the TSO EDIT command to enable you to prepare your note. Typically, the editor starts in INPUT mode. Enter the text of your note line by line. When your note is complete, enter a null line (that is, do not type anything when prompted), and press Enter. The editor switches to EDIT mode.

In EDIT mode, you can use all the functions of the editor. You can also return to INPUT mode, send the note, or cancel the note. For a complete description of the EDIT command, refer to the *IBM TSO Extensions Command Language Reference*.

If you invoke the SMTPNOTE command with the REUSE or DATASET parameter, you are immediately placed in EDIT mode. The contents of the previously canceled note, or the data set that you specify, are already part of the note. You can add to or change the data that is already present.

To send the note, enter **END SAVE** in EDIT mode, and then enter **SEND**. To cancel the note, enter **END SAVE** or **END NOSAVE** in EDIT mode, and then enter **CANCEL**. If you cancel a note, you can recover what you entered by invoking the SMTPNOTE command with the REUSE parameter. The recipients and subject of the note are not saved, and must be reentered.

Figure 11 is an example of preparing and sending mail.

```
READY
smtpnote
TO:
irvine@mvs2.accounting
bekker@mvs2.accounting
mcgregor@mvs1.accounting

CC:

SUBJECT:
Travel Expenses

ENTER "END SAVE" TO SAVE THE NOTE.
FOR A COMPLETE LIST OF EDIT SUBCOMMANDS ENTER "HELP".
INPUT
Could we please postpone the expense review because I will be out of town the week of the 19th.

How about the 23rd? Thank you, John.

EDIT
end save
ENTER "SEND" TO SEND THE NOTE.
ENTER "CANCEL" TO TERMINATE WITHOUT SENDING THE NOTE.

send
READY
```

Figure 11. Example of preparing and sending mail
Receiving mail
Use the TSO RECEIVE command to receive SMTP mail, as you would other mail and messages. Figure 12 shows an example of using the RECEIVE command.

Figure 12. TSO RECEIVE command

If there is no message to be received, the following message is displayed:

INMR003I You have no messages or data sets to receive.
***

For more information, refer to the IBM TSO Extensions Interactive Data Transmission Facility: User’s Guide.

Undelivered notes
When SMTP cannot deliver a piece of mail, a nondelivery note explaining the reason for nondelivery is sent to the sender. Nondelivery can occur for several reasons, such as the destination host is unreachable, or the recipient does not have a user ID on the destination host. If a note cannot be delivered, the body of the original piece of mail is returned as part of the nondelivery notification. For an example of a nondelivery note, see Figure 13 on page 291. The nondelivery note is shown in bold.
If a recipient is unknown at the destination host, the destination host does not accept the mail for delivery, and a nondelivery note is forwarded to the sender. For an example of an unknown recipient note, see Figure 14. The unknown recipient note is shown in bold.

**SMTP exit for unwanted mail**

You can use an SMTP exit to inspect and filter unwanted mail (spam) sent through SMTPPROC.

The SMTP component contains the SMTPPROC application which is the mail to mail interface used on the Internet. This component is responsible for routing mail between the sender and receiver. A user exit may be used to reject mail (spam) based on criteria determined by the user. This exit examines the data sent by the remote SMTP client application and determines what action to take based on the contents of that mail item or mail command.
This exit can be replaced dynamically without stopping the SMTPPROC program. The procedure for doing this follows:

1. Issue a "SMSG smtpprocnamet STOPEXIT" TSO command. The TSO userid must be in the authorized list for SMTPPROC to issue this command. This will cause SMTP to issue the termination call to the exit and then set a flag so that the exit will not be called anymore. Processing of mail will continue as if there is no exit.

2. Remove the old exit using the SETPROG EXIT operator command or by updating SYS1.PARMLIB(PROGxx) and issuing the refresh console command. An example of updating SYS1.PARMLIB follows:
   a. In SYS1.PARMLIB(PROGxx) have this line:
      ```
      EXIT DELETE EXITNAME(EZBTCPIPSMTPEXIT) MODNAME(OLDEXIT) FORCE(YES)
      ```
   b. At the MVS console issue `SET PROG=xx`.

3. Add the new exit using the SETPROG EXIT operator command or by updating SYS1.PARMLIB(PROGxx). An example of updating SYS1.PARMLIB follows:
   a. In SYS1.PARMLIB(PROGxx) have this line:
      ```
      EXIT ADD EXITNAME(EZBTCPIPSMTPEXIT) MODNAME(NEWEXIT)
      ```
   b. At the MVS console issue `SET PROG=xx`.

4. Issue a "SMSG smtpprocnamet STARTEXIT" TSO command. This will cause SMTP to issue the initialization call to the exit. A flag is then set so the exit will be called from then on for new mail connections. Processing of new mail will continue with the exit being called. The first smtp command to be seen by a reinstated exit will be HELO. The exit will not be called in the middle of a currently processing exchange.

### Monitoring the status of SMTP using the SMSG command

The SMSG command provides an interactive interface to the SMTP server to:

- Query the operating statistics of the SMTP server
- Query the SMTP mail delivery queues
- Perform privileged system administration tasks, such as shutting down the SMTP server and enabling or disabling various tracing and debugging options

Responses to commands are sent back to the originator of the command using MSG commands (or MSGNOH commands if SMTP is running with privilege class B).

SMTP SMSG is not supported in batch. It uses VMCF and the PASCAL interface to queue up information and will not print it to a DD card.
**SMSG SMTP command for the general user**

**Purpose**
As a general user, use the SMSG SMTP command to monitor the status of SMTP.

**Format**

```
$SMSG SMTP HEIp
 $SMSG SMTP QUeues
 $SMSG SMTP STats
```

**Note:** The minimum abbreviation for each parameter is shown in uppercase letters.

**Parameters**

**HElp**
Provides a list of valid SMTP SMSG commands.

**QUeues**
Provides a list of mail queued on the various SMTP mail delivery queues.

**STats**
Provides operating statistics about the SMTP server since the SMTP server was started.

**Examples**
The following example shows the output from the SMTP QUEUES command.

- **R** Retry
- **A** Active
- **Q** Pending - waiting for TCP/IP connection
  - `xxx.xxx.xxx.xxx` is the IP address
  - `1` is the number of pieces of mail
  - `HostName.DomainName` is the symbolic name

```
Msg from SMTP: * ----- Mail Queues ----- 
Msg from SMTP: * Spool Queue: 0 
Msg from SMTP: * R: xxx.xxx.xxx.xxx : 1 HostName.DomainName 
Msg from SMTP: * Undeliverable Queue: 0 
Msg from SMTP: * --- Resolver Queues --- 
Msg from SMTP: * Process Queue: 0 
Msg from SMTP: * Send Queue: 0 
Msg from SMTP: * Wait Queue: 0 
Msg from SMTP: * Retry Queue: 0 
Msg from SMTP: * Completed Queue: 0 
Msg from SMTP: * Error Queue: 0 
```

The following example shows the output from the SMTP STATS command.

Chapter 6. Sending electronic mail using SMTP commands  293
Usage
The statistics about SMTP include:

- The date that SMTP was last started.
- Statistics about mail handled by SMTP over the past two days, including:
  - The number of pieces of mail that arrived over TCP connections
  - The number of pieces of mail that arrived from spool (local or NJE senders)
  - The number of pieces of mail generated in response to requests to VERBose batch SMTP connections
  - The number of pieces of mail generated to return error mail to the sender
  - The number of pieces of mail delivered to local recipients
  - The number of pieces of mail delivered to recipients on the RSCS network
  - The number of pieces of mail delivered to recipients on the TCP/IP network
  - The number of TCP connections through which mail was received
  - The number of TCP connections through which mail was delivered
SMSG SMTP command for the privileged user

**Purpose**
As a privileged user, use the SMSG SMTP command to monitor the status of SMTP.

**Format**

```
> SMSG SMTP DEbug NODebug NODEbug NOTrace SHutdown STOPEXIT STARTEXIT TRace

```

**Note:** The minimum abbreviation for each parameter is shown in uppercase letters.

**Parameters**

**DEbug**
Enables connection debugging and tracing information to the SMTP.DEBUG data set. This is the same as adding DEBUG to the SMTP.CONFIG data set.

**EXpire IP_address**
Causes the domain name resolution for mail queued for delivery to this IP address to expire. SMTP will again try to resolve the IP addresses for this mail if it is still within the retry time window.

**NODebug**
Disables connection debugging and tracing.

**NOTrace**
Disables resolver tracing.

**SHutdown**
Causes the SMTP server to shut down.

**STARTEXIT**
Causes SMTP to enable a user exit program by issuing the initialization call to the user exit program if one exists. For more information regarding user exit programs refer to the [z/OS Communications Server: IP Configuration Guide](https://www.ibm.com/software/network/communications/zos/).

**STOPEXIT**
Causes SMTP to disable a currently running user exit program by issuing the termination call to the user exit program if one exists. For more information regarding user exit programs refer to the [z/OS Communications Server: IP Configuration Guide](https://www.ibm.com/software/network/communications/zos/).

**TRace**
Enables resolver tracing. The output of the resolver trace is sent to the SMTP console. This is the same as adding TRACE RESOLVER to the TCPIP.DATA data set.

**Examples**

**Note:** If the screen remains inactive even after you issue the SMSG SMTP command, press Enter a few times until any one of the following messages is displayed.
In this example, MVSUSER starts the debugging and tracing:

System: READY
User: smsg smtp debug
System: +Msg from SMTP: * Session Debugging Enabled

Stop debugging and tracing:

System: READY
User: smsg smtp nodebug
System: +Msg from SMTP: * Session Debugging Disabled
READY

Begin resolver tracing:

System: READY
User: msg smtp trace
System: +Msg from SMTP: * Resolver Tracing Enabled
READY

Stop resolver tracing:

System: READY
User: smsg smtp notrace
System: Msg from SMTP: * Resolver Tracing Disabled
READY

Shut down the SMTP server:

System: READY
User: smsg smtp shutdown
System: +Msg from SMTP: * ok, About to End SMTP
READY

Cause all mail queued for name resolution to expire:

System: READY
User: smsg smtp3 expire 9.67.112.25
System: +Msg from SMTP3: * No Mail Found for IP address: 9.67.112.25
READY

Usage
Privileged user SMSG commands are accepted only from users specified in the SMSGAUTHLIST statement.

Context
Refer to z/OS Communications Server: IP Configuration Reference for information about the TCPIP.DATA data set and for a description of the SMSGAUTHLIST configuration statement.
Electronic mail gateway

A system administrator can configure the SMTP server to run as a mail gateway between TCP network users and users located on an NJE or Remote Spooling Communications Subsystem (RSCS) network that is attached to the local host. Figure 15 is an example of a mail gateway.

In Figure 15, the following abbreviations are used:

A The local MVS host running both the TCP/IP program and NJE.
B and C Hosts attached to host A through an NJE network.
D and E Hosts attached to host A through a TCP network.

![Figure 15. SMTP as a mail gateway](image)

Users on hosts A, B, and C can send mail to users on TCP hosts D and E using the SMTPNOTE command. This method is described in "Using the SMTPNOTE command from your terminal" on page 286.

Users on TCP hosts D and E can send mail to the users on host A using addresses in the following format:

```
user_id@A.domain
```

Where:

- `user_id` is the user ID of the TSO user on host A.
- `A.domain` is the TCP host name of host A.

The users on TCP hosts D and E can send mail to users on NJE hosts B and C using addresses in the following format:

```
user_id%NJEHost@A.domain
```

Where:

- `user_id` is the user ID of the user on the host.
- `NJEHost` is the name of the NJE host (B or C).
- `A.domain` is the TCP host name of host A.

Path address

Path addresses are rewritten according to the following rules:

- If the local part of a mailbox name includes a percent sign (%) and the domain of the mailbox is the host system, SMTP rewrites the address, treating the portion of the local part to the right of the percent sign (%) as the real destination host. For example, the path address:
  
  John%yourhost@ourhost.our.edu

  is rewritten by SMTP running at ourhost.our.edu as:
Path addresses with source routes are accepted and rewritten to remove the domain name of the host system. For example, the path address:
@ourhost.our.edu,@next.host.edu
:John@yourhost

is rewritten by SMTP running at ourhost.our.edu as:
@next.host.edu:John@yourhost

SMTP also optimizes a path address. For example,
@some.host.edu,@ourhost.our.edu,
@next.host.edu:John@yourhost

is rewritten by SMTP running at ourhost.our.edu as:
@next.host.edu:John@yourhost

### SMTP commands

This section describes the SMTP commands as listed in Table 20.

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Define the following information as data.</td>
<td>299</td>
</tr>
<tr>
<td>EXPN</td>
<td>Verify if a mailbox exists on the local host.</td>
<td>301</td>
</tr>
<tr>
<td>HELO</td>
<td>Identify the domain name of the sending host to SMTP.</td>
<td>302</td>
</tr>
<tr>
<td>HELP</td>
<td>Get help with SMTP commands.</td>
<td>303</td>
</tr>
<tr>
<td>MAIL FROM</td>
<td>Specify the sender of the mail.</td>
<td>304</td>
</tr>
<tr>
<td>NOOP</td>
<td>Return a 250 OK return code when SMTP is responding.</td>
<td>305</td>
</tr>
<tr>
<td>QUEU</td>
<td>Get information about mail queued at SMTP for delivery.</td>
<td>306</td>
</tr>
<tr>
<td>QUIT</td>
<td>End an SMTP connection.</td>
<td>308</td>
</tr>
<tr>
<td>RCPT TO</td>
<td>Specify the recipients of the mail.</td>
<td>309</td>
</tr>
<tr>
<td>RSET</td>
<td>Reset the SMTP connection to the initial state.</td>
<td>310</td>
</tr>
<tr>
<td>TICK</td>
<td>Insert an identifier into the batch SMTP response data set.</td>
<td>311</td>
</tr>
<tr>
<td>VERB</td>
<td>Enable or disable verbose mode. <strong>Note:</strong> VERB ON may cause spool problems for SMTP if the REPLY TO: user is not a valid NJE node.user.</td>
<td>312</td>
</tr>
<tr>
<td>VRFY</td>
<td>Verify if a mailbox exists on the local host.</td>
<td>313</td>
</tr>
</tbody>
</table>

Data sets containing SMTP commands can be written to the JES spool as SYSOUT data sets. These SYSOUT data sets contain either punch or NETDATA records. Data sets originate from users on the same system as the SMTP address space or from users on any system connected to the host system through an NJE network.
DATA command—Define the following information as data

Purpose
Use the DATA command to define the following information as the data text of the mail body.

Format

```
DATA
```

Parameters
There are no parameters for this command.

Examples
After entering the DATA command, you receive the following message (response code 354) when you can transmit the body of your mail:

```
354 Enter mail body. End new line with just a '.'
```

You end transmitting the body of your mail by entering a single ASCII period (.) on a line by itself.

The following message indicates success:

```
250 Mail Delivered
```

Usage

- Use the DATA command after a HELO command, a MAIL FROM command, and at least one RCPT TO command have been accepted.
- When receiving mail over a TCP connection, the ASCII period should be followed by the ASCII <carriage return> <line feed> character sequence. If any record in the body of the mail begins with a period, the sending SMTP program must convert the period into a pair of periods (··). When the receiving SMTP encounters a record that begins with two periods in the body of the mail, it discards the leading period. This convention permits the body of mail to contain records that would otherwise be interpreted as signaling the end of the body of mail. These rules must be followed over both TCP and batch SMTP connections.
- The SMTPNOTE command performs this period doubling on all mail spooled to SMTP. If the body of the mail in a batch SMTP command is not explicitly ended by a record with a single period, SMTP adds one. After a period has been received, the SMTP connection is reset to the initial state (the state before any sender or recipients have been specified). Additional MAIL FROM, RCPT TO, DATA, and other commands can now be sent.
- If no more mail is to be delivered, end the connection with the QUIT command. If a QUIT command is not found at the end of a batch SMTP command data set, it is implied.
- If SMTP runs out of local mail storage space, it sends a reply with a 451 code. If the length of the body of the mail exceeds MAXMAILBYTES (defined in SMTP configuration data sets to be 512KB), SMTP sends a reply with a 552 code. For more information about MAXMAILBYTES, refer to the z/OS Communications Server: IP Configuration Reference.
- When mail arrives over a batch SMTP connection from a Remote Spooling Communications Subsystem (RSCS) network host, and 822 header rewriting is enabled (with the REWRITE822HEADER configuration option), then header fields
are modified to ensure that all addresses are fully qualified domain names. Refer to the [z/OS Communications Server: IP Configuration Reference](#) for more information about header rewriting.

- When mail arrives over the JES spool interface using SMTP batch or TSO transmit command, an NL (newline) or < character occurring in column 80 of the data is interpreted as a continuation character by the SMTP transport layer. The last byte of data should not be a continuation character. If the last byte is a continuation character, then the last record will not be processed correctly and you will be missing data. Check your note file, correct the data and resend the mail.
EXPN command—Verify whether a mailbox exists on the local host

**Purpose**
Use the EXPN command to verify whether a given mailbox exists on the local host.

**Format**

```
> EXPN mailbox
```

**Parameters**

*mailbox*

Specifies a user-defined identifier for a mailbox. This name can specify a single mailbox or a mailing list.

**Examples**

Verify whether mike is a mailbox on host abc.com:

```
System: READY
User:  EXPN mike
System: 250 mike@abc.com
```

Verify whether users-hackers is a mailing list on host abc.com:

```
System: READY
User:  EXPN users-hackers
System: 250-carol@abc.com
250-greg@abc.com
250-marsha@abc.com
250-peter@abc.com
```

The hyphen (-) as the fourth character of a response indicates that the response is continued on the next line.

**Usage**

- The EXPN command operates exactly the same as the VRFY command.
- The EXPN command can verify the existence of one or more mailboxes on the system. The mailboxes are defined by configuration statements in the SMTP.SMTP.CONFIG data set.
- The MVS SMTP server verifies only TSO user IDs if RACF is installed on the local system. TSO user IDs that are verified as valid are accepted with a reply code of 250. If RACF is not installed, any character string up to eight characters is accepted with a reply code of 250.
HELO command—Identify the domain name of the sending host to SMTP

**Purpose**
Use the HELO command to identify the domain name of the sending host to SMTP before a MAIL FROM command.

**Format**
```
>HELO domain_name
```

**Parameters**
- `domain_name` Specifies the domain name of the sending host.

**Usage**
- The HELO command is sent once before a MAIL FROM command.
- If `domain_name` is not known, you can send messages, but the message never heard of you is returned. This message indicates that the HELO command is accepted, but that the host name could not be resolved either through the name server or through the host tables.
- HELO commands received over a TCP connection that has a `domain_name` IP address that does not match the IP address of the ForeignSocket field of the connection information record are accepted. However, the reply text indicates the mismatch. HELO commands received over a batch SMTP connection with a `domain_name` that does not match the host name of the origination point of the spool data set are accepted, but the reply text indicates the mismatch.
HELP command—Get help with SMTP commands

**Purpose**
The HELP command returns a multiline reply with some general information about the SMTP commands.

**Format**
```
HELP command_name
```

**Parameters**
- `command_name`
  Indicates any of the SMTP commands.
MAIL FROM command—Specify the sender of the mail

Purpose
Use the MAIL FROM command to specify the sender of the mail.

Format

```
MAIL FROM: <sender_path_address>
```

Parameters

`sender_path_address`

Specifies the full path address of the sender of the mail.

Note: The angle brackets (< >) surrounding the `sender_path_address` in the syntax diagram are required.

Usage

- The MAIL FROM command is used once after a HELO command.
- If the host system has never heard of the sender host, a positive reply is sent, but a message is logged in the SMTP console.
NOOP command—Return a 250 OK return code when SMTP is responding

**Purpose**
Use the NOOP command to return a 250 OK return code when SMTP is responding.

**Format**
```
NOOP
```

**Parameters**
There are no parameters for this command.
QUEU command—Get information about mail queued at SMTP for delivery

**Purpose**  
Use the QUEU command to get information about mail queued at SMTP for delivery.

**Format**

```
QUEU [DATE]
```

**Parameters**

- **DATE**
  Causes information about the age of the mail to be returned. The default is to not return age information.

**Usage**

QUEU returns a multiline reply with information about mail queued at SMTP for delivery. The following information is returned about mail:

- **Spool Queue**  
  Contains mail that is destined for recipients on the local MVS system, or for recipients on an NJE system attached to the local MVS system. This queue is generally empty, because SMTP can deliver this mail quickly by spooling it to the local recipient or to the NJE address space for delivery to an NJE network recipient.

- **Active**  
  Indicates that if SMTP is currently transmitting to a TCP network destination, all the mail queued for that destination is shown to be active.

- **Queued**  
  All mail that arrives over a batch SMTP connection, and mail from TCP connections that is to be forwarded to another TCP network destination through source routing, is placed on the queued list. As soon as SMTP receives resources from the TCPIP address space, mail that is queued is promoted to active.

- **Retry Queue**  
  Mail is placed here after SMTP tries to transmit mail to each of the TCP network hosts, but was unable to either open a connection or complete delivery over the connection. After the number of minutes specified by RETRYINT, mail is promoted from the retry queue to the QUEUED state. For more information about the RETRYINT variable, refer to the z/OS Communications Server: IP Configuration Reference.

- **Undeliverable Queue**  
  Mail is placed here if SMTP cannot deliver mail to a local MVS recipient, or to a recipient on the NJE network attached to the local MVS system, because spool space on the local MVS system is full. After spool space has been increased and SMTP has been restarted, delivery is attempted again.

- **Resolution Queues**  
  SMTP uses the following queues for processing queries to the name server. If the SMTP server is configured to use the site tables rather than the name server, these queues are not used.
If the queue is empty, the word Empty appears to the right of the queue. If the queue contains queries, the queries appear on separate lines below the queue. However, due to the speed of the SMTP server, the output may show that the queue is active without containing any entries. In this case, the word Empty does not appear.

**Resolver Process Queue**
Is generally empty; it contains queries waiting to be sent to the SMTP resolver. After the query has been processed, it is put in the resolver send queue.

**Resolver Send Queue**
Contains queries waiting to be processed by the SMTP resolver. SMTP staggers the number of queries sent by the resolver to prevent the overloading of the network and the name server.

**Resolver Wait Queue**
Contains queries for which the SMTP resolver is waiting for responses. Queries remain in this queue for the period of time it takes to receive a reply from the name server. If a reply is not received, the queries are removed from this queue after the resolver timeout has occurred, and are placed in the resolver retry queue. If the query is successful, the query is placed in the resolver completed queue.

**Note:** The SMTP resolver timeout is specified by the RESOLVERTIMEOUT statement in the TCPIP.DATA data set.

**Resolver Retry Queue**
Contains queries that have previously failed, either because the name server did not reply, or the name server returned a temporary error that forced the SMTP resolver to retry the query. A temporary error occurs if, for example, the name server truncates a packet, or if the name server detects a processing error.

The RESOLVERRETRYINT statement specifies the number of minutes SMTP waits before retrying the query. The RETRYAGE statement specifies the number of days SMTP should continue to resolve the query before returning the mail to the sender.

**Resolver Completed Queue**
Contains queries that have been resolved and are waiting to be recorded into the mail. After the Internet addresses are recorded, SMTP attempts to deliver the mail.

**Resolver Error Pending Queue**
Contains queries that the name server has returned without answers. The corresponding mail message is returned to the sender with an unknown recipient error.
QUIT command—End an SMTP connection

**Purpose**
Use the QUIT command to end an SMTP connection.

**Format**
```
QUIT
```

**Parameters**
There are no parameters for this command.
RCPT TO command—Specify the recipients of the mail

Purpose
Use the RCPT TO command to specify mail recipients.

Format

```plaintext
RCPT TO: <recipient_path_address>
```

Parameters

- `recipient_path_address`
  Specifies the full path address of the mail recipient.

Note: The angle brackets (< >) surrounding `recipient_path_address` in the syntax diagram are required.

Usage

- You can use the RCPT TO command a maximum of 3000 times in a single SMTP job. If you use more than 3000 RCPT commands, the excess commands will receive a failure reply code **552 Too many recipients**. The mail will not be delivered to the excess recipients.
- You can use the RCPT TO command only after a MAIL FROM command has been issued.
- If the host system has never heard of the recipient host, the RCPT TO command receives a negative reply.
- If a name server is used for domain name resolution, MX records are used to resolve the recipient IP address before trying A records.

Context

For more information about MX and A records, refer to the *z/OS Communications Server: IP Configuration Guide*.
RSET command—Reset the SMTP connection to the initial state

Purpose
Use the RSET command to reset the SMTP connection to the initial state in which the sender and recipient buffers are erased and the connection is ready to begin a new mail transaction.

Format

```
RSET
```

Parameters
There are no parameters for this command.
TICK command—Insert an identifier into the batch SMTP response data set

**Purpose**
Use the TICK command, in combination with the VERB ON command, to insert an identifier into the batch SMTP response data set. This command is useful with mail systems that keep track of batch SMTP response data sets.

**Format**

```
TICK identifier
```

**Parameters**

`identifier`

Specifies a string used to identify the origin of batch SMTP responses.

**Usage**

The TICK command has no effect when it is issued over a TCP connection to SMTP.
VERB command—Enable or disable verbose mode

Purpose
Use the VERB command to enable or disable verbose mode.

Format
>

Parameters
ON
Enables verbose mode. When enabled, the batch SMTP commands and associated replies are recorded in the batch SMTP response data set. Also, the batch SMTP response data set is sent back to the origination point of the batch SMTP command data set.

To avoid receiving spool errors, ensure that the origination point is a valid JES user and node on the SMTP sending system. The origination point information is taken from the TSO transmit (XMIT) command headers.

OFF
Disables verbose mode. When disabled, only the replies (not the commands) are recorded in the batch SMTP response data set. This is the default.

Usage
The VERB command has no effect when issued over a TCP connection to SMTP.
VRFY command—Verify whether a mailbox exists on the local host

Purpose
Use the VRFY command to verify whether a given mailbox exists on the local host.

Format

VRFY mailbox

Parameters

mailbox
Specifies a user-defined identifier for a mailbox. This name can specify a single mailbox or a mailing list.

Examples

Verify whether mike is a mailbox on host abc.com:

System: READY
User: VRFY mike
250 mike@abc.com

Verify whether users-hackers is a mailing list on host abc.com:

System: READY
User: VRFY users-hackers
250-carol@abc.com
250-greg@abc.com
250-marsha@abc.com
250-peter@abc.com

The hyphen (-) as the fourth character of a response indicates that the response is continued on the next line.

Usage

- The VRFY command operates exactly the same as the EXPN command.
- The VRFY command can verify the existence of one or more mailboxes on the system. The mailboxes are defined by configuration statements in the SMTP.SMTP.CONFIG data set.
- The MVS SMTP server verifies only TSO user IDs if RACF is installed on the local system. TSO user IDs that are verified as valid are accepted with a reply code of 250. If RACF is not installed, any character string up to eight characters is accepted with a reply code of 250.
SMTP responses

SMTP commands arrive over a TCP connection (to your terminal) or over a batch SMTP connection. With either connection, a response to each command is generated. All responses are prefixed with a 3-digit number. You can determine the response by inspecting the first digit of the response code:

<table>
<thead>
<tr>
<th>First digit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Indicates a positive response. Command accepted.</td>
</tr>
<tr>
<td>3</td>
<td>Indicates a positive response. Send the data associated with the command.</td>
</tr>
<tr>
<td>4</td>
<td>Indicates a temporary negative response. Try again later.</td>
</tr>
<tr>
<td>5</td>
<td>Indicates a permanent negative response. The command has been rejected.</td>
</tr>
</tbody>
</table>

If SMTP commands arrive over a TCP connection, all responses (positive or negative) are returned over that TCP connection. If SMTP commands arrive over a batch SMTP connection, all responses are written to the batch SMTP response data set.

If verbose mode is enabled for a batch SMTP connection, SMTP returns the batch SMTP response data set to the origination point of the spool data set. The origination point is determined from the NETDATA header if the data set arrives in NETDATA format, or the MAIL FROM command if the data set arrives in punch format. If the batch SMTP connection is not in verbose mode, the batch SMTP response data set is not returned to the point of origin, and is discarded.

If an error occurs during the processing of commands over a batch SMTP connection, such as reception of a negative response (with a first digit of 4 or 5), an error report is mailed back to the sender. The sender is determined from the last MAIL FROM command received that was valid. If the sender cannot be determined from a MAIL FROM command, the sender is assumed to be the origination point of the batch SMTP command data set. The error report mailed to the sender includes the batch SMTP response data set and the text of the undeliverable mail.

All SMTP commands and data that arrive over TCP or batch SMTP connections are subject to the restrictions imposed by constants defined in the SMTPGLOB macro. Changes to the SMTPGLOB macro require recompiling the SMTP server. The default values of these constants are:

- Command lines must not exceed MaxCommandLine (512 characters).
- Data lines longer than MaxDataLine (1024 characters) are wrapped.
- Path addresses must not exceed MaxPathLength (256 characters).
- Domain names must not exceed MaxDomainName (256 characters).
- User names, the local part of a mailbox specification, must not exceed MaxUserName (256 characters).

A limit on the number of RCPTs in a single SMTP job is 3 000. If you have more than 3 000 the excessive RCPT commands will receive an failure reply code 552 Too many recipients. Abend B37 can still occur for either of the following reasons:

- No more space is available on volume.
- Volume table of contents (VTOC) is full.
Batch SMTP examples

The following sections contain examples that demonstrate batch SMTP capabilities.

Sending mail to a TCP network recipient
The following is an example of sending mail from an NJE network host to two TCP network recipients. The NJE network is BITNET, and the NJE host is named YOURMVS. This batch SMTP data set is spooled to SMTP at OURMVS, which is running with the run-time arguments GATEWAY and NJEDOMAIN BITNET. OURMVS is a BITNET host and is also connected through a TCP network to the hosts rsch.our.edu and ai.our.edu.

HELO YOURMVS
MAIL FROM:<CAROL@YOURMVS>
RCPT TO:<msgs@rsch.our.edu>
RCPT TO:<alice@ai.our.edu>
DATA
Date: Thur, 26 Mar 92 21:48:57 EST
From: Carol <CAROL@YOURMVS>
To: <msgs@rsch.our.edu>
Cc: <alice@ai.our.edu>
Subject: update

Mike: Cindy stubbed her toe. Bobby went to baseball camp. Marsha made the cheerleading team. Jan got glasses. Peter has an identity crisis. Greg made dates with 3 girls and couldn't remember their names.

QUIT

SMTP rewrites the From: line to reflect that the mail has been transferred from an NJE network (in this case, BITNET) to a TCP network. The TCP network recipients receive:

From: carol <CAROL@YOURMVS.BITNET@ourhost.our.edu>

Querying the SMTP delivery queues
The SMTP delivery queues can be queried by sending a data set with the SMTP commands VERB ON and QUEU to the SMTP address space. A batch SMTP response data set is returned with the result of the VERB ON command.

Using batch SMTP command in TSO utilities

The batch SMTP commands are also used with the TSO transmit (XMIT) command and the IEBGENER utility in TSO/ISPF.

Use the TSO transmit (XMIT) command to spool a batch SMTP command sequential data set to SMTP. After you create the batch SMTP command sequential data set, use the following command to spool the data set to SMTP:

XMIT jesnode-name.smtp DA(batsmtp.text)

Where:
jesnode
Is the JES nodename or hostname.
smtp
Is the SMTP address space name. This is the default.
batsmtp.text
Is the batch SMTP commands data set.
Note: If sending a member of a partitioned data set (pds) via the TSO transmit command, the member must be converted to a SEQuential file. Use the SEQ option on the TSO transmit (XMIT) command to accomplish this as shown in the following example.

```
XMIT jesnod-name.smtp DA(pds(member)) SEQ
```

Where *pds* is the name of the partitioned data set and *member* is the name of the member to be sent to SMTP.

To code the batch SMTP commands as inline input for SYSUT1 and SYSUT2, create the following JCL using the IEBGENER utility on the TSO/ISPF application.

```
//BATSMTP JOB (userid,nn),MSGCLASS=B,PRTY=12,MSGLEVEL=(2,1)
//IEBGENER EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSUT1 DD *
HELO YOURMVS
MAIL FROM:<CAROL@YOURMVS>
RCPT TO:<msgs@rsch.our.edu>
RCPT TO:<alice@ai.our.edu>
DATA
Date: Thur, 26 Mar 92 21:48:57 EST
From: Carol <CAROL@YOURMVS>
To: <msgs@rsch.your.edu>
Cc: <alice@ai.your.edu> Subject: update

Mike: Cindy stubbed her toe. Bobby went to
  baseball camp. Marsha made the cheerleading team.
  Jan got glasses. Peter has an identity crisis.
  Greg made dates with 3 girls and couldn’t
  remember their names.

QUIT
/*
//SYSUT2 DD SYSOUT=(B,smtp)
//*   SMTP address space name for external writer
//*   SYSOUT class
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//
```

SMTP with DBCS support

Mail in EBCDIC DBCS Big-5, Kanji, Hangeul, Simplified Chinese, or Traditional Chinese can be sent to a remote host using SMTPNOTE, if the SMTP server is configured with the corresponding DBCS support. Refer to the Z/OS Communications Server: IP Configuration Reference for more information about configuring DBCS support for the SMTP server.

Conversion of DBCS mail

The transmission of DBCS mail by SMTP actually uses two different translation tables, one SBCS and one DBCS. SBCS characters in the mail headers and in the mail body are converted using *hlq.STANDARD.TCPKJBIN, TCPHGBIN, TCPSCBIN,* or TCPCHBIN.

Both SBCS and DBCS translation tables are required for the transmission of DBCS mail, as the mail can contain mixed-mode strings. A mixed-mode string contains both SBCS and DBCS data, delimited by shift-out and shift-in characters.
DBCS conversion is performed only on outgoing and incoming mail to and from other hosts. Mail spooled to SMTP by SMTPNOTE for the local host is delivered directly, without any DBCS code conversion.
Chapter 7. Sending electronic mail using z/OS UNIX sendmail

This chapter briefly describes how to use z/OS UNIX sendmail, provided with z/OS CS, to prepare and send electronic mail using the facilities of the z/OS shell.


You can also find more information about sendmail on Web site [http://www.sendmail.org](http://www.sendmail.org).
sendmail command—Send file contents

Purpose

Use the sendmail command to send the contents of a file.

Format

```
sendmail user_name command_line_switch
```

Parameters

- **user_name**: The user name of the person you want to receive the mail.
- **command_line_switch**: Refer to the `z/OS Communications Server: IP System Administrator's Commands` for information about sendmail command line arguments or switches.

Examples

To test run sendmail by hand, complete the following steps:

1. Create a file named `sendstuff` with the following contents:
   ```
   This is a one line message.
   ```
2. Mail this file to yourself with the following command line, where `you` is your login name:
   ```
   /usr/sbin/sendmail you <sendstuff
   ```

You run sendmail directly by specifying its full path name. When you run sendmail, any command-line arguments that do not begin with a hyphen (-) are considered to be the names of the people to whom you are sending the mail message.

**Note**: The full path name may differ on your system. If so, simply specify the full sendmail path name as it is configured on your system.

The `<sendstuff` sequence causes the contents of the file that you have created (`sendstuff`) to be redirected into the sendmail program. The sendmail program treats everything it reads from its standard input (up to the end of the file) as the mail message to transmit.

Results

There are several ways to view the message that you just sent, depending on how your system is configured. You can type MAIL to view your mail, or use the mh package and type INC to receive and SHOW to view your mail. After viewing your mail, save the mail message to a file. The file will look something like this:
This file begins with eight lines of text that were not in your original message. Those lines were added by sendmail and your local delivery program and are called the header.

The last line of the file is the original line from your sendstuff file. It is separated from the header by one blank line. The body of a mail message comes after the header and consists of everything that follows the first blank line.

Ordinarily, when you send mail with your MAILUSERAGENT (MUA), the MUA adds a header and feeds both the header and the body to sendmail. This time, however, you ran sendmail directly and supplied only a body; the header was added by sendmail.
Chapter 8. Using remote printing

z/OS CS provides client and server support for remote printing. The remote printing application enables you to spool data sets remotely to a Line Printer Daemon (LPD). The Line Printer Requester (LPR) sends the spooled data set to a specified print server host and to a specified printer.

This chapter describes the remote printing commands listed in Table 21.

**Note:** Although this chapter describes the commands and parameters that are valid for the MVS LPR client, you may not get the same results from non-MVS servers, because those servers may not support the same commands and parameters.

**Table 21. Remote printing commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPQ</td>
<td>Request a list of the printer queue on a remote printer.</td>
<td>324</td>
</tr>
<tr>
<td>LPR</td>
<td>Print to a remote printer.</td>
<td>326</td>
</tr>
<tr>
<td>LPRM</td>
<td>Remove a job from the printer queue on a remote host.</td>
<td>339</td>
</tr>
<tr>
<td>LPRSET</td>
<td>Set the default printer and host name.</td>
<td>341</td>
</tr>
</tbody>
</table>
LPQ command—Request a list of the printer queue on a remote printer

Purpose

Use the LPQ command to request a list of the printer queue on a remote printer from the LPD server controlling that printer.

Notes:

1. Do not use the forward slash character (/) in any parameter value for this command.
2. The TSO LPQ command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

Format

LPQ

job_id

Opt Parms 1:

- ALL
- Printer name
- Host host
- AT host

Opt Parms 2:

- TRace
- Type
- Version

Parameters

job_id

Specifies either a user ID (this cannot start with a digit), or a job number in the remote printer queue. If you do not specify job_id with the LPQ command, all the jobs in the remote printer queue are listed.

Note: job_id is case-sensitive on some systems.

ALL

Prints a long report, which shows the source host and other print job information.

Printer name

Specifies the name of the printer for which you want the printer queue listed.

The printer name cannot contain an @ symbol.

Host host

Specifies the name or IPv4 IP address of the printer host. The name must resolve to an IPv4 address. AT is a synonym for this option.

AT host

Specifies the name or IPv4 IP address of the printer host. The name must resolve to an IPv4 address. Host is a synonym for this parameter.
**TRace**

Turns on the trace details for interaction with the remote printer. TRace always overrides TYpe because TYpe is a subset of TRace. Use this option whenever you need to document a problem.

**TYpe**

Displays the progress of the command.

**Version**

Displays the version of the program.

**Examples**

- Query the printer lp0 on the system os2sys1 and print a short listing of the jobs that are queued for the lp0 printer:

  \[ \text{LPQ (PRINTER lp0 HOST os2sys1)} \]

- If the LPRSET command was previously issued (LPRSET lp0@os2sys1), using the following LPQ command has the same effect as issuing the command in the previous example.

  \[ \text{LPQ} \]

- Get a long listing of the jobs queued, including the name of the host that created the jobs:

  \[ \text{LPQ (PRINTER lp0 HOST os2sys1 ALL)} \]

- List the jobs for a user named smith:

  \[ \text{LPQ smith (PRINTER lp0 HOST os2sys1)} \]

- Get information only about job 123:

  \[ \text{LPQ 123 (PRINTER lp0 HOST os2sys1)} \]

**Usage**

- If the printer or host name are not specified in the LPQ command, the last LASTING.GLOBALV variables for PRINTER and PRTHOST in the user_id.LASTING.GLOBALV data set are used as the defaults. You can specify these variables with the LPRSET command. You can use these variables to set up a default printer and host to be used if you do not specify a printer or host.

- User names in a query are case-sensitive. For example, smith and SMITH are not the same names. Also for example, on UNIX systems, lp0 and LP0 can refer to different printers.

- Some systems do not answer with the job information when you use a job number for a job that was not produced by the querying system.

- You would not normally use the LPQ command to query an MVS system, because the LPD queue on MVS processes so quickly.
LPR command—Print to a remote printer

Purpose

Use the LPR command to print to a remote printer.

Notes:
1. Do not use the forward slash character (/) in any parameter value for this command.
2. The TSO LPR command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

Format

```
LPR data_set_name
```

Optional parameters:

326 z/OS V1R4.0 CS: IP User’s Guide and Commands
More optional parameters:
filter:

- f
- 1
- p
- r

jobname:

- DEST
- FOR
- FORM
- IDENTIFIER
- LINECOUNT
- OTHERS
- PASS
- PRIORITY

Parameters

data_set_name
- Specifies the name of the data set to be printed. This cannot be the name of an HFS file.

AT host
- Specifies the name or IPv4 IP address of the printer host machine. If host is a name, it must resolve to an IPv4 address. Host is a synonym for this option.

BIG5
- Converts data from Traditional Chinese host DBCS to Big-5 DBCS when transferring data to a remote system. LPR loads the BIG5 DBCS translation table from TCPCHBIN binary table data set.
NOBlnary
Causes LPR to convert the data from EBCDIC to ASCII when it is sent to the remote system. This is the default.

BINary
Causes LPR to send the data without translation and without any indication of record boundaries. Use this option if the data set is already in ASCII.

Note: The MVS LPD server always converts data sets in ASCII to EBCDIC, and there is no option to disable this conversion.

BUrst
Causes a burst page to be printed on the remote printer. This is the default.

NOBUrst
Prevents a burst page from being printed on the remote printer.

CC
Causes the remote system to interpret the first character of each line as carriage control.
Records containing control characters that are not valid are deleted. If all records in the file are deleted, LPR processes the empty file.
CC is the default if the record format is FA, FBA, FBM, FM, VA, VBA, VBM, or VM. The characters used to specify these record formats have the following meanings:

- F Fixed record length
- V Variable record length
- B Blocked records
- A Records containing ISO/ANSI control characters
- M Records containing machine code control characters

NOCc
Prevents the remote system from interpreting the first character of each line as an ASA carriage control.

CFfirst
Specifies that LPR will send the ControlFile describing the data before it sends the DataFile that contains the data. This option enables some LPD servers to print larger jobs since the data can be printed as it is received.
Specify this option when small jobs print, but large jobs do not.

Note: Even when this option is specified, the print job may still be too complex to print.

CLass class
Specifies the class name to the remote system. The class name is printed on the banner pages. The default is the sending host name.
To override the SYSOUT CLASS of a job, the CLASS value must be only one alphanumeric character. For example, CLASS c.

COpies copies
Specifies the number of copies to be printed. The default is one copy.

EOFf
Causes an ASCII line feed after the last line of data of a file formatted by ANSI carriage control. Since ANSI carriage control positions the paper before the line is printed, you cannot specify how the paper should be positioned after the last
line prints. The normal network standard is to terminate every file with a line feed. Some LPDs do not print the last line if the line feed is not added. For example, in ANSI carriage control, a 'F1'X means eject to a new page before printing the lines. ANSI or ASA CC is used in files with RECFM=FBA.

EOFIf is the default for ANSI CC files. Specify NOEOIf if your paper is positioned incorrectly and an extra line is printed because LPR adds a line feed. This parameter does not impact files that are not ANSI CC. For example, BinNary, Postscript, LAndscape and Machine CC files are not affected by EOFIf.

**NOEOIf**

Inhibits adding the ASCII line feed after the last line of a file that is formatted by ANSI carriage control. Since ANSI carriage control positions the paper before the line is printed, how the paper should be positioned after the last line cannot be specified by the carriage control characters. The normal network standard is to terminate every file with a line feed. Specifying this parameter causes the last byte of data to be the last byte sent; a line feed is not added.

For example, an ANSI carriage control 'F1'X means eject to a new page before printing the line. ANSI or ASA CC is used, for example, in files with RECFM=FBA.

Specify NOEOIf if your paper is positioned wrong because an extra line is being printed at the end of the file. Some applications specify the positioning and do not have the extra line feed sent by LPR.

For example, BinNary, Postscript, and Machine CC files are not affected by this parameter. This option does not apply when you specify LAndscape.

**Euckanji**

Causes the data to be converted from EBCDIC DBCS to Extended UNIX Code kanji ASCII DBCS when it is sent to the remote system. LPR loads the EUCKANJI DBCS translation table from the TCPKJBIN binary translate table data set.

**Filter**

Specifies the type of processing to be done on the data by the remote system. The filter is written as a single letter. Both uppercase and lowercase letters are accepted, but uppercase letters are converted to lowercase.

**Note:** The filter values must also be defined in the SERVICE statement of the configuration data set for the MVS LPD server. Refer to the [z/OS Communications Server: IP Configuration Reference](https://www.ibm.com/support/knowledgecenter/SSECG2_1.4.0/com.ibm.zos.zos/doc/sg247997/sg247997.htm) for more information on the LPD server configuration data set.

In addition to the following list of filter codes, there are filters supported by other servers described in RFC 1179. [For a list of Requests for Comments (RFCs), see Appendix F, “Related protocol specifications (RFCs)” on page 409.]

<table>
<thead>
<tr>
<th>Filter code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>Print as a sequence of lines</td>
</tr>
<tr>
<td>l</td>
<td>Print, passing through all control characters</td>
</tr>
<tr>
<td>p</td>
<td>Print with pagination</td>
</tr>
<tr>
<td>Filter code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>r</td>
<td>Print, interpreting the first column as FORTRAN carriage control characters. The supported IBM FORTRAN carriage control characters are 1, 0, +, and −. When using the MVS LPD server to print files with filter code r, the PAGESIZE parameter on the SERVICE statement for the printer you are printing to also affects pagination. See “Usage” on page 337 for further information.</td>
</tr>
</tbody>
</table>

**HAngeul**
Causes the data to be converted from EBCDIC DBCS to Hangeul ASCII DBCS when it is sent to the remote system. LPR loads the HANGEUL DBCS translation table from the TCPHGBKIN binary translate table data set.

**HEader**
Causes a page header to be inserted by the client at the top of every printed page if the NOCc and NOBinary options are in effect. To cause the server to insert page headers, use p as the value of filter and specify the NOHeader option.

**NOHeader**
Prevents the client from inserting page headers.

**IBMkanji**
Causes the data to be sent without translation as IBM (EBCDIC) kanji. This parameter performs the same function as the BINary parameter.

**Indent number**
Specifies the number of columns the remote system indents the output when f or p is specified as the value of filter.

**JIS78kj ASCII**
Causes the data to be converted from EBCDIC DBCS to JIS 1978 kanji ASCII DBCS, using the ASCII shift-in escape sequence ESC(B, when it is sent to the remote system. LPR loads the JIS78KJ DBCS translation table from the TCPJGBKIN binary translate table data set.

**JIS78kj JISROMAN**
Causes the data to be converted from EBCDIC DBCS to JIS 1978 kanji ASCII DBCS, using the JISROMAN shift-in escape sequence ESC(J, when it is sent to the remote system. LPR loads the JIS78KJ DBCS translation table from the TCPJGBKIN binary translate table data set.

**JIS83kj ASCII**
Causes the data to be converted from EBCDIC DBCS to JIS 1983 kanji ASCII DBCS, using the ASCII shift-in escape sequence ESC(B, when it is sent to the remote system. LPR loads the JIS83KJ DBCS translation table from the TCPJGBKIN binary translate table data set.

**JIS83kj JISROMAN**
Causes the data to be converted from EBCDIC DBCS to JIS 1983 kanji ASCII DBCS, using the JISROMAN shift-in escape sequence ESC(J, when it is sent to the remote system. LPR loads the JIS83KJ DBCS translation table from the TCPJGBKIN binary translate table data set.

**JNum number**
Specifies a specific job number for the print request, where number is a unique, 3-digit number in the range 000–999. This job number is used by LPR to name the temporary data and control files, such as:
The JNum parameter is not valid from NPF; do not specify JNum in the NPF OPTIONS file.

If JNum is not specified for LPR, the three-digit number is randomly generated by LPR.

**Job**

Specifies the job name to the remote system. The default name is the full data set name. The job name is printed on the banner pages.

The following parameters are available with the Job parameter and are entered as `PARAMETER=value`. When you use these parameters, you must separate them from Job or `jobname` by a blank and from other options by a comma and no blank. For example,

Job **PASS**=password,**FOR**=userid

**DEST**

Sets the destination node. The default is the node on which the LPR client is running.

**FOR**

Specifies a user ID other than the sending user ID for which the output is to be spooled. The default is the sender’s ID.

**FORM**

Identifies the form on which the data is printed. This is the equivalent of the form-name subparameter in the MVS SYSOUT parameter on the DD card. Therefore, the form name specified can be 1 through 4 alphanumeric or national ($, #, @) characters.

**IDENTIFIER**

Sets the destination ID. The default is SYSTEM.

**LINECOUNT**

Specifies a numeric field indicating the number of lines on a page. This option overrides the PAGESIZE parameter of LPD.

**OTHERS**

Causes all subsequent options to be ignored. This option is ignored by the MVS LPD server.

**PASS**

Specifies the password. The default is no password, which causes the job to fail if the RACF option is specified for the service.

**PRIORITY**

Specifies the transmission priority. The default is 50.

**Ksc5601**

Causes the data to be converted from EBCDIC DBCS to Korean Standard Code KSC-5601 ASCII DBCS when it is sent to the remote system. LPR loads the KSC5601 DBCS translation table from the TCPHGBIN binary translate table data set.

**LAndscape**

Converts a non-PostScript data set to a PostScript data set for printing with print lines parallel to the long edge of the paper. If the remote printer can process PostScript output, the data set is printed in landscape format (rotated 90 degrees). Some nonprinting EBCDIC characters below X'3F' are changed to blanks.

**Note:** An ASCII cntl-Z X'1A' is sent after the data. If this dos-EOF character causes problems, use the LANDNOcz option instead.

**LANDNOcz/LNcz**

Converts a non-PostScript data set to a PostScript data set for printing with
print lines parallel to the long edge of the paper. If the remote printer can process PostScript output, the data set is printed in landscape format (rotated 90 degrees). Some nonprinting EBCDIC characters below X'3F' are changed to blanks.

**Note:** Normally, you should use the LAndscape option, unless the dos-EOF character (X'1A') sent after the file causes problems. LN and LNcz are abbreviations for LANDNOcz.

**LATEconn**
Causes LPR to process the input data file before making any TCPIP connection with the printer. When this option is not specified, the TCPIP connection is made before the data file processing begins. This option can be specified when the initial processing of very large files causes the connection with a printer to be dropped because of a timeout.

**Lineno count**
Determines the number of lines to be printed before a new heading is printed. This option is meaningful only for a data set that does not have the CC option specified either explicitly or by default.

The valid range for Linenocount is 0 to the Pascal integer MAX number 2 147 483 647. The default value is 55. To suppress printing a header before each new page, specify Linenocount 0. Specifying Linenocount 0 has the same effect as specifying the NOLineno count option.

**NOLineno count**
Prevents a header being printed before each new page. Specifying NOLineno count has the same effect as specifying Linenocount 0.

**Mail**
Causes mail to be sent to the user when the printing operation ends (for those servers that support this).

**NAme name**
Specifies the job information to be provided by the remote system in response to a query. Only the name or files portion of the query is displayed. This option is not honored by all remote printing servers.

**P0stscript**
Inserts the header required by some systems to recognize a PostScript data set.

**NOP0stscript**
Prevents a PostScript data set from being recognized as a PostScript data set.

**Printer name**
Specifies the name of the printer on which you want the data set printed. The printer name is case sensitive.

**Host host**
Specifies the name or IPv4 IP address of the printer host machine. If host is a name, it must resolve to an IPv4 address. AT is a synonym for this option.

**SChinese**
Converts data from Simplified Chinese host DBCS to Simplified Chinese PC DBCS when transferring data to a remote system. LPR loads the SCHINESE DBCS translation table from TCPSCBIN binary table data set.

**SJiskanji**
Causes the data to be converted from EBCDIC DBCS to Shift JIS kanji ASCII.
DBCS when it is sent to the remote system. LPR loads the SJISKANJI DBCS translation table from the TCPKJBIN binary translate table data set.

**Slowshutdown**
Causes slower TCP/IP connection termination after the job is sent to the printer. This option is rarely needed. This option is provided for print servers which discard the print job just successfully received when LPR uses the fast shutdown. Before using this option, check the print server for errors (such as spool file full) that may be causing the job to be discarded instead of printed.

**SOsi**
Determines how any EBCDIC DBCS shift-out (‘0E’X) and shift-in (‘0F’X) characters in the input file are handled. The ASCII, EBCDIC, or SPACE parameters specifies what is used as shift characters in the ASCII output from LPR. If SOsi is specified without a following parameter, ASCII is used as the default.

If you do not specify SOsi, shift-out/shift-in characters are not used in the ASCII data stream. Therefore, the EBCDIC DBCS shift characters are just removed during the translation to ASCII. This is the same as specifying SOsi NONE.

SOsi has no effect on DBCS translations JIS78KJ, JIS83KJ, and IBMKANJI. It is used with other DBCS translation such as BIG5, EUCKANJI, HANGEUL, KSC5601, SCHINESE, SJISKANJI, and TCHINESE.

**SOsi ASCII**
Specifies that DBCS data strings in the ASCII output are delimited by special shift-out/shift-in characters. As the data is translated from EBCDIC to ASCII, input EBCDIC shift-out (‘0E’X) becomes (‘1E’X), and ‘0F’X becomes ‘1F’X.

**SOsi EBCDIC**
Specifies that DBCS data strings in the ASCII output are delimited by EBCDIC shift-out/shift-in characters. As the data is translated from EBCDIC to ASCII, input EBCDIC shift-out (‘0E’X) remains EBCDIC shift-out (‘0E’X), and ‘0F’X remains ‘0F’X.

**SOsi NONE**
Specifies that DBCS data strings in the ASCII output are not delimited by any shift-out/shift-in characters. As the data is translated from EBCDIC to ASCII, any EBCDIC DBCS shift characters are removed.

**SOsi SPACE**
Specifies that DBCS data strings in the ASCII output are delimited by ASCII space (‘20’X) characters. As the data is translated from EBCDIC to ASCII, input EBCDIC shift-out (‘0E’X) becomes ASCII space (‘20’X) and EBCDIC shift-in (‘0F’X) also becomes ‘20’X.

**TChinese**
Causes the data to be converted from EBCDIC DBCS to Traditional Chinese (5550) ASCII DBCS when it is sent to the remote system. LPR loads the TCHINESE DBCS translation table from the TCPCHBIN binary translate table data set.

**TIMEout**
Specifies that LPR wait 5 minutes for an ACK or NACK from the LPD printer. If ACK or NACK does not arrive, LPR terminates the connection with Error Number=73. Since LPR waits for ACK in several places, this error can occur in different LPR messages, such as EZB1048E.

For some printers, the ACK is not returned until the job has printed. For these printers, specifying the TIMEout option can cause long jobs to stop printing.
When the TIMEout option is not specified, LPR waits as long as the TCP/IP connection exists. For some printers, this could tie up the PORT (and any NPF thread using LPR on that PORT) until an operator intervenes to fix the printer. For NPF users, TIMEout should be added as an LPR option for these printers. Refer to the [z/OS Communications Server: IP Network Print Facility](https://www.ibm.com/support/docview.wss?uid=swg21262406) for information about configuring LPR options in NPF.

**Title title**
Specifies the title assigned to a data set printed with the FILTER p option.

**TOPmargin number**
Specifies the number of lines designated for the top margin.

**NOTOPmargin**
Indicates that blank lines are not inserted at the top of each page.

**TRACe**
Turns on the trace details for interaction with the remote printer. TRace always overrides TYpe because TYpe is a subset of TRace.

**TRANslatetable name**
Specifies the SBCS translate table to be used by the client. The name parameter is preceded by either the userid or the hlq and followed by TCPXLBIN to form the data set name of the translate table (userid.name.TCPXLBIN or hlq.name.TCPXLBIN). If both data sets exist, a search order hierarchy determines which one is to be used.

Refer to the [z/OS Communications Server: IP Configuration Reference](https://www.ibm.com/support/docview.wss?uid=swg21262406) for more information about search order hierarchy, loading, and customizing SBCS translation tables. XLatetable is a synonym for this option.

**TYpe**
Displays the progress of the command as the data set is being processed.

**USCFxlate**
Specifies that a single byte translation table such as JPNKANA be used for the print data. The control file generated by LPR and sent to LPD contains upper- and lowercase alphanumeric characters. Specify this option if any of them are being translated incorrectly, causing the LPD to reject the print jobs.

**User name**
Specifies a name that overrides the user identification of the program that is requesting the print job, prints on the banner page, and becomes the user identification of the mail option. The name field cannot be longer than eight characters. If you do not enter the user name parameter, it defaults to the system user identification or to the job name.

**Version**
Displays the version of the program.

**Width width**
Specifies the line width of a data set printed with the FILTER options f, l, p, or r.

**Xlatetable name**
Specifies the SBCS translate table to be used by the client. The name parameter is preceded by either the userid or the hlq and followed by TCPXLBIN to form the data set name of the translate table (userid.name.TCPXLBIN or hlq.name.TCPXLBIN). If both data sets exist, a search order hierarchy determines which one is to be used.
Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about search order hierarchy, loading, and customizing of SBCS translation tables. TRANslatetable is a synonym for this option.

**-o option**

Specifies an option that the control file in PSF for AIX (PSF/6000) or InfoPrint uses to format the print job. Any -o option honored by PSF/6000 can be passed to the control file using this parameter of the LPR command. This parameter must be issued without a blank between the -o and the option.

Below are some sample options. None of these options are standard LDP options. They are not honored by most LPD print servers.

- **-ochars=GT15**
  
  Sets the value of the CHARS parameter to GT15.

- **-obin=2**
  
  Sets the input bin to 2 (use alternate input bin).

- **-opagedef=P13700**
  
  Sets the value of PAGEDEF parameter to P13700.

- **-oformdef=F1SEPA**
  
  Sets the value of the FORMDEF parameter to F1SEPA.

- **-ocopies=002**
  
  Sets the JCL COPIES count to 2. This parameter is not honored by most LPD servers. Other LPD servers ignore all control file information and print one copy. For these printers, issue the LPR command multiple times to get multiple copies.

*IBM Print Services for AIX (S544-3878-03)* contains detailed descriptions of the -o options for PSF/6000.

**Examples**

- Print the data set TEST.LISTING on a printer named lp0 on the system mvs1:
  
  LPR TEST.LISTING (PRINTER lp0 HOST mvs1)

- If TEST.LISTING has a record format that contains carriage control such as VBA, the first character of each line is interpreted as carriage control. To prevent the first character of each line from being interpreted as carriage control, use the following command:
  
  LPR TEST.LISTING (PRINTER lp0 HOST mvs1 NOCC)

- If this LPRSET command was issued:
  
  LPRSET lp0@mvs1

  the following LPR command would also print the data set on printer LP0 on the host MVS1 and prevent the first character of each line from being interpreted as carriage control:

  LPR TEST.LISTING (NOCC)

- Print the data set TEST.LISTING in landscape mode:
  
  LPR TEST.LISTING (LANDSCAPE)

  The following example shows the PostScript attributes used in the LANDscape option:

  ```
  614 25 translate 90 rotate .88 .76 scale
  /n 1 def
  /fs 10 def
  /ls 11.2 def
  /ld ls 2 mul def
  /lt ls 3 mul def
  ```
To understand these attributes you may need to reference a PostScript manual.

- If a data set TEST has a low-level qualifier of LISTPS (PostScript), use the following command to send TEST to a PostScript-capable printer without specifying the PostScript option:

```
LPR TEST.LISTPS (PRINTER lp0 HOST mvs1
```

- Print a FORTRAN source program with 57 lines on each page:

```
LPR TEST.FORTRAN (LINECOUNT 57
```

### Usage

- When sending a print job to a printer that has RACF in its definition in the LPD.CONFIG data set, you must specify the password. If it is for a different user ID, you must specify that password and user ID as follows:

```
Job PASS=password,FOR=userid
```

- If the printer or host name are not specified in the LPR command, the last LASTINGGLOBALV variables for PRINTER and PRTHOST in the user_id.LASTINGGLOBALV data set are used as the defaults. You can specify these variables with the LPRSET command.

- LPR normally issues messages only if there is an error. If you want to track the progress of the command, use the TYPE or TRACE parameter.

- You can use the LPR command to send PostScript data sets to a printer that can print documents in that language. LPR checks that no incompatible options were given, if it is a PostScript data set. You can override this check, if you want to print a PostScript program with the NOPostscript option. UNIX systems examine the first few characters of a data set (looking for %!) to determine if a data set is a PostScript data set. If you have PostScript data sets that do not contain the characters %!, use the POSTscript parameter to add them.

- Carriage control is interpreted line by line. A data set can mix ASA and machine carriage control. Interpretation is done by converting the controls to the appropriate ASCII sequences, before the data set is sent to the remote system. Lines that have incorrect carriage control are not printed.

- When a data set is printed without carriage control, LPR adds a heading line that shows the name of the data set, the title of the system on which the LPR command is running, and a page number. You can specify the number of lines to be printed (excluding the 3 heading lines) with the LInecount parameter.
When you specify a filter code, LPR ignores CC, HEader, NOCc, NOHeader, and TOpmargin. When a filter code of f, l, p, or r is specified, LPR stops paginating the data set it is printing. Instead, it sends the data in the data set as plain lines. The following list provides a description of these filter codes:

<table>
<thead>
<tr>
<th>Filter code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>Print as a sequence of lines.</td>
</tr>
<tr>
<td>l</td>
<td>Print, passing through all control characters.</td>
</tr>
<tr>
<td>p</td>
<td>Print with pagination.</td>
</tr>
<tr>
<td>r</td>
<td>Print, interpreting the first column as FORTRAN carriage control characters. The supported IBM FORTRAN carriage control characters are 1, 0, +, and -.</td>
</tr>
</tbody>
</table>

When using the MVS LPD server to print files with filter code r, the PAGESIZE parameter on the SERVICE statement for that printer is ignored in the LPD CONFIG file. The PAGESIZE parameter defaults to 60 if it is not specified. To prevent unwanted page ejects, change the value specified on the PAGESIZE parameter to a number greater than the actual number of lines on the pages being printed. For example, you can specify 100 000 for the PAGESIZE parameter.

When you specify a filter code of c, d, g, n, t, or v, LPR transmits the data as a byte stream (as though you specified the BINary option).
LPRM command—Remove a job from the printer queue on a remote host

**Purpose**

Use the LPRM command to remove a job from the printer queue on a remote host.

**Notes:**

1. Do not use the forward slash character (/) in any parameter value for this command.
2. The TSO LPRM command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

**Format**

```
LPRM [job_id] Opt Parms 1: Opt Parms 2:
```

Opt Parms 1:

```
( Printer name Host host AT host
```

Opt Parms 2:

```
TRace TYpe Version
```

**Parameters**

*job_id*

Specifies either a user ID (this must not start with a digit), or a job number in the remote printer queue. If you do not specify *job_id* with the LPRM command, your currently active job is removed.

*Printer name*

Specifies the name of the printer associated with the job.

*Host host*

Specifies the name or IPv4 IP address of the printer host. If *host* is a name, it must resolve to an IPv4 address. AT is accepted as a synonym for HOST.

*AT host*

Specifies the name or IPv4 IP address of the printer host. If *host* is a name, it must resolve to an IPv4 address. Host is a synonym for this option.

*TRace*

Turns on the trace details for interaction with the remote printer. TRace always overrides TYpe because TYpe is a subset of TRace.

*TType*

Displays the progress of the command.
Version
Displays the version of the program.

Examples

- Cancel job number 123 on the printer lp0 on the local system os2sys1:
  LPRM 123 (PRINTER lp0 HOST os2sys1
  LPRM 123 (PRINTER lp0 AT os2sys1

  If the job is in the queue, it is removed. If the job is currently active, it is stopped.
  - If the LPRSET command was previously issued (LPRSET lp0@os2sys1), using
    the following LPRM command has the same effect as issuing the command in
    the previous example:
      LPRM 123

- Cancel the currently active job:
  LPRM (PRINTER lp0 HOST os2sys1

Usage

- If the printer and host name are not specified in the LPRM command, the last
  LASTING.GLOBALV variables for PRINTER and PRTHOST in the
  user_id.LASTING.GLOBALV data set are used as the defaults. You can set these
  variables with the LPRSET command. You can use these variables to set up a
  default printer, which is used if you do not specify a printer.

- Removing the currently active job can depend on the number of jobs currently
  printing. If you have two jobs printing, and you use the LPRM command without
  the job_id parameter, the first job may finish, but you could inadvertently remove
  the second job instead.
LPRSET command—Set the default printer and host name

Purpose

Use the LPRSET command to set the default printer and host name. The printer and host name can also be included in the line printer commands (LPR, LPQ, and LPRM).

Notes:

1. Do not use the forward slash character (/) in any parameter value for this command.
2. The TSO LPRSET command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

Format

```
LPRSET printer@host
```

Optional Parameters:

```
QUERY  TYPe  VERSION
```

Parameters

* `printer@host`  
  Specifies the name of the printer and host to be used. If `host` is a name, it must resolve to an IPv4 address.

* `Query`  
  Displays the current settings for the default printer and host.

* `TRace`  
  Turns on the trace details for the recording of the printer and remote host name. TRace always overrides TYPe because TYPe is a subset of TRace.

* `TYPe`  
  Displays the progress of the command.

* `Version`  
  Displays the version of the program.

Examples

- Set the default printer and host as the printer `lp0` on the local system `mvs1`:

  ```
  LPRSET lp0@mvs1
  ```

- Display the current version of LPRSET:

  ```
  LPRSET (VERSION
  ```

- Display the current settings:

  ```
  LPRSET (QUERY
  ```
Usage

- When you use LPRSET to set the printer and host, a data set by the name `userid.LASTING.GLOBALV` (where `userid` is the TSO user ID) is created or updated. If this data set does not exist, you must be able to create this data set. `userid.LASTING.GLOBALV` is the name required by LPRSET.

- Printer names can be case sensitive. The printer name must be spelled the way the host uses it. For example, on UNIX systems, `lp0` and `LP0` can refer to different printers.

Also be aware that ISPF panels default to uppercase unless otherwise specified.

- When you query the current settings on the same command that you set the default printer and host, the query of the current settings is done before the specified default printer and host are set. For example, if the printer was set to `PRINT1@RALVM13`, and you perform the query, `LPRSET PRINT2@RALVM13(Q`, the message returned says, `PRINTER is PRINT1@RALVM13`. However, it also changes the default printer to `PRINT2`. So if you query again, it would respond, `PRINTER is PRINT2@RALVM13`.

If you want to set the default printer and host and see that the defaults were set in the same LPRSET command, use the TYPE parameter. For example, if you want to set the default printer and host to `PRINT2` on RALVM13 and then see the results afterward, you would enter the following command.

```
LPRSET PRINT2@RALVM13 (TYPE
```
Chapter 9. Using GDDMXD/MVS with the X Window system

This chapter describes GDDMXD/MVS and the GDDMXD CLIST. This chapter also describes how to use GDDMXD/MVS user-specified options and keyboard functions. Problem determination information associated with GDDMXD/MVS is also described in z/OS Communications Server: IP Diagnosis.

Note: The feature HIP612X is required for GDDMXD.

The following subjects are covered in this chapter:

- “Overview of GDDMXD/MVS”
- “Using GDDMXD/MVS” on page 345
- “User-specified options” on page 348
- “Keyboard functions” on page 361

Overview of GDDMXD/MVS

GDDMXD/MVS is an interface that allows graphics from the IBM Graphical Data Display Manager/MVS to be displayed on workstations that support the X Window System. When GDDMXD/MVS is installed and activated, the data stream created by the GDDM application is translated to the X Window System protocol and transmitted by TCP/IP to the X Window System server for the display. If GDDMXD/MVS is installed and not activated, or has been made inactive, the GDDM application transmits data to its supported display device as if GDDMXD/MVS were not present.

Keyboard and character set mappings

The following member names contain the described character sets:

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDXALTCS</td>
<td>A member of hlq.SEZAINST that contains a second character set (the 3270 alternate character sets).</td>
</tr>
<tr>
<td>GDXAPLCS</td>
<td>A member in the hlq.SEZAINST data set that contains sample keyboard mapping for APL2®.</td>
</tr>
<tr>
<td>KEYCODE</td>
<td>A member of hlq.SEZALOAD that displays key codes. You can edit this data set to change keyboard mappings.</td>
</tr>
</tbody>
</table>

Executable code

The following member names contain the described executable code or sample:

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDXLIOX0</td>
<td>A member of hlq.SEZALOAD that contains executable code</td>
</tr>
<tr>
<td>XDEFAULT</td>
<td>A member of hlq.SEZAINST that contains the sample XDEFAULT member for GDDMXD/MVS</td>
</tr>
</tbody>
</table>

GDDM application limitations

When GDDMXD/MVS is inactive, there are no GDDM application restrictions. The following types of functions are not supported by GDDMXD/MVS:

- Multiple instances of the GDDM application
- Opening multiple display devices at one time
- Operator windows
GDDM display limitations

GDDMXD/MVS appears as an IBM 3179G device model to the GDDM application. When the HostRast option is active, the device model is an IBM 3279. The IBM 3179 Model G Color Graphics Display Station is used regardless of the display device nickname presented by the application.

The following are characteristics of the GDDMXD/MVS IBM 3179G display station:

**Alphanumeric Cursor**
When the graphics cursor is not enabled, the alphanumeric cursor can be repositioned by pointing the X Window System pointer device cursor for the GDDMXD window at the desired character location and pressing the pointing device button.

**Attached Graphics Cursor**
When the graphics cursor is attached, the X Window System pointer device cursor for the GDDMXD window changes to a crosshair pattern and moves as the pointer device is moved.

**Blinking Character Attribute**
Ignores the blinking character attribute.

**Character Display**
Characters with an EBCDIC value of less than hex 40 are displayed as blanks.

**Color Mixing**
GDDMXD/MVS supports only the overpaint foreground color mix mode. The initial color of the image area is black, and mixing with the actual background colors is not performed.

An exception is made for data passed by an image data order. In this exception, a combined foreground color mix mode is supported, if the series of begin image orders have exactly the same parameter values.

When the HostRast option is active, color mixing is performed by GDDM, and the preceding exception does not apply.

**Default Vector Symbol Set**
The Default Vector Symbol Set is not supported.

**Detached Graphics Cursor**
When the graphics cursor is detached by the data stream or is not attached, the X Window System pointer device cursor for the GDDMXD window changes to an open arrow when the keyboard is unlocked, or changes to an X shape when the keyboard is locked.

**Detectable Fields**
Ignores the detectable fields.

**Pixel Spacing**
When the HostRast option is not active, the vertical and horizontal pixel spacing of the actual display device that is obtained from the X Window System is supplied to the GDDM application. When the HostRast option is active, the pixel spacing of an IBM 3279 Color Display Station is supplied to the GDDM application.

**Visual Appearance**
For all programmed symbol and image data that is received from the GDDM application, each GDDM pixel is mapped to one X Window System display pixel, which causes a different appearance from the same data.
stream displayed on an IBM 3179G Color Graphics Display Station. This map process can also cause display differences in the placement of alphanumeric field data over the graphics display and in the appearance of the filled areas of the graphic display. When the HostRast option is active, aspect ratio distortion of the displayed graphics appears, unless the aspect ratio of the X Window System display is the same as the IBM 3279.

Using GDDMXD/MVS

Before GDDM data can be sent to an X Window System display, activate GDDMXD/MVS by invoking the GDDMXD CLIST. Make sure that you have already copied the GDDMXD CLIST to your system CLIST data set. If you have not, refer to the z/OS Communications Server: IP Configuration Reference for instructions.

Note: If you do not want to run GDDM applications through the X Window System, do not enable GDDMXD/MVS.

To invoke the GDDMXD CLIST, use the GDDMXD command in the format described in “GDDMXD command—Invoke the GDDMXD CLIST” on page 346.
GDDMXD command—Invoke the GDDMXD CLIST

**Purpose**
Use the GDDMXD command to invoke the GDDMXD CLIST.

**Format**
```
GDDMXD [ON|OFF]
```

**Parameters**
- **ON**
  Enables GDDMXD/MVS. GDDM output is sent to the X Window System display. The system responds with `GDDMXD/MVS active`.
- **OFF**
  Disables GDDMXD/MVS. The system erases the data set that was created when GDDMXD/MVS was activated and responds with `GDDMXD/MVS inactive`.
Identifying the target display

**Purpose**
A TSO global variable is used by the X Window System to identify the IP address of the target display based on the contents of the data set `user_id.XWINDOWS.DISPLAY`.

**Format**

```
> [internet_address::target_server].target_screen
```

**Parameters**

- `internet_address`
  - Specifies the IPv4 IP address of the host machine on which the X Window System server is running.
- `:target_server`
  - Specifies the number of the display server on the host machine.
- `.target_screen`
  - Specifies the screen to be used on the `target_server`.

**Examples**

Examples of the contents of the `user_id.XWINDOWS.DISPLAY` data set.

- `charm.cambridg.ibm.com:0.0`
- `129.42.3.109:0.0`
GDDMXD usage notes

- When you use the TSO Session Manager with GDDMXD, enter a null line in the host session window after the GDDMXD graphics window disappears. This updates and activates the host session window.
- When you run PL/I GDDM applications, do not let the ISASIZE run-time option default to 0. This causes excessive allocation of storage below the 16 megabyte line, and causes a variety of storage allocation abends. Enter a run-time option for ISASIZE, such as ISASIZE(20K), to prevent storage allocation abends.
- Although GDDMXD provides its own device information to the GDDM application, normal GDDM device initialization occurs. A full-screen 3270 TSO session from a real or emulated 3270 terminal with 80 columns and 32 rows is required to invoke the GDDM application.

Resizing the GDDMXD graphics window

GDDMXD supports four graphics presentation space sizes. The size of the graphics presentation space used by GDDMXD is determined by the window width specified by the Geometry option in the user_id.X.DEFAULTS data set (see “User-specified options” on page 348 for more information). The size is also determined dynamically when you resize the GDDMXD graphics window. The relationship between the size of the graphics presentation space and the window width is shown in Table 22.

Table 22. Supported graphics presentation space sizes

<table>
<thead>
<tr>
<th>Window width (pixels)</th>
<th>GDDMXD graphics presentation space (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 650</td>
<td>480 horizontal by 352 vertical</td>
</tr>
<tr>
<td>&gt;=650 to &lt; 850</td>
<td>720 horizontal by 512 vertical</td>
</tr>
<tr>
<td>&gt;= 850 to &lt;= 1024</td>
<td>960 horizontal by 682 vertical</td>
</tr>
<tr>
<td>&gt; 1024</td>
<td>1200 horizontal by 864 vertical</td>
</tr>
</tbody>
</table>

For graphics presentation space sizes other than the default size (720 pixels by 512 pixels), bit-mapped data, such as symbol sets and images, is expanded or compressed to meet the scaling requirements of the specified graphics presentation space.

You can expand bit-mapped data by duplicating rows and columns of the data. The resulting view can differ slightly from the default-size view. You can compress single-plane bit-mapped data by combining rows and columns of the data with a logical OR function. Because this might not yield acceptable results when a black on white image is viewed, the Compr option is provided to specify that a logical AND function be used to compress the data. See “Compr option—Control the technique used to compress bit-mapped data” on page 352 for more information about using this option. You can compress multiplane bit-mapped data by eliminating certain rows and columns. Data compression produces a view that is different from the default-size view.

User-specified options

The user-specified options for GDDMXD/MVS are entries in a data set called user_id.X.DEFAULTS. The user_id.X.DEFAULTS data set is searched during initialization of GDDMXD/MVS.
**Note:** The values in the `user_id.X.DEFAULTS` data set are case sensitive and must be entered as shown.

The options listed in Table 23 are supported by GDDMXD/MVS.

### Table 23. GDDMXD/MVS options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANFontn</td>
<td>Specifies the X Window System font used for characters in the alphanumeric presentation space.</td>
<td>350</td>
</tr>
<tr>
<td>CMap</td>
<td>Specifies whether the default color map is loaded or bypassed.</td>
<td>351</td>
</tr>
<tr>
<td>Compr</td>
<td>Controls the technique used to compress bit-mapped data when a graphics window size of 480 by 352 pixels is specified.</td>
<td>352</td>
</tr>
<tr>
<td>Enter</td>
<td>Overrides the default key mapping for Enter.</td>
<td>353</td>
</tr>
<tr>
<td>GColornn</td>
<td>Specifies a color name.</td>
<td>354</td>
</tr>
<tr>
<td>Geometry</td>
<td>Specifies the size and location of the initial GDDMXD graphics presentation space.</td>
<td>355</td>
</tr>
<tr>
<td>GMCPnn</td>
<td>Overrides GDDM multicolor patterns with workstation color names.</td>
<td>356</td>
</tr>
<tr>
<td>HostRast</td>
<td>Performs raster image processing at the host.</td>
<td>357</td>
</tr>
<tr>
<td>NewLine</td>
<td>Overrides the default key mapping for NewLine.</td>
<td>358</td>
</tr>
<tr>
<td>XSync</td>
<td>Requests that the X Window System process one request at a time.</td>
<td>359</td>
</tr>
<tr>
<td>ZWL</td>
<td>Tells GDDMXD/MVS to draw all lines using 0-width lines.</td>
<td>360</td>
</tr>
</tbody>
</table>
ANFontn option—Specify the X Window system font used for characters in the alphanumeric presentation space

**Purpose**
Use the ANFontn option to specify the X Window System font that GDDMXD should use to display characters in the alphanumeric presentation space of the GDDMXD window.

**Format**

```plaintext
  gddmx*ANFontn:—fontname
```

**Parameters**

- **n**: Specifies presentation space size.
- **fontname**: Specifies the name of the X Window System font.

**Examples**
The following are examples of ANFontn options.

- `gddmx*ANFont1: Rom8`
- `gddmx*ANFont3: Rom14`

**Usage**
Graphics mode 1 and 2 characters in the graphics presentation space are not affected by this option. The value of `n` is in the range 1–4 and defines the X Window System font for each of the four sizes of presentation space supported by GDDMXD. You can specify the ANFontn option for any, all, or none of the four values for `n`. The X Window System fonts specified should be fixed-space fonts that have characters that fit into the character box size required by each of the four presentation space sizes.

<table>
<thead>
<tr>
<th>n</th>
<th>Presentation space</th>
<th>Character box</th>
<th>Example font</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>480 x 352</td>
<td>6 x 11</td>
<td>Rom8</td>
</tr>
<tr>
<td>2</td>
<td>720 x 512</td>
<td>9 x 16</td>
<td>Rom11</td>
</tr>
<tr>
<td>3</td>
<td>960 x 682</td>
<td>12 x 21</td>
<td>Rom14</td>
</tr>
<tr>
<td>4</td>
<td>1200 x 864</td>
<td>15 x 27</td>
<td>Rom17</td>
</tr>
</tbody>
</table>

If you select a font that has characters that are larger than the character box size, the characters might overlap when displayed.
CMap option—Specify whether the default color map is loaded or bypassed

Purpose
Use the CMap option to specify whether the default color map is loaded or bypassed.

Format

```
gdmxD/MVS: CMap: \(Y\) \(N\)
```

Parameters

- **Y** Directs GDMX/MVS to load the default color map. This is the default.
- **N** Directs GDMX/MVS to bypass loading the default color map.

Usage
During initialization, GDMX/MVS issues the X Window System call, XInstallColormap, to load the default color map. If the CMap option is specified as N, the XInstallColormap call is not made. This option is for X Window System servers that load their own color map and do not want the clients to load any other color map.
Compr option—Control the technique used to compress bit-mapped data

**Purpose**
Use the Compr option to control the technique used to compress bit-mapped data when a graphics window size of 480 by 352 pixels is specified.

**Format**
```
> gddmx*Compr:
```

**Parameters**

- **O or o**
  Specifies that a logical OR function must be used when compressing bit-mapped data. This is the default.

- **A or a**
  Specifies that a logical AND function must be used when compressing bit-mapped data.
Enter option—Override the default key mapping for Enter

**Purpose**
The Enter option can be specified in the `user_id.X.DEFAULTS` data set to identify which X Window System Keysym is to be mapped to the Enter function. This option overrides the default mapping of the Keysym `XK_Execute` to the Enter function.

**Format**

```
gddmx*Enter: keysym_name
```

**Parameters**

*keysym_name*

Specifies the X Window System Keysym representing the physical key. For standard Keysyms, the `XK_` prefix is not included in specifying the option.

**Examples**

In the following example of the Enter option, the X Window System Keysym, `XK_Return`, is mapped to the Enter function.

```
gddmx*Enter: Return
```
GColornn option—Specify a color name

Purpose
GDDMXD/MVS provides a default mapping of GDDM colors to X Window System colors. Use the GColornn option to override a default color name or to specify a color if a default color name is not available by your X Window System server.

Format

```
\text{gddmx*GColornn:} c
```

Parameters

\textit{nn}: Specifies the GDDM color entry that is mapped.

\textit{c}: Specifies the X Window System color that is used as the GDDM color.

Usage

Table 24 lists the GDDM colors that GDDMXD/MVS maps to the X Window System.

<table>
<thead>
<tr>
<th>GColornn</th>
<th>GDDM color</th>
<th>X Window System color</th>
</tr>
</thead>
<tbody>
<tr>
<td>GColor1</td>
<td>Blue</td>
<td>Blue</td>
</tr>
<tr>
<td>GColor2</td>
<td>Red</td>
<td>Red</td>
</tr>
<tr>
<td>GColor3</td>
<td>Magenta</td>
<td>Magenta</td>
</tr>
<tr>
<td>GColor4</td>
<td>Green</td>
<td>Green</td>
</tr>
<tr>
<td>GColor5</td>
<td>Turquoise</td>
<td>Turquoise</td>
</tr>
<tr>
<td>GColor6</td>
<td>Yellow</td>
<td>Yellow</td>
</tr>
<tr>
<td>GColor7</td>
<td>White</td>
<td>White</td>
</tr>
<tr>
<td>GColor8</td>
<td>Black</td>
<td>Black</td>
</tr>
<tr>
<td>GColor9</td>
<td>Dark Blue</td>
<td>Dark Slate Blue</td>
</tr>
<tr>
<td>GColor10</td>
<td>Orange</td>
<td>Orange</td>
</tr>
<tr>
<td>GColor11</td>
<td>Purple</td>
<td>Plum</td>
</tr>
<tr>
<td>GColor12</td>
<td>Dark Green</td>
<td>Dark Green</td>
</tr>
<tr>
<td>GColor13</td>
<td>Dark Turquoise</td>
<td>Dark Turquoise</td>
</tr>
<tr>
<td>GColor14</td>
<td>Mustard</td>
<td>Wheat</td>
</tr>
<tr>
<td>GColor15</td>
<td>Gray</td>
<td>Gray</td>
</tr>
<tr>
<td>GColor16</td>
<td>Brown</td>
<td>Brown</td>
</tr>
</tbody>
</table>

Examples

The following is an example of using a GColornn option to override a default color:
```
gddmx*GColor3: Pink
```

In this example, specifying the GColor3 entry in the user_id.X.DEFAULTS data set maps the GDDM color, magenta, to the X Window System new color of pink.
Geometry option—Specify the size and location of the initial GDDMXD graphics presentation space

**Purpose**
Use the Geometry option to specify the size and location of the initial GDDMXD graphics presentation space.

**Format**
```
gddmx*Geometry: width x height + x_offset + y_offset
```

**Parameters**
- **width**
  Specifies the initial width of the GDDMXD graphics window. The width determines the initial size of the graphics presentation space.

- **height**
  Specifies the initial height of the GDDMXD graphics window.

- **x_offset**
  Specifies the location of the upper left corner of the window where \( x_{\text{offset}} \) is the horizontal offset from the upper left corner of the display.

- **y_offset**
  Specifies the location of the upper left corner of the window where \( y_{\text{offset}} \) is the vertical offset from the upper left corner of the display.

**Examples**
The following is an example of a Geometry option:
```
gddmx*Geometry: 750x600+20+20
```
GMCPnn option—Override GDDM multicolor patterns with workstation color names

**Purpose**

Use the GMCPnn option to override GDDM multicolor patterns with workstation color names.

**Format**

```
GDDM+xGMCPnn: c
```

**Parameters**

*nn:*

  Specifies the GDDM multicolor pattern.

*c:*

  Specifies the color that is used with the defined GDDM multicolor pattern.

**Examples**

The following is an example of a GMCPnn option:

```
gddmx*GMCP126: MediumBlue
```

In this example, the color medium blue is used when multicolor pattern 126 is specified by the GDDM application.
HostRast option—Perform raster image processing at the System/370 Host

**Purpose**
Use the HostRast option to perform raster image processing at the System/370 host.

Use the HostRast option when:
- Multiplane character symbol sets are required by the application.
- GDDM color mixing is important to the application.

The default device model for GDDMXD/MVS is an IBM 3179G with a mouse and the raster image processing is performed at the workstation.

**Format**

```
$ gddmx*HOSTRAST: N
```

**Parameters**

- **N** Directs GDDMXD/MVS to use the IBM 3179G as a device model. This is the default.
- **Y** Directs GDDMXD/MVS to use the IBM 3279 as a device model.
- **X** Directs GDDMXD/MVS to use the IBM 3279 as a device model and expand the pixel mapping to reduce aspect ratio distortion.

**Usage**
- The APL2 character set is not supported when the HostRast option is active.
- When the HostRast option is specified as Y, the GDDM application performs the raster image processing and transmits the picture as a series of characters whose pixel definitions have been transmitted to Programmed Symbol Sets. The picture is mapped exactly as an IBM 3279.
- If the ratio of horizontal to vertical pixel spacing is not the same as that of an IBM 3279, the aspect ratio can be distorted.
NewLine option—Override the default key mapping for NewLine

Purpose
The NewLine option can be specified in the users id.X.DEFAULTS data set to identify which X Window System Keysym is to be mapped to the NewLine function. This option overrides the default mapping of the Keysym XK_Return to the NewLine function.

Format

```
gddmx*NewLine: keysym_name
```

Parameters

`keysym_name`
Specifies the X Window System Keysym representing the physical key. For standard Keysyms, the XK_prefix is not included in specifying the option.

Examples
In the following example of the NewLine option, the X Window System Keysym, KP_Enter, is mapped to the NewLine function:

```
gddmx*NewLine: KP_Enter
```
XSync option—Request that the X Window system process one request at a time

**Purpose**
The X Window System operates asynchronously. By the time an error has been detected, more requests could have been issued by the application.

Use the XSync option to request that the X Window System process one request at a time.

**Format**

```
gdmx*XSync: N Y
```

**Parameters**

- **N**: Allows the X Window System to operate asynchronously. This is the default.
- **Y**: Directs GDDMXD/MVS to cause the X Window System to operate synchronously.

**Usage**
Be aware that system performance goes down when you use XSync=Y.
ZWL option—Tell GDDMXD/MVS to draw all lines using 0-width lines

Purpose
The X Window System supports a range of line widths. Because some X Window System servers draw wide lines at a slow rate, you can use the Zero Width Lines (ZWL) to tell GDDMXD/MVS to draw all lines using 0-width lines. The X Window System server uses the fastest process to draw the lines. The resulting line might not be exactly the same as if it had been drawn as a wide line.

Format

\[ \text{gddmx\*ZWL:} \begin{bmatrix} N \\ Y \end{bmatrix} \]

Parameters

N  Directs GDDMXD/MVS not to use 0-width lines for all drawing. This is the default.

Y  Directs GDDMXD/MVS to use 0-width lines for all drawing.
Keyboard functions

The following sections detail different keyboard functions supported by GDDMXD/MVS.

GDDMXD/MVS keyboard functions

When you enter input to the GDDM application by GDDMXD/MVS, use the following 3270 keyboard functions.

- All alphanumeric keys
- F1 – F24
  If F13 – F24 are not available, use Shift + F1 to Shift + F12
- Tab or Shift + Tab
- Directional arrows
- End key to erase to the end of the field
- Insert key and Delete key
- PA1, PA2, and PA3
- Enter key
- Newline key

Note: The Backspace key is treated as a cursor left key.

If you cannot locate these keys on your workstation, refer to your workstation X Window System documentation to determine the mapping of X Window System key symbol definitions to the physical keys.

GDDMXD/MVS to X Window system keyboard functions

The following are the GDDMXD/MVS keyboard functions that translate to X Window System key symbol definitions. Key functions not listed are not supported.

<table>
<thead>
<tr>
<th>GDDMXD/MVS keyboard function</th>
<th>X Window System key symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>APL2 character set toggle</td>
<td>XK_Backspace with state Mod1Mask</td>
</tr>
<tr>
<td>Clear</td>
<td>XK_Pause</td>
</tr>
<tr>
<td>Delete</td>
<td>XK_Delete</td>
</tr>
<tr>
<td>Down</td>
<td>XK_Down</td>
</tr>
<tr>
<td>End</td>
<td>XK_End</td>
</tr>
<tr>
<td>Enter</td>
<td>XK_Execute</td>
</tr>
<tr>
<td>Insert</td>
<td>XK_Insert</td>
</tr>
<tr>
<td>F1 – F12</td>
<td>XK_F1 – XK_F12</td>
</tr>
<tr>
<td>Left</td>
<td>XK_Left</td>
</tr>
<tr>
<td>Newline</td>
<td>XK_Return</td>
</tr>
<tr>
<td>PA1</td>
<td>XK_Prior</td>
</tr>
<tr>
<td>PA2</td>
<td>XK_Next</td>
</tr>
<tr>
<td>PA3</td>
<td>XK_Home</td>
</tr>
<tr>
<td>Right</td>
<td>XK_Right</td>
</tr>
<tr>
<td>Tab</td>
<td>XK_Tab</td>
</tr>
<tr>
<td>Up</td>
<td>XK_Up</td>
</tr>
</tbody>
</table>
APL2 character set keyboard

The APL2 character set is activated by simultaneously pressing the X Window System XK_Backspace key (usually the Backspace key) and the State Mod1Mask key (usually the Alt key). For example, if you use the IBM 101 Enhanced Keyboard, the APL2 character set is toggled on and off by pressing and holding the Alt key, and then pressing the Backspace key.

When the APL2 character set is active, the characters APL are displayed in the title bar of the GDDMXD/MVS window.

In the X Window System, a key code is assigned to each key on the keyboard. GDDMXD/MVS uses key codes in combination with modifier keys. For example, the Shift and Alt keys determine the data that should be passed back from GDDMXD/MVS to the X Window System application to identify the user’s keystroke data.

GDDMXD looks for the data set hlq.SEZAINST(GDXALTCS) when it is initialized. Before using GDDMXD, copy the installed TCP/IP copy of hlq.SEZAINST(GDXALTCS) to user_id.GDXALTCS.PSS, or allocate the common installed copy of hlq.GDXALTCS.PSS to ddname GDXDACSP.

A default map for the APL2 character set is provided in GDDMXD/MVS, which corresponds to the IBM 101 Key Enhanced Keyboard. You can override this default map by creating a data set called GDXAPLCS.MAP to define the map for your workstation. When GDDMXD/MVS is initialized, the system searches for a data set called GDXAPLCS.MAP. If the GDXAPLCS.MAP data set exists, the data in the GDXAPLCS.MAP data set replaces the default mapping for all keys.

Setting up hlq.GDXAPLCS.MAP

The following steps describe how to set up the GDXAPLCS.MAP data set.

1. Invoke the program KEYCODE from the TCP/IP load module library in hlq.SEZALOAD to determine the key codes for the keyboard keys.

   When KEYCODE is executed from your workstation session to the host system, the key code is displayed for each key pressed at the workstation. Therefore, you can establish the association between a key and the character you want to generate.

   See Appendix B, “Mapping values for the APL2 character set” on page 389 for more information about the mapping values that are defined in the GDXAPLCS.MAP data set.

2. Copy the hlq.GDXAPLCS.SAMPMPM installed with TCP/IP to hlq.GDXAPLCS.MAP.

3. Edit GDXAPLCS.MAP to establish the association between the key codes in the program KEYCODE and the character set and code values in Appendix B, “Mapping values for the APL2 character set” on page 389.

4. GDDMXD looks for the data set user_id.GDXAPLCS.MAP when it is initialized. If you want to use a different data set name, allocate the data set to ddname GDXDACSM.
Chapter 10. Executing commands on a remote host

The Remote Execution Protocol (REXEC) and the Remote Shell Protocol (RSH) are remote execution clients that enable you to execute a command on a remote host and receive the results on the local host. You can execute either REXEC or RSH from the TSO command line or as a batch program. When executed as a batch program, the results are stored in a data set for later use.

To use REXEC, you must have a REXEC daemon running on the remote host. The REXEC client passes the user name, password, and command to the REXEC daemon. The daemon provides automatic logon and user authentication, depending on the parameters that you set.

To use RSH, you must have a REXEC daemon running on the remote host. The RSH client passes the local user name, remote user name, and command to the RSH daemon. The remote user name may be in the form user/password when the RSH daemon is on an MVS host. The daemon provides automatic logon and user authentication, depending on the parameters that you set.

This chapter describes how to use the REXEC and RSH clients.

The following subjects are covered in this chapter:

- **"REXEC command—Execute a command on the remote host and receive the results on your local host" on page 364**
- **"Using the NETRC data set" on page 366**
- **"Submitting REXEC and RSH requests in batch" on page 366**
- **"Using remote execution clients in a z/OS UNIX environment" on page 372**
- **"The z/OS UNIX orexec/rexec command—Execute a command on the remote host" on page 373**
REXEC command—Execute a command on the remote host and receive the results on your local host

Purpose

Use the REXEC command to execute a command on the remote host and receive the results on the local host.

Note: The TSO REXEC command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

Format

```
REXC [option] [foreign_host command]
```

Notes:

1. The REXEC parameters -d, -l, -n, -p, -s, and -t are case sensitive and must be entered in lowercase letters. The user_id and password parameters may be case sensitive, depending on the operating system of the remote host.

Parameters

- `-?` Displays the help message.
- `-d` Activates debug tracing.
- `-n` Prevents automatic logon, if the same foreign host name is defined in the NETRC data set.
- `-l user_id` Specifies the user ID on the foreign host.
- `-p password` Specifies the password for the user ID on the foreign host.
- `-s port` Specifies the TCP port number of the REXEC server on the foreign host. The default, 512, is the port number defined in /etc/services.
- `-t dataset_name` Specifies the data set name of a translation table other than STANDARD. The data set name is userid.tran_table.TCPXLBIN. The default data set name is userid.STANDARD.TCPXLBIN. If this data set is not found, REXEC uses a compiled translation table.
- `foreign_host` Specifies the name or IP address of the foreign host to which you are sending the REXEC command. Specify the foreign host by its host name or IP address. `foreign_host` must be an IPv4 address or resolve to an IPv4 address.
- `command` Specifies the command to be executed on the remote host.
Examples

Use the REXEC command without using NETRC.DATA:

```
READY rexec -l user28 -p user28 -s 512 mvs1 lista
MVS TCP/IP REXEC CS V1R2
SYS1.HELP
GIM.SGIMCLS0
DSN230.DSNCLIST
USER.CLIST
BUILD.CLIST
SYS1.HRFCLST
USER28.RSHD5.JOB00160.D0000103.?
```

Usage

- If you omit the `user_id`, the `password`, or both when entering the REXEC command, the system prompts you to supply the parameters, if the foreign host is not specified with a user ID password in NETRC data set.
- When issuing a command to be executed on the remote host, do not place the command in quotation marks. Doing so may result in unexpected results.

**Note:** There is no such restriction when using the z/OS UNIX orexec command.
- A condition code of 12 will be set when an REXEC batch request encounters one of the following error conditions:
  - The client cannot connect to TCP/IP.
  - The host name cannot be resolved.
  - The translation table cannot be loaded.
Using the NETRC data set

The NETRC data set provides you with an alternative to specifying the user_id and password as REXEC parameters. REXEC uses the following search order to find the NETRC data set to use:
1. NETRC DD statement
2. userid.NETRC.DATA
3. tso_prefix.NETRC
4. userid.NETRC

For information about using the NETRC data set in a batch file, see "Submitting REXEC and RSH requests in batch".

If the user ID is specified on the -I parameter on the REXEC command, no NETRC data sets are used. The user ID and password are case sensitive. If the user ID and password are not in the correct case, failures might occur when connecting to a nonmainframe REXEC server. The following is the format of the NETRC data set:
machine hostname login user_id password password

Note: You can omit your password in the NETRC data set. If you do, REXEC prompts you for your current password.

The following is an example of an NETRC data set:
machine mvs1 login user28 password abcdef

The following is a sample of the response that is displayed as a result of using the REXEC command and the NETRC data set.

```
READY rexec mvs1 lista
MVS TCP/IP REXEC CS V1R2
SYS1.HELP
GIM.SGIMCLS0
DSN230.DSNCLIST
USER.CLIST
BUILD.CLIST
SYS1.HRFCLST
USER28.RSHD5.JOB00161.D0000103.?
```

Submitting REXEC and RSH requests in batch

You usually run REXEC and RSH interactively by entering the command and then receiving the results at your terminal. However, you can also run REXEC and RSH as a batch job. To accomplish this, you must supply the necessary job control language (JCL) and submit it to the job entry subsystem (JES) using the TSO SUBMIT command.

The command format when submitted as a batch job is the same as the command format described in "REXEC command—Execute a command on the remote host and receive the results on your local host" on page 364. You enter the command as a parameter on the EXEC statement. The results of the command executed on the remote host are stored on the local host according to how you define the SYSPRINTF DD statement. The data set characteristics should be consistent with the output from the command you are executing at the remote host.

When you invoke the REXEC command, a check is made to see if a data set is allocated to INPUT. If a data set is allocated, any input is read from that data set.
rather than from your terminal. Similarly, a check is made to see if data set is allocated to OUTPUT. If so, all REXEC output is written to that data set rather than to your terminal.

The REXECD or RSH server does not support output being returned to output DD statements other than SYSTSPRT, SYSPRINT or OUTPUT. Sometimes it is necessary to direct the output from a REXEC or RSH batch request to an OUTPUT DD and to include a SYSPRINT DD with SYSOUT= specified. This is particularly true if the REXEC or RSH command to be executed contains a slash(/).

There may be times when REXEC and RSH are being executed and the user does not want the output to be directed to the SYSPRINT DD file. If a //OUTPUT DD card is coded, output will be directed to it. This DD statement must be directed to a SYSOUT or a validly defined data set. If multiple REXEC or RSH commands are being executed in one step, then it is recommended that DISP=MOD be coded on the //OUTPUT DD card. This will allow each command execution to be appended to the previous output. If the //OUTPUT DD card specifies a nullfile (ie. DSN= NULLFILE or DD DUMMY), then the //OUTPUT DD statement will not be used.

When using the REXECD server, the procedure specified in the TSOPROC argument of the startup procedure must have the //SYSTSPRT DD statement appearing before any other output DD specifications in the procedure. For example, if the batch procedure specified was TSOPROC=TESTJOB, the following example would be the correct specification for the batch procedure for REXECD:

```
//TESTJOB EXEC PGM=IKJEFT01,REGION=4M,DYNAM=30,REGION=4M
//STEPLIB DD DSN=A.LOADLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=*
```

Adding a PARM argument to the EXEC JCL DD statement for commands to be submitted to batch might cause unpredictable output, characters to be lost, or output to be presented out of order.

If DDs are dynamically allocated, the order of output returned might be unpredictable. For example, the following TSO/E CLIST were invoked as the remote execution command, using the procedure TESTJOB:

```
PROC 0
TIME
ALLOC FI(OUT) SYSOUT(*)
OPENFILE OUT OUTPUT
SET &OUT = &STR('THIS; IS THE FIRST LINE')
PUTFILE OUT
SET &OUT = &STR('THIS; IS THE SECOND LINE')
PUTFILE OUT
TIME
SET &OUT = &STR('THIS; IS THE LAST LINE')
PUTFILE OUT
CLOSEFILE OUT
FREE FI(OUT)
EXIT
```

In this case, the output might be returned as:

```
THIS IS THE FIRST LINE
THIS IS THE SECOND LINE
THIS IS THE LAST LINE
output from time
output from time
```
In the following example, the TESTJOB procedure was modified to add the OUT DD statement:

```
//TESTJOB EXEC PGM=IKJEFT01,REGION=4M,DYNAM=30,REGION=4M
//STEPLIB DD DSN=A.LOADLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//OUT DD SYSOUT=*
```

The CLIST was also modified, as in the following example (notice that the ALLOC FI(OUT) and the FREE FI(OUT) have been removed):

```
PROC 0
TIME
OPENFILE OUT OUTPUT
SET &OUT = &STR(THIS; IS THE FIRST LINE)
PUTFILE OUT
SET &OUT = &STR(THIS; IS THE SECOND LINE)
PUTFILE OUT
TIME
SET &OUT = &STR(THIS; IS THE LAST LINE)
PUTFILE OUT
CLOSFILE OUT
EXIT
```

The output appears as in the following example:

```
output from time
output from time
THIS IS THE FIRST LINE
THIS IS THE SECOND LINE
THIS IS THE LAST LINE
```

Be aware that output being returned to the client from DDs other than SYSTSPRT or SYSPRINT might have characters truncated. This behavior has been observed only in the first line of new output files other than SYSTSPRT or SYSPRINT. If you use DDs other than SYSTSPRT or SYSPRINT, you might want to ensure that the first line of the output file contains a blank line, so that no data is lost.

**Notes:**

1. You can also use the NETRC data set name described in "Using the NETRC data set" on page 366 to specify the user ID and password. You can override the NETRC data set search order by specifying a NETRC DD statement in the batch job. The NETRC DD statement identifies the NETRC data set to be used. You must provide all REXEC command information by using the NETRC data set and the PARM keyword on the EXEC statement.

2. Submitting a long running command may cause the REXEC program to end abnormally with a 522 system abend code. This can be avoided by specifying TIME=1440 on the EXEC statement of the JCL you submit. Job step timing is suppressed, including the collection of SMF job time accounting information.

3. If the command to be executed on the remote host contains a slash (/), you must use a preceding slash (/) in the input stream of the NETRC data set or the PARM.

Following is an example of invoking the RSH program in batch and executing the command:

```
LS ./bin/temp/*
```

4. A condition code of 12 will be set when an REXEC batch request encounters one of the following error conditions:

- The client cannot connect to TCP/IP.
- The host name cannot be resolved.
- The translation table cannot be loaded.

5. A condition code of 12 will be set when an RSH batch request encounters an error condition in which the client cannot connect to TCP/IP or when the host name cannot be resolved.

The following example shows REXEC JCL Spooling Output to JES:

```
//REXEC JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID
//STP1 EXEC PGM=REXEC,REGION=512K,
//       PARM='-l userid -p password foreign_host command'
//SYSPRINT DD SYSOUT=*  
```

**Note:** The data set containing the JCL cannot have sequence numbers.

The following example shows the use of the `userid.NETRC.DATA` containing the user ID and password. The output is sent to a permanent data set for later use.

```
//REXEC JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID
//STP1 EXEC PGM=REXEC,REGION=512K,
//       PARM='foreign_host command'
//SYSPRINT DD DSN=USERID.REXEC.SYSPRINT,DISP=(NEW,CATLG),
//       UNIT=3380,VOL=SER=MYVOL  
```

**Note:** When running REXEC in batch, the user ID assigned to the job is used as the `user_id` in the NETRC data set.

The following example shows the use of the NETRC DD statement in batch. The NETRC DD statement can be used in batch to override the default `userid.NETRC.DATA` or `userid.NETRC` files.

```
//REXEC JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID
//STP1 EXEC PGM=REXEC,REGION=512K,
//       PARM='foreign_host command'
//SYSPRINT DD SYSOUT=*  
//NETRC DD DSN=TST.REXEC.NETRC,DISP=SHR  
```

**Note:** The user ID and password are retrieved from `TST.REXEC.NETRC` instead of `userid.NETRC.DATA` or `userid.NETRC`. 

Chapter 10. Executing commands on a remote host 369
RSH command—Execute a command on a remote host and receive the results on your local host

Purpose
The Remote Shell Protocol (RSH) is a remote execution client similar to REXEC that enables you to execute a command on a remote host and receive the results on your local host.

You can use the RSH command from TSO, or as a batch job.

Note: The TSO REXEC command uses the Pascal socket API, so VMCF must be started for the command to be successful. If VMCF is not started, an ABEND0D6 can occur.

Format

```
RSH [-? -d -l user/password -s port] foreign_host command
```

Note: The parameters for the RSH command could be case sensitive, depending on the remote host you are targeting.

Parameters

- `?-` Lists the valid parameters for the RSH command.
- `-d` Turns on debug tracing.
- `-l user/password`
  Specifies the remote user ID and password.
  You must enter the slash (/) between the user ID and password if the target system is MVS. The character you must use can vary, depending on the target host. For VM hosts, you should use an @ character.
- `-s port`
  Specifies the server port. The default is 514.

*foreign_host*
  Specifies the name or IPv4 IP address of the remote host on which you are executing the command. If foreign_host is a name, it must resolve to an IPv4 address.

*command*
  Specifies the command to be executed on the remote host.
Examples

- Execute a command on a remote host and receive the results on your local host:

  ```
  rsh -l user28/user28 mvs1 lista
  READY
  ```

  ```
  SYS1.HELP
  BUILD.HELP
  GIM.SGIMCLS0
  ISR.V3R5M0.ISRCLIB
  DSN230.DSNCLIST
  DSN230.NEW.DSNCLIST
  DSN230.DSNAMACS
  USER.CLIST
  BUILD.CLIST
  SYS1.HRFCLIST
  ISP.V3R5M0.ISPEXEC
  READY
  ```

- Use JCL to submit the RSH command as a batch job:

  ```
  //USER28 JOB ,CARTER,MSGLEVEL=(1,1),NOTIFY=USER28
  //RSH31 EXEC PGM=RSH,
  //  PARM='/-d -l user28/user28 mvs1 lista'
  //SYSPRINT DD SYSOUT=* 
  //SYSTCPD DD DSN=USER28.TCP/IP.DATA,DISP=SHR 
  //SYSIN DD DUMMY
  ```

Usage

- You must enter the required parameters on the command line. The RSH command does not prompt you for missing parameters, or enable you to use the NETRC.DATA set.

- A condition code of 12 will be set when an RSH batch request encounters an error condition in which the client cannot connect to TCP/IP or when the host name cannot be resolved.
**RHOSTS.DATA data set**

The `user_id`.RHOSTS.DATA data set provides you with an alternative to specifying RSH parameters `user_id` and `password` when you invoke the RSH command.

The `user_id`.RHOSTS.DATA data set contains one or more entries. Each entry consists of two parts, a fully qualified name of a local host and a local userid associated with that local host. The local userid is case sensitive.

The user ID specified in `user_id`.RHOSTS.DATA can be either the user ID you would otherwise specify as an RSH parameter or your logon ID on your local host.

If your user ID is the same at both the local and remote hosts, use this common ID to create RHOSTS.DATA. In this case, you do not need to include the `user_id/password` parameter on the RSH command, as shown in the following example:

```
   rsh mvsone lista
```

If your user ID at the remote host is different from your user ID at the local host, use the user ID of the remote host to create RHOSTS.DATA. In this case, you can invoke the RSH command without the password, as follows:

```
   rsh -l user28 mvsone lista
```

If you do not create the RHOSTS.DATA data set on the remote host, you must specify both the user ID and the password with the RSH command, as shown in the following example:

```
   rsh -l user28/abcdef mvsone lista
```

The host names in the following examples are the official, fully qualified names of local hosts from which you want to run RSH. The user IDs are the logon IDs for those local hosts. Nicknames are not allowed.

<table>
<thead>
<tr>
<th>local.host.name</th>
<th>user_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvsthree.raleigh.ibm.com</td>
<td>user30</td>
</tr>
<tr>
<td>mvsthree.raleigh.ibm.com</td>
<td>user31</td>
</tr>
<tr>
<td>mvstfour.raleigh.ibm.com</td>
<td>user30</td>
</tr>
</tbody>
</table>

**Using remote execution clients in a z/OS UNIX environment**

z/OS UNIX Remote Execution Protocol (z/OS UNIX REXEC) is a remote execution client that you can use to execute a command on a remote host and receive the results on the local host.

You must have the z/OS UNIX REXEC daemon (orexed) running on the remote host to use z/OS UNIX REXEC to run z/OS UNIX commands. The z/OS UNIX REXEC client passes the user name, password, and command to the z/OS UNIX REXEC daemon. The daemon provides automatic logon and user authentication, depending on the parameters that you set.

You do not have to have the z/OS UNIX REXEC daemon (orexed) running on the remote host if you are not running z/OS UNIX commands. The z/OS UNIX REXEC client can use any REXEC daemon (not just z/OS UNIX).
The z/OS UNIX orexec/rexec command—Execute a command on the remote host

**Purpose**
Use the z/OS UNIX orexec/rexec command to execute a command on the remote host and receive the results on the local host.

**Notes:**
1. The rexec command is a synonym for the orexec command in the z/OS UNIX shell. The rexec command syntax is the same as that for the orexec command.
2. If the -s parameter is not used to specify the port, the port to be used by the client must be defined in the /etc/services file as an exec entry defined to TCP. For information on /etc/services, Protocol Number and Port Assignments, refer to z/OS Communications Server: IP Configuration Reference.

**Format**

```
orexec [-? | -d | -l user_id | -p password | -s port] foreign_host command
```

**Note:** Enter the orexec parameters -d, -l, -p, and -s in lowercase letters because they are case sensitive. The user_id and password parameters may be case sensitive, depending on the operating system of the remote host.

**Parameters**
- `-?` Displays the help message.
- `-d` Activates debug tracing.
- `-l user_id` Specifies the user ID on the foreign host.
- `-p password` Specifies the password for the user ID on the foreign host.
- `-s port` Specifies the TCP port number of the rexec server on the foreign host. The default is the port number defined in /etc/services.
- `-V` Writes the name and the Authorized Program Analysis Report (APAR) level to syslog.
- `-C` Changes all messages to uppercase.

**foreign_host** Specifies the name or IP address of the foreign host to which you are sending the orexec command. Specify the foreign host by its host name or IP address.

**command** Specifies the command that is sent to the foreign host. The command is composed of one or more words. Coding is assigned after checking the prefixed parameters (-l, -p, -s) and assigning the remaining string as the command. The
command you specify must not require a response from you to complete. orexec cannot interact with you after you enter data in the command format.

Examples
Use the orexec command to execute a command on a remote host:

```
orexec -l user28 -p user28 -s 512 mvs1 lista
```

SYS1.HELP
GIM.SGIMCL50
DSN230.DSNCLIST
USER.CLIST
BUILD.CLIST
SYS1.HRFCLST
USER28.ORSHD5.JOB00160.D0000103.?.

Usage
If you omit the user_id, the password, or both when entering the orexec command, the system prompts you to supply the parameters.
Chapter 11. Using the network database system

This chapter describes the Network Database System (NDB), that is used to access relational database systems in a TCP/IP environment.

The Network Database System enables:

- Workstation or mainframe users to issue Structured Query Language (SQL) statements interactively
- Database sharing between computers with different system architectures and operating systems

The NDB client command enables you to issue SQL statements that you can enter directly from the client machine to query or access data from a DB2 subsystem. You can use the NDBCLNT command from the following environments:

- DOS
- MVS
- AIX on RS/6000®
- UNIX on Sun Microsystems
- VM

The following subjects are covered in this chapter:

- “NDBCLNT command—Issue SQL statements to a DB2 subsystem” on page 376
- “SQL statements” on page 378
- “NDB commands” on page 378
- “Return codes for NDB” on page 378
- “Format of output displayed on the client” on page 381
NDBCLNT command—Issue SQL statements to a DB2 subsystem

Purpose

Use the NDBCLNT command to issue SQL statements to a DB2 subsystem.

Note: The SQL statement that you want to enter is not part of the NDBCLNT command. You enter your SQL statement after the NDBC prompt message is displayed.

Format

```
```

Parameters

- **-h host_name**
  Specifies the host on which the DB2 subsystem resides. `host_name` must be IPv4. You are prompted for a name if you do not enter one.

  **Note:** For the NDBCLNT command, substituting the IP address for `host_name` is accepted only if doing so is supported by the remote procedure call (RPC) implementation of the TCP/IP product running on the client machine.

- **-u userid**
  Specifies the host user ID to be used to access the DB2 system. You are prompted for an ID if you do not enter one.

- **-p password**
  Specifies the password for the TSO `userid`. You are prompted for a password if you do not enter one.

  **Note:** If you enter your password from the command line as part of the NDBCLNT invocation, the password is displayed. If you do not enter your password as part of the invocation, you are prompted for the password and the password is not echoed to the screen.

- **-d DB2_subsystem**
  Specifies the name of the DB2 subsystem to be accessed for this NDB session. You are prompted for a DB2 subsystem name if you do not enter one.

- **-o output_file**
  Specifies the file that stores the SQL statement result. Ensure that you use the correct naming convention for the system being used. This option is required if you specify the -s option.

- **-s**
  Prevents output from being displayed on the terminal. If you do not specify the -o option, it is assumed, and you are prompted for the name of the output file.

- **-l**
  Write listing of NDB session to output file, if one is specified. This includes the SQL statement and the outcome of that SQL statement. If no output file is specified, this option is ignored.
-c Specifies that the output from an SQL statement should be sent as one continuous stream. The default is to have only one buffer of data at a time displayed (enter the word continue to get the next buffer of data).

**Examples**

```plaintext
ndbclnt -h mvs1 -d db2a -u user1 -p user1

Port server has allocated NDB Prog: 536870945.

NDBC: create table empinfo (empno int,name char(15),salary dec(8,2),hiredate date)
SQL statement completed successfully

NDBC: insert into empinfo values (10001, 'Andersen', 23456.78,'01/02/1983')
1 rows were affected by this SQL statement

NDBC: insert into empinfo values (10204,'Jones',25890.21,'02/01/1983')
1 rows were affected by this SQL statement

NDBC: insert into empinfo values (10997,'Smith',27654.32,CURRENT DATE)
1 rows were affected by this SQL statement

NDBC: commit
SQL statement completed successfully

NDBC: select * from empinfo
10001 Andersen 23456.78 1983-01-02
10204 Jones 25890.21 1983-02-01
10997 Smith 27654.32 1994-07-25

NDBC: update empinfo set salary = salary * 1.50
3 rows were affected by this SQL statement

NDBC: select * from empinfo
10001 Andersen 35185.17 1983-01-02
10204 Jones 38835.31 1983-02-01
10997 Smith 41481.48 1994-07-25

NDBC: rollback
SQL statement completed successfully

NDBC: select * from empinfo
10001 Andersen 23456.78 1983-01-02
10204 Jones 25890.21 1983-02-01
10997 Smith 27654.32 1994-07-25

NDBC: drop table empinfo
SQL statement completed successfully

NDBC: end
```

**Usage**

- If you specify more than once any of the NDB invocation options that require a parameter (for example, -h, -u, -p, -d, -o), the last occurrence is the one used by the NDB client code.
- The NDBCLNT command format and parameter descriptions are displayed when you specify any of the following:
  - No parameters
  - No valid parameters
- After entering the NDBCLNT command, you are prompted by the NDBC: prompt for either an SQL statement or an NDB command.
- The reply buffer used by the NDB clients and servers is 8192 bytes. The size of the reply buffer can be reset by changing the value of the definition NDB_REPLY_LENGTH at the beginning of the NDBCLT.C program of the sample client code, and then recompiling the NDB client.
SQL statements

NDB accepts SQL statements up to 2048 characters long. NDB supports all SQL statements that DB2 allows to be interactively or dynamically prepared along with the SELECT statement.

If you are using NDBCLNT with an operating system that supports redirected input from a file, you must ensure that the last statement in the file is the NDB command END. Refer to the user's guide or online help facility for your workstation operating system for more information about using redirected input.

NDB commands

The NDB commands are as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>continue</td>
<td>Requests additional buffers of data when the SQL query returns more than one buffer of data.</td>
</tr>
<tr>
<td>end</td>
<td>Ends the NDB client session. END causes an implicit SQL commit work to take place. All changes made since the last commit or rollback (or since the beginning of the session if no commits or rollbacks were issued) become permanent.</td>
</tr>
<tr>
<td>exit</td>
<td>Emergency use only. Certain errors can cause the NDB client to be unable to continue processing NDB commands or SQL statements, including the NDB END command. These error conditions result in the same or other error conditions continually being reported by the NDB client. If this happens, issue the NDB emergency exit command (EEXIT) to immediately exit. Note that using (EEXIT) does not allow the DB2 thread to terminate. This will cause the database resources (for example, locks and buffers) to continue to be held. The NDB server and the port that the NDB client is currently using will also continue to be held until the address space containing this NDB server has been cancelled and restarted.</td>
</tr>
</tbody>
</table>

Return codes for NDB

The valid NDB return codes and messages are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Problem with RPC.</td>
</tr>
</tbody>
</table>

Look at the error messages displayed before this one. Refer to the Remote Procedure Call (RPC) messages and codes information for the client platform from which you are running the NDB client.

-20000 RPC error. Unable to call the server.

The NDB sample client code detected that the NDB client was unable to successfully send the SQL statement or NDB command to the NDB server and to receive a response.

NDB uses RPC to handle requests between NDB clients and servers. Look at the error messages and RPC information that was displayed before this error. Refer to the RPC messages and codes information for the client platform from which you are running the NDB client.
Invalid NDBC control block. Reason: <string>

The NDB server detected that the NDBC control block, used for communication between an NDB client and its assigned NDB Server, contained an incorrect value.

Listed below are the possible reasons why the error occurred:

- NDB version numbers do not match.
  The version number in the client code must match the version number in the server code. The NDB sample client code sets a field in the control block NDBC called `ndbver` to indicate the current version of the NDB client. The NDB server also contains a version number in its code. For TCP/IP for MVS 3.1, the version value is 2. For all previous releases of TCP/IP for MVS, the version value is 1. Since the version value for NDB changes when the NDBC control block is changed, one level (version, release and modification) of NDB for TCP/IP for MVS may not be compatible with another.

- NDB release numbers do not match.
  The release number in the client code must match the release number in the server code. The NDB sample client code sets a field in the control block NDBC called `ndbrel` to indicate the current release of the NDB client. The NDB server also contains a release number in its code. For TCP/IP for MVS 3.1 and all previous releases, the release value is 1. Since the version value for NDB changes when the NDBC control block is changed, one level (version, release and modification) of NDB for TCP/IP for MVS may remain compatible with another, but there are differences in the control block field values that affect NDB server processing.

- Eyecatcher is not for NDBC control block.
  The NDB sample client code sets a field in the control block NDBC called `ndbccb` to indicate the eyecatcher of the main control block being used by the NDB client. The NDB server looks for this eyecatcher before processing information contained in the control block. For TCP/IP for MVS 3.1 and all previous releases, the control block eyecatcher is `NDBC`.

- NDB Server name is unrecognized.
  The NDB sample client code sets a field in the control block NDBC called `ndbsname` to indicate the NDB server name by which the NDB client code knows the NDB server. The NDB server did not find a recognized name in the `ndbsname` field of the NDBC control block. For TCP/IP for MVS 3.1 and all previous releases, the NDB server name is `netdbsrv`.

Fix the NDBC control block field that is causing the error and rebuild the NDB client.

You (or PUBLIC) do not have authority to run DB2 plan EZAND320.

For NDB to access data stored in DB2, the DBUITIL2 application program that acts as interface code was stored in DB2. DBUITIL2 is in the form of a DB2 plan name EZAND320. To use NDB (which uses the EZAND320), you must be granted execute authority for EZAND320 by your database administrator. This authority can be granted to your specific user ID or to everyone by granting it to PUBLIC.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>-20102</td>
<td>DB2 Subsystem is unavailable. DB2 is not active at OPEN DB.</td>
</tr>
<tr>
<td></td>
<td>The NDB server detected that the DB2 subsystem specified at NDB server</td>
</tr>
<tr>
<td></td>
<td>startup was not running at the time the NDB server tried to establish</td>
</tr>
<tr>
<td></td>
<td>a connection with DB2. Contact your system programmer.</td>
</tr>
<tr>
<td>-20103</td>
<td>Error occurred while using DB2 CAF.</td>
</tr>
<tr>
<td></td>
<td>An error occurred when the NDB server used the DB2 Call Attachment</td>
</tr>
<tr>
<td></td>
<td>Facility (CAF) to establish a connection with DB2. Contact your system</td>
</tr>
<tr>
<td></td>
<td>programmer.</td>
</tr>
<tr>
<td>-20104</td>
<td>Error occurred while trying to allocate storage.</td>
</tr>
<tr>
<td></td>
<td>An error occurred when the NDB server allocated storage for a data</td>
</tr>
<tr>
<td></td>
<td>structure used by DB2 or for an internal buffer. Contact your system</td>
</tr>
<tr>
<td></td>
<td>programmer.</td>
</tr>
<tr>
<td>-20105</td>
<td>Single row of query data is larger than reply buffer.</td>
</tr>
<tr>
<td></td>
<td>The length of one row of data to be returned from the current query</td>
</tr>
<tr>
<td></td>
<td>was greater than the length of the NDB reply buffer (size of which is</td>
</tr>
<tr>
<td></td>
<td>specified in the NDBC field NDBRPDLN). You can take either of the</td>
</tr>
<tr>
<td></td>
<td>following actions:</td>
</tr>
<tr>
<td></td>
<td>• Retrieve fewer columns per query either by issuing multiple queries</td>
</tr>
<tr>
<td></td>
<td>or by making the query more restrictive in the number of columns</td>
</tr>
<tr>
<td></td>
<td>requested.</td>
</tr>
<tr>
<td></td>
<td>• Increase the size of the NDB reply buffer by changing the define</td>
</tr>
<tr>
<td></td>
<td>compiler directive for NDB_REPLY_LENGTH in the program NDBCLT.C</td>
</tr>
<tr>
<td></td>
<td>and rebuilding the NDB client.</td>
</tr>
<tr>
<td>-20108</td>
<td>Host userid or password not valid.</td>
</tr>
<tr>
<td></td>
<td>The NDB server has detected that the host user ID or host password did</td>
</tr>
<tr>
<td></td>
<td>not pass the security check on the host system. Enter a valid user ID</td>
</tr>
<tr>
<td></td>
<td>and password combination.</td>
</tr>
<tr>
<td>-20999</td>
<td>Should never occur error has occurred.</td>
</tr>
<tr>
<td></td>
<td>During the execution of NDB server code, a programming error occurred.</td>
</tr>
<tr>
<td></td>
<td>Contact your system programmer.</td>
</tr>
<tr>
<td>+xxx</td>
<td>Positive return codes, other than +25, are informational and warning</td>
</tr>
<tr>
<td></td>
<td>codes from DB2.</td>
</tr>
<tr>
<td>-xxx</td>
<td>Negative return codes between -001 and -999 or that are less than or</td>
</tr>
<tr>
<td></td>
<td>equal to -30000 are from DB2.</td>
</tr>
</tbody>
</table>

Refer to [z/OS Communications Server: IP and SNA Codes](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.java/zos_communication_server_ipsna_code.htm) for more information about TCP/IP return codes. Refer to [IBM Database 2™: Messages and Codes](https://www.ibm.com/support/knowledgecenter/en/SSD7IP_2.2.0/com.ibm.db2z.connectivity.jdbc.doc_z/ibm_database_2_messages_and_codes.htm) for more information about DB2 return codes.
Format of output displayed on the client

The output from an SQL query (SELECT) is presented in row and column format. The NDB Server distinguishes between a null value that is retrieved from a table in DB2 and a null value that is the result of an arithmetic error in the SQL SELECT statement (for example, division by 0). A null data value is represented by dashes (------), while a null value due to an arithmetic error is represented by slashes (/////). 

For more information on null values, refer to IBM DATABASE2: Application Programming and SQL Guide.

NDB supports the following SQL data types:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECIMAL</td>
<td>All formats supported by DB2</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Single and double precision</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Long, or fullword, integer</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>Short, or halfword, integer</td>
</tr>
<tr>
<td>CHAR</td>
<td>Fixed length, up to 254 characters</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>Variable length, up to 254 characters</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>Long variable length, over 254 characters</td>
</tr>
<tr>
<td>DATE</td>
<td>All formats supported by DB2</td>
</tr>
<tr>
<td>TIME</td>
<td>All formats supported by DB2</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>All formats supported by DB2</td>
</tr>
</tbody>
</table>

For unsupported SQL data types, values are replaced with asterisks (******) unless the value is null. If the value is null, it is represented as previously described. Refer to the IBM DATABASE 2: SQL Reference for information about SQL data types.
Appendix A. Specifying data sets and files

This appendix describes the file-naming formats for the following operating systems:

- "MVS data sets"
- "AIX and UNIX files" on page 386
- "AS/400 operating system" on page 387
- "VM files" on page 388

MVS, AIX, UNIX, AS/400, and VM operating systems. Examples of each format are provided to show how the files appear to a TCP/IP user who is logged on to the different operating systems.

**MVS data sets**

FTP subcommands can require a data set or file name. The format used to name a data set depends on the host system. Some systems limit the length of a data set name, and some systems are case sensitive.

Data set names in MVS consist of one or more names, called qualifiers, each from one to eight characters long, that are delimited from one another by periods.

The leftmost qualifier in the data set name is the **high-level qualifier**. The rightmost qualifier in the data set name is the **low-level qualifier**. Partitioned data sets may be further qualified with a member name in the rightmost position. Qualifiers lying between them are called intermediate-level qualifiers.

For example, in the data set name `dog.bulldog.winston`, `dog` is the high-level qualifier, `bulldog` is the intermediate-level qualifier, and `winston` is the low-level qualifier.

Specify TSO sequential data sets in the following format:

```
>> [hlev_qual.] [llev_qual.] [llev_qual.]
```

Specify TSO partitioned data sets in the following format:

```
>> [hlev_qual.] [llev_qual.] [membername]
```

**hlev_qual**

Specifies the high-level qualifier of the data set. The default is the current working directory. If you specify this parameter, the complete data set name must be enclosed within single quotation marks (').

**llev_qual**

Specifies the low-level qualifier of the data set. You must specify this qualifier for sequential data sets.
membername

Specifies the member name of a partitioned data set (PDS). You must include parentheses around membername only when you also specify lev_qual or hlev_qual.

Sequential data sets

A sequential data set is a single file that can be allocated with any record length specified. The naming requirements for a sequential data set on an MVS host are minimal, and most of the requirements apply to any data set name under MVS.

The naming requirements for a sequential data set are:

- No part of the name can start with a numeric.
- No part of the name can be more than 8 characters in length.
- Each part of the name is separated by a period.
- A sequential data set name can have a minimum of 2 and a maximum of 44 characters.
- If single quotation marks (') are not used when specifying the data set name, the MVS system appends the current working directory as the first part of the name.

The following examples show the naming conventions for sequential data sets on an MVS host.

To access the sequential data set KC00852.SEQ.NAMES, the user, with the current working directory KC00852, enters one of the following:

- 'KC00852.SEQ.NAMES'
- SEQ.NAMES

Either of these formats is acceptable for accessing a sequential data set.

Partitioned data sets

A partitioned data set (PDS) is a group of files contained in a library. The individual files that make up a PDS are called members. You can access an entire PDS or any individual member of a PDS.

The naming requirements for a partitioned data set are:

- No part of the name can start with a numeric.
- No part of the name can be more than eight characters in length.
- Each part of the name is separated by a period.
- If single quotation marks are not used when specifying the PDS name, the MVS system appends the current working directory as the first part of the name.

The difference between a sequential and partitioned data set specification is that the partitioned data set user accesses the directory of members in the PDS, and the sequential data set user accesses an individual file.

The following examples show the naming conventions for partitioned data sets on an MVS host.

To access the partitioned data set KC00852.PDS.NAMES, the user, with the current working directory 'KC00852', enters one of the following:

- 'KC00852.PDS.NAMES'
- PDS.NAMES
Either of these formats is acceptable to access a partitioned data set.

**Note:** You can use the special character asterisk (*) as a global name character (wildcard) for pattern matching when you specify a data set name, with the following restrictions:

- The asterisk must be the last, or only, character specified for a level of qualifier.
- When the data set name is enclosed in quotation marks, you cannot use the asterisk as a wildcard in the high-level qualifier of the data set name. Data set names not enclosed in quotation marks will use the setting of the current directory as the high-level qualifier.
- You can use the asterisk more than once in the complete data set name, but the asterisk must be the last character for each level of data set name qualifier.
- If you specify a member name, you cannot use an asterisk anywhere in the data set name.
- If you use an asterisk as all or part of the member name, you cannot use an asterisk anywhere else in the data set name.

To access an individual member of a PDS, the member name is entered in parentheses.

To access the member PROPER in the PDS KC00852.PDS.NAMES, the user, with the current working directory KC00852, enters one of the following:

- ‘KC00852.PDS.NAMES(PROPER)’
- ‘PDS.NAMES(PROPER)’

Either of these formats is acceptable to access an individual member of a partitioned data set.

**Transferring data between partitioned and sequential data sets**

When transferring data between partitioned and sequential data sets, ensure that the `local_file` and `foreign_file` parameters of the FTP subcommands are compatible with the type of data set you are transferring to or from. For example, if your local working directory is a partitioned data set and you want to GET the sequential file TEST.FILE1, you cannot issue the subcommand GET TEST.FILE1 to retrieve the file, because this subcommand by default tries to use the local PDS member name TEST.FILE1, which is not a valid member name.

To keep the local and remote file names compatible with the type of data set used, do one of the following:

- Specify both the local and remote file names

  For example:

  ```
  GET TEST.FILE TESTFL1
  PUT TESTPDS(FILE1) FILE1
  ```

- Change the directory to the lowest level qualifier.

  For example, to transfer between the PDS ‘USER14.TESTPDS(NAME1)’ and the sequential data set ‘USER17.SEQ.NAME1’, do the following:

  1. Change the local directory (LCD) to ‘USER14.TESTPDS’
  2. Change the directory (CD) to ‘USER17.SEQ’
  3. Enter one of the following:

     - `GET NAME1` to get ‘USER17.SEQ.NAME1’ as ‘USER14.TESTPDS(NAME1)’
Data transfer methods

You must use the appropriate transmission attributes to preserve the content and structure of the data when you transfer data sets or files between two hosts. Use the FTP MODE subcommand to specify how the bits of data are to be transmitted, and the FTP TYPE subcommand to define the way that data is represented during the data transfer.

TCP/IP supports only the data transfer of a data set or file structured as a continuous sequence of data bytes. This ensures that the correct record format is preserved across MVS hosts.

Table 25 shows how to set the transmission attributes for different host systems.

<table>
<thead>
<tr>
<th>Transfer between host types</th>
<th>Transfer type</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBCDIC to EBCDIC—text data</td>
<td>EBCDIC</td>
<td>Stream</td>
</tr>
<tr>
<td>EBCDIC to EBCDIC—binary data</td>
<td>EBCDIC</td>
<td>Block</td>
</tr>
<tr>
<td>EBCDIC to ASCII—text data</td>
<td>ASCII</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII to EBCDIC—text data</td>
<td>ASCII</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII to EBCDIC—binary data</td>
<td>Image (binary)</td>
<td>Stream</td>
</tr>
<tr>
<td>ASCII to EBCDIC to ASCII—all data</td>
<td>Image (binary)</td>
<td>Stream</td>
</tr>
</tbody>
</table>

Note: The EBCDIC host is used for storage only. Data is not used on the EBCDIC host.

Transferring PDS directory information

When a PDS member is transmitted in block or compressed data transfer mode with a representation type of EBCDIC, the user data associated with the PDS member is also transferred to the directory on the target host. This transfer occurs only when using an MVS client.

AIX and UNIX files

For the Advanced Interactive Executive (AIX) and UNIX operating systems, data is stored in files. Related files are stored in a directory. z/OS CS files are UNIX files.

Specify AIX and UNIX files in the following format:

```bash
(directory)/filename
```
directory
Specifies a directory name. Directories contain the names of files, other
directories, or both.

filename
Specifies a file name. It can be up to 14 characters long.

The complete name of an AIX and UNIX file contains the directory name and the
file name. The following is an example:
/mailfiles/cooks

Where:
mailfiles
The directory name
cooks The file name

In the AIX and UNIX operating systems, you specify the first slash (/) only when you
begin at the root directory. If you are specifying a file in the current directory, enter
only the file name. For example, if you are in the current directory mailfiles and you
want to access the cooks file, specify:
cooks

The directory name and file name can each be up to 14 characters in length. The
AIX and UNIX operating systems distinguishes between uppercase and lowercase
letters in file names.

A directory name and file name should not include characters such as backslash (\),
ampersand (&), and period (.) that have a special meaning to the shell.

AS/400 operating system

For the AS/400 operating system, data is stored in files.

Specify AS/400 files in the following format:

library/file.member

library
A library name. Libraries contain the names of programs, files, and commands.

file.member
The file name.

In the AS/400 operating system, files can have one or more members. Each file can
consist of data records, source programs, or database definitions.

The FTP subcommand PUT is used to copy a local file member into a file at the
remote host. The following is an example:
PUT PDS.DATA(MBR1) LIB1/FILEA.MBR1

In this example, the PUT subcommand copies the file member MBR1 in file FILEA
into library TCPA at the local host to MBR1 in FILEA in library TCPB at the remote host.
If the member already exists at the remote host, it is overwritten.
VM files

Data is stored in files on VM hosts. Specify VM files in the following format:

```
filename.filetype
```

*filename*  
Specifies the file name.

*filetype*  
Specifies the file type.

**Note:** The file mode is not accepted by foreign VM hosts; it is taken to be the file mode associated with the current working directory. The file mode is not used in TCP/IP commands.

For example, if you want to specify a file named `accounts` with a file type `cprog`, enter the following:

```
accounts.cprog
```

Where *filename* is `accounts` and *filetype* is `cprog`.

All VM file specifications are treated as if they are entered in uppercase. The file name and the file type consist of 1 to 8 alphanumeric characters. Other valid characters are $, #, @ (at character), + (plus), - (hyphen), and _ (underscore).

You can use the special character * (asterisk) for pattern matching.
Appendix B. Mapping values for the APL2 character set

This appendix lists the GDDMXD/MVS default mapping values for the APL2 character set. However, if the hlq.GDXAPLCS.MAP data set exists, the default mapping values are overridden.

Each entry in the hlq.GDXAPLCS.MAP data set (alternative character set) contains the mapping for a particular physical key that corresponds to three characters. The characters correspond to the physical key by:

- Pressing the key alone
- Pressing the key and the Shift key simultaneously
- Pressing the key and the Alt key simultaneously

The hlq.GDXAPLCS.MAP data set entries must contain the following seven single-byte hexadecimal values entered as EBCDIC characters:

- Value 1 is the hexadecimal keycode for the physical key.
- Values 2, 4, and 6 identify whether the character is in the primary or alternative character set for the emulated 3179G. If the character is in the primary set, the value is 0; if the character is in the alternative set, the value is 8.
- Values 3, 5, and 7 specify the EBCDIC code of the character in the character set.

The combination of values 2 and 3 define the bytes that describe the character when the key corresponding to the keycode is pressed alone.

The combination of values 4 and 5 define the bytes that describe the character when the key corresponding to the keycode and the Shift key are pressed simultaneously.

The combination of values 6 and 7 define the bytes that describe the character when the key corresponding to the keycode and the Alt key are pressed simultaneously.

Table 26 lists the mapping values for the APL2 character set.

Table 26. Mapping values for the APL2 character set

<table>
<thead>
<tr>
<th>Character name</th>
<th>Character set value</th>
<th>EBCDIC value</th>
<th>Default keycode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad Jot</td>
<td>8</td>
<td>73</td>
<td>9 + Shift</td>
</tr>
<tr>
<td>Quad Slope</td>
<td>8</td>
<td>CE</td>
<td>9 + Alt</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>F1</td>
<td>A</td>
</tr>
<tr>
<td>Diaeresis</td>
<td>8</td>
<td>72</td>
<td>A + Shift</td>
</tr>
<tr>
<td>Down Tack Up Tack</td>
<td>8</td>
<td>DA</td>
<td>A + Alt</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>F2</td>
<td>D</td>
</tr>
<tr>
<td>Overbar</td>
<td>8</td>
<td>A0</td>
<td>B + Shift</td>
</tr>
<tr>
<td>Del Tilde</td>
<td>8</td>
<td>FB</td>
<td>B + Alt</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>F3</td>
<td>C</td>
</tr>
<tr>
<td>&lt;</td>
<td>0</td>
<td>4C</td>
<td>C + Shift</td>
</tr>
<tr>
<td>Del Stile</td>
<td>8</td>
<td>DC</td>
<td>C + Alt</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>F4</td>
<td>D</td>
</tr>
<tr>
<td>Character name</td>
<td>Character set value</td>
<td>EBCDIC value</td>
<td>Default keycode</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Not Greater</td>
<td>8</td>
<td>8C</td>
<td>0 + Shift</td>
</tr>
<tr>
<td>Delta Stile</td>
<td>8</td>
<td>DD</td>
<td>0 + Alt</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>F5</td>
<td>E</td>
</tr>
<tr>
<td>=</td>
<td>0</td>
<td>7E</td>
<td>E + Shift</td>
</tr>
<tr>
<td>Circle Stile</td>
<td>8</td>
<td>CD</td>
<td>E + Alt</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>F6</td>
<td>F</td>
</tr>
<tr>
<td>Not Less</td>
<td>8</td>
<td>AE</td>
<td>F + Shift</td>
</tr>
<tr>
<td>Circle Slope</td>
<td>8</td>
<td>CF</td>
<td>F + Alt</td>
</tr>
<tr>
<td>&gt;</td>
<td>0</td>
<td>6E</td>
<td>10 + Shift</td>
</tr>
<tr>
<td>Circle Bar</td>
<td>8</td>
<td>ED</td>
<td>10 + Alt</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>F8</td>
<td>11</td>
</tr>
<tr>
<td>Not Equal</td>
<td>8</td>
<td>BE</td>
<td>11 + Shift</td>
</tr>
<tr>
<td>Circle Star</td>
<td>8</td>
<td>FD</td>
<td>11 + Alt</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>F9</td>
<td>12</td>
</tr>
<tr>
<td>Down Caret</td>
<td>8</td>
<td>78</td>
<td>12 + Shift</td>
</tr>
<tr>
<td>Down Caret Tilde</td>
<td>8</td>
<td>CB</td>
<td>12 + Alt</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>F0</td>
<td>13</td>
</tr>
<tr>
<td>Up Caret</td>
<td>8</td>
<td>71</td>
<td>13 + Shift</td>
</tr>
<tr>
<td>Up Caret Tilde</td>
<td>8</td>
<td>CA</td>
<td>13 + Alt</td>
</tr>
<tr>
<td>+</td>
<td>0</td>
<td>4E</td>
<td>14</td>
</tr>
<tr>
<td>-</td>
<td>0</td>
<td>60</td>
<td>14 + Shift</td>
</tr>
<tr>
<td>!</td>
<td>8</td>
<td>DB</td>
<td>14 + Alt</td>
</tr>
<tr>
<td>Times</td>
<td>8</td>
<td>B6</td>
<td>15</td>
</tr>
<tr>
<td>Divide</td>
<td>8</td>
<td>B8</td>
<td>15 + Shift</td>
</tr>
<tr>
<td>Quad Divide</td>
<td>8</td>
<td>EE</td>
<td>15 + Alt</td>
</tr>
<tr>
<td>Q</td>
<td>0</td>
<td>D8</td>
<td>19</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>6F</td>
<td>19 + Shift</td>
</tr>
<tr>
<td>Q Underbar</td>
<td>8</td>
<td>58</td>
<td>19 + Alt</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
<td>E6</td>
<td>1A</td>
</tr>
<tr>
<td>Omega</td>
<td>8</td>
<td>B4</td>
<td>1A + Shift</td>
</tr>
<tr>
<td>W Underbar</td>
<td>8</td>
<td>66</td>
<td>1A + Alt</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>C5</td>
<td>1B</td>
</tr>
<tr>
<td>Epsilon</td>
<td>8</td>
<td>B1</td>
<td>1B + Shift</td>
</tr>
<tr>
<td>E Underbar</td>
<td>8</td>
<td>45</td>
<td>1B + Alt</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
<td>D9</td>
<td>1C</td>
</tr>
<tr>
<td>Rho</td>
<td>8</td>
<td>B3</td>
<td>1C + Shift</td>
</tr>
<tr>
<td>R Underbar</td>
<td>8</td>
<td>59</td>
<td>1C + Alt</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>E3</td>
<td>1D</td>
</tr>
<tr>
<td>Character name</td>
<td>Character set value</td>
<td>EBCDIC value</td>
<td>Default keycode</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Tilde</td>
<td>8</td>
<td>80</td>
<td>1D + Shift</td>
</tr>
<tr>
<td>T Underbar</td>
<td>8</td>
<td>63</td>
<td>1D + Alt</td>
</tr>
<tr>
<td>Y</td>
<td>0</td>
<td>E8</td>
<td>1E</td>
</tr>
<tr>
<td>Up Arrow</td>
<td>8</td>
<td>8A</td>
<td>1E + Shift</td>
</tr>
<tr>
<td>Y Underbar</td>
<td>8</td>
<td>68</td>
<td>1E + Alt</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>E4</td>
<td>1F</td>
</tr>
<tr>
<td>Down Arrow</td>
<td>8</td>
<td>8B</td>
<td>1F + Shift</td>
</tr>
<tr>
<td>U Underbar</td>
<td>8</td>
<td>64</td>
<td>1F + Alt</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>C9</td>
<td>20</td>
</tr>
<tr>
<td>Iota</td>
<td>8</td>
<td>B2</td>
<td>20 + Shift</td>
</tr>
<tr>
<td>I Underbar</td>
<td>8</td>
<td>49</td>
<td>20 + Alt</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>D6</td>
<td>21</td>
</tr>
<tr>
<td>Circle</td>
<td>8</td>
<td>9D</td>
<td>21 + Shift</td>
</tr>
<tr>
<td>O Underbar</td>
<td>8</td>
<td>56</td>
<td>21 + Alt</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>D7</td>
<td>22</td>
</tr>
<tr>
<td>Star</td>
<td>0</td>
<td>5C</td>
<td>22 + Shift</td>
</tr>
<tr>
<td>P Underbar</td>
<td>8</td>
<td>57</td>
<td>22 + Alt</td>
</tr>
<tr>
<td>Left Arrow</td>
<td>8</td>
<td>9F</td>
<td>23</td>
</tr>
<tr>
<td>Right Arrow</td>
<td>8</td>
<td>8F</td>
<td>23 + Shift</td>
</tr>
<tr>
<td>Quad Quote</td>
<td>8</td>
<td>DE</td>
<td>23 + Alt</td>
</tr>
<tr>
<td>Left Brk Right Brk</td>
<td>8</td>
<td>CC</td>
<td>24</td>
</tr>
<tr>
<td>Iota Underbar</td>
<td>8</td>
<td>74</td>
<td>24 + Shift</td>
</tr>
<tr>
<td>Delta Underbar</td>
<td>8</td>
<td>FC</td>
<td>24 + Alt</td>
</tr>
<tr>
<td>Equal Underbar</td>
<td>8</td>
<td>E1</td>
<td>25</td>
</tr>
<tr>
<td>Epsilon Underbar</td>
<td>8</td>
<td>E1</td>
<td>25 + Shift</td>
</tr>
<tr>
<td>Diaeresis Dot</td>
<td>8</td>
<td>75</td>
<td>25 + Alt</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>C1</td>
<td>27</td>
</tr>
<tr>
<td>Alpha</td>
<td>8</td>
<td>B0</td>
<td>27 + Shift</td>
</tr>
<tr>
<td>A Underbar</td>
<td>8</td>
<td>41</td>
<td>27 + Alt</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
<td>E2</td>
<td>28</td>
</tr>
<tr>
<td>Up Stile</td>
<td>8</td>
<td>8D</td>
<td>28 + Shift</td>
</tr>
<tr>
<td>S Underbar</td>
<td>8</td>
<td>62</td>
<td>28 + Alt</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>C4</td>
<td>29</td>
</tr>
<tr>
<td>Down Stile</td>
<td>8</td>
<td>8E</td>
<td>29 + Shift</td>
</tr>
<tr>
<td>D Underbar</td>
<td>8</td>
<td>44</td>
<td>29 + Alt</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>C6</td>
<td>2A</td>
</tr>
<tr>
<td>Underbar</td>
<td>0</td>
<td>6D</td>
<td>2A + Shift</td>
</tr>
<tr>
<td>F Underbar</td>
<td>8</td>
<td>46</td>
<td>2A + Alt</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>C7</td>
<td>2B</td>
</tr>
</tbody>
</table>
Table 26. Mapping values for the APL2 character set (continued)

<table>
<thead>
<tr>
<th>Character name</th>
<th>Character set value</th>
<th>EBCDIC value</th>
<th>Default keycode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Del</td>
<td>8</td>
<td>BA</td>
<td>2B + Shift</td>
</tr>
<tr>
<td>G Underbar</td>
<td>8</td>
<td>47</td>
<td>2B + Alt</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>C8</td>
<td>2C</td>
</tr>
<tr>
<td>Delta</td>
<td>8</td>
<td>BB</td>
<td>2C + Shift</td>
</tr>
<tr>
<td>H Underbar</td>
<td>8</td>
<td>48</td>
<td>2C + Alt</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>D1</td>
<td>2D</td>
</tr>
<tr>
<td>Jot</td>
<td>8</td>
<td>AF</td>
<td>2D + Shift</td>
</tr>
<tr>
<td>J Underbar</td>
<td>8</td>
<td>51</td>
<td>2D + Alt</td>
</tr>
<tr>
<td>K</td>
<td>0</td>
<td>D2</td>
<td>2E</td>
</tr>
<tr>
<td>Quote</td>
<td>0</td>
<td>7D</td>
<td>2E + Shift</td>
</tr>
<tr>
<td>K Underbar</td>
<td>8</td>
<td>52</td>
<td>2E + Alt</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>D3</td>
<td>2F</td>
</tr>
<tr>
<td>Quad</td>
<td>8</td>
<td>90</td>
<td>2F + Shift</td>
</tr>
<tr>
<td>L Underbar</td>
<td>8</td>
<td>53</td>
<td>2F + Alt</td>
</tr>
<tr>
<td>Left Bracket</td>
<td>8</td>
<td>AD</td>
<td>30</td>
</tr>
<tr>
<td>(</td>
<td>0</td>
<td>4D</td>
<td>30 + Shift</td>
</tr>
<tr>
<td>Down Tack Jot</td>
<td>8</td>
<td>FE</td>
<td>30 + Alt</td>
</tr>
<tr>
<td>Right Bracket</td>
<td>8</td>
<td>BD</td>
<td>31</td>
</tr>
<tr>
<td>)</td>
<td>0</td>
<td>5D</td>
<td>31 + Shift</td>
</tr>
<tr>
<td>Up Tack Jot</td>
<td>8</td>
<td>EF</td>
<td>31 + Alt</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>E9</td>
<td>36</td>
</tr>
<tr>
<td>Left Shoe</td>
<td>8</td>
<td>9B</td>
<td>36 + Shift</td>
</tr>
<tr>
<td>Z Underbar</td>
<td>8</td>
<td>69</td>
<td>36 + Alt</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>E7</td>
<td>37</td>
</tr>
<tr>
<td>Right Shoe</td>
<td>8</td>
<td>9A</td>
<td>37 + Shift</td>
</tr>
<tr>
<td>X Underbar</td>
<td>8</td>
<td>67</td>
<td>37 + Alt</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>C3</td>
<td>38</td>
</tr>
<tr>
<td>Up Shoe</td>
<td>8</td>
<td>AA</td>
<td>38 + Shift</td>
</tr>
<tr>
<td>C Underbar</td>
<td>8</td>
<td>43</td>
<td>38 + Alt</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>E5</td>
<td>39</td>
</tr>
<tr>
<td>Down Shoe</td>
<td>8</td>
<td>AB</td>
<td>39 + Shift</td>
</tr>
<tr>
<td>V Underbar</td>
<td>8</td>
<td>65</td>
<td>39 + Alt</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>C2</td>
<td>3A</td>
</tr>
<tr>
<td>Down Tack</td>
<td>8</td>
<td>AC</td>
<td>3A + Shift</td>
</tr>
<tr>
<td>B Underbar</td>
<td>8</td>
<td>42</td>
<td>3A + Alt</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>D5</td>
<td>3B</td>
</tr>
<tr>
<td>Up Tack</td>
<td>8</td>
<td>BC</td>
<td>3B + Shift</td>
</tr>
<tr>
<td>N Underbar</td>
<td>8</td>
<td>55</td>
<td>3B + Alt</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
<td>D4</td>
<td>3C</td>
</tr>
<tr>
<td>Character name</td>
<td>Character set value</td>
<td>EBCDIC value</td>
<td>Default keycode</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Stile</td>
<td>0</td>
<td>4F</td>
<td>3C + Shift</td>
</tr>
<tr>
<td>M Underbar</td>
<td>8</td>
<td>54</td>
<td>3C + Alt</td>
</tr>
<tr>
<td>,</td>
<td>0</td>
<td>6B</td>
<td>3D</td>
</tr>
<tr>
<td>;</td>
<td>0</td>
<td>5E</td>
<td>3D + Shift</td>
</tr>
<tr>
<td>Up Shoe Jot</td>
<td>8</td>
<td>DF</td>
<td>3D + Alt</td>
</tr>
<tr>
<td>period</td>
<td>0</td>
<td>4B</td>
<td>3E</td>
</tr>
<tr>
<td>:</td>
<td>0</td>
<td>7A</td>
<td>3E + Shift</td>
</tr>
<tr>
<td>Slope Bar</td>
<td>8</td>
<td>EB</td>
<td>3E + Alt</td>
</tr>
<tr>
<td>/</td>
<td>0</td>
<td>61</td>
<td>3F</td>
</tr>
<tr>
<td>\</td>
<td>0</td>
<td>E0</td>
<td>3F + Shift</td>
</tr>
<tr>
<td>Slash Bar</td>
<td>8</td>
<td>EA</td>
<td>3F + Alt</td>
</tr>
<tr>
<td>Space</td>
<td>0</td>
<td>40</td>
<td>45</td>
</tr>
</tbody>
</table>
Appendix C. TELNET extensions

This appendix describes the Telnet 3270 DBCS Transform special operations. The following sections are included:

- "Character set cross reference table"
- "Special key operation for TELNET" on page 397
- "Operation of PF and PA keys" on page 399
- "Sense codes for special key operation" on page 400

Character set cross reference table

Table 27 describes the language, codefiles, and character sets for the Telnet 3270 DBCS Transform extended language support.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Codefile</th>
<th>Description</th>
<th>Character set</th>
<th>Description</th>
<th>CCSID¹</th>
<th>CPID²</th>
</tr>
</thead>
<tbody>
<tr>
<td>KANJI</td>
<td></td>
<td></td>
<td>JIS87KJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J8EETA</td>
<td>J8EATE</td>
<td>JIS 8 Bit English SBCS</td>
<td>ASCII</td>
<td>JIS X0201 8 Bit</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J8KETA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J8KATE</td>
<td>J8KATE</td>
<td>JIS 8 Bit Katakana SBCS</td>
<td>ASCII</td>
<td>JIS X0201 8 Bit</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JIS78ETA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIS78ATE</td>
<td>JIS78ATE</td>
<td>JIS 1978 Kanji DBCS</td>
<td>ASCII</td>
<td>JIS X0208 1978</td>
<td>00955</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JIS83KJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J8EETA</td>
<td>J8EATE</td>
<td>JIS 8 Bit English SBCS</td>
<td>ASCII</td>
<td>JIS X0201 8 Bit</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J8KETA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J8KATE</td>
<td>J8KATE</td>
<td>JIS 8 Bit Katakana SBCS</td>
<td>ASCII</td>
<td>JIS X0201 8 Bit</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JIS83ETA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIS83ATE</td>
<td>JIS83ATE</td>
<td>JIS 1983 Kanji DBCS</td>
<td>ASCII</td>
<td>JIS X0208 1990</td>
<td>00952</td>
</tr>
<tr>
<td>SJISKANJI</td>
<td>A8EETA</td>
<td>A8EATE</td>
<td>8 Bit English SBCS ASCII</td>
<td>ASCII</td>
<td>ISO/ANSI Multilingual</td>
<td>00819</td>
</tr>
<tr>
<td></td>
<td>A8KETA</td>
<td>A8KATE</td>
<td>8 Bit Katakana SBCS ASCII</td>
<td>ASCII</td>
<td>ISO/ANSI Multilingual</td>
<td>00819</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SJISETA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SJISATE</td>
<td>SJISATE</td>
<td>SJIS 1978 Kanji DBCS from EZAKJLAT</td>
<td>ASCII</td>
<td>SJIS 0941</td>
<td>0300</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SJISETA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SJISATE</td>
<td>SJISATE</td>
<td>SJIS 1995 Kanji DBCS from EZAKJ941</td>
<td>ASCII</td>
<td>SJIS 0941</td>
<td>0300</td>
</tr>
<tr>
<td>DECKANJI</td>
<td>SJDECEETA</td>
<td>SJDECEATE</td>
<td>DEC English SBCS ASCII</td>
<td>ASCII</td>
<td>none</td>
<td>none</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1990, 2002
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Codefiles</th>
<th>Description</th>
<th>Character set</th>
<th>Description</th>
<th>CCSID¹</th>
<th>CPGID²</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJDCKETA</td>
<td>SJDCKATE</td>
<td>DEC Katakana SBCS</td>
<td>ASCII</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>JDECETA</td>
<td>JDECATE</td>
<td>DEC Kanji DBCS</td>
<td>ASCII</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>EUCKANJI</td>
<td>SJECEETA</td>
<td>8 Bit English SBCS</td>
<td>ASCII</td>
<td>JIS X0201 8 Bit</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>SJECKETA</td>
<td>SJECKATE</td>
<td>8 Bit Katakana SBCS</td>
<td>ASCII</td>
<td>JIS X0201 8 Bit</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>JEUCETA</td>
<td>JEUCATE</td>
<td>Japanese Extended Unix DBCS</td>
<td>ASCII</td>
<td>JIS X0208 1990</td>
<td>00952</td>
<td>00952</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EBCDIC</td>
<td>Japanese Katakana Host SBCS</td>
<td>00290</td>
<td>00290</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HANGEUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KSC5601</td>
<td>SKSHETA</td>
<td>Korean Standard Code KSC 5601 SBCS</td>
<td>ASCII</td>
<td>KSC 5601 SBCS</td>
<td>01088</td>
<td>01088</td>
</tr>
<tr>
<td></td>
<td>SKSHATE</td>
<td></td>
<td>EBCDIC</td>
<td>Korean Host SBCS</td>
<td>00833</td>
<td>00833</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HANGEUL</td>
<td>SHANETA</td>
<td>Hangeul SBCS</td>
<td>ASCII</td>
<td>Korean PC SBCS</td>
<td>00891</td>
<td>00891</td>
</tr>
<tr>
<td></td>
<td>SHANATE</td>
<td></td>
<td>EBCDIC</td>
<td>Korean Host SBCS</td>
<td>00833</td>
<td>00833</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCHINESE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCHINESE</td>
<td>STCHETA</td>
<td>Traditional Chinese SBCS</td>
<td>ASCII</td>
<td>T-Chinese PC SBCS</td>
<td>00904</td>
<td>00904</td>
</tr>
<tr>
<td></td>
<td>STCHATE</td>
<td></td>
<td>EBCDIC</td>
<td>CECP Host SBCS</td>
<td>00037</td>
<td>00037</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCHINESE</td>
<td>TBKETA</td>
<td>Traditional Chinese DBCS</td>
<td>ASCII</td>
<td>T-Chinese PC DBCS</td>
<td>00927</td>
<td>00927</td>
</tr>
<tr>
<td></td>
<td>TBAHATE</td>
<td></td>
<td>EBCDIC</td>
<td>T-Chinese Host DBCS</td>
<td>00835</td>
<td>00835</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIG5</td>
<td>SBG5ETA</td>
<td>Big-5 Chinese SBCS I</td>
<td>ASCII</td>
<td>Big-5 Chinese PC SBCS</td>
<td>01114</td>
<td>01114</td>
</tr>
<tr>
<td></td>
<td>SBG5ATE</td>
<td></td>
<td>EBCDIC</td>
<td>CECP Host SBCS</td>
<td>00037</td>
<td>00037</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIG5</td>
<td>BG5ETA</td>
<td>Big-5 Chinese DBCS</td>
<td>ASCII</td>
<td>Big-5 PC DBCS</td>
<td>00947</td>
<td>00947</td>
</tr>
<tr>
<td></td>
<td>BG5ATE</td>
<td></td>
<td>EBCDIC</td>
<td>T-Chinese Host DBCS</td>
<td>00835</td>
<td>00835</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCHINESE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCHINESE</td>
<td>SSCHETA</td>
<td>Simplified Chinese SBCS</td>
<td>ASCII</td>
<td>S-Chinese PC SBCS</td>
<td>01115</td>
<td>01115</td>
</tr>
<tr>
<td></td>
<td>SSCHATE</td>
<td></td>
<td>EBCDIC</td>
<td>S-Chinese Host SBCS</td>
<td>00836</td>
<td>00836</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCHINESE</td>
<td>SCHETA</td>
<td>Simplified Chinese DBCS</td>
<td>ASCII</td>
<td>S-Chinese PC DBCS</td>
<td>01380</td>
<td>01380</td>
</tr>
<tr>
<td></td>
<td>SCHATE</td>
<td></td>
<td>EBCDIC</td>
<td>S-Chinese Host DBCS</td>
<td>00837</td>
<td>00837</td>
</tr>
</tbody>
</table>

¹ Coded Character Set ID  
² Code Page Group ID
Special key operation for TELNET

Table 28 contains information about the operation of special keys for the TELNET function when you are using a terminal that is not part of the 3270 family.

Use these key combinations if you are using TELNET and your terminal does not have the key that you want to use. For example, if you want to clear your screen and your terminal does not have an ERASE INPUT key, press CTRL+Y to get the same result.

To use the TELNET extensions, TCP/IP must be configured using the DBCSTRANSFORM option. For more information, refer to the z/OS Communications Server: IP Configuration Reference.

Table 28. Special key conversions

<table>
<thead>
<tr>
<th>Function name</th>
<th>Input keys</th>
<th>Function description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate</td>
<td>Ctrl+D</td>
<td>Press CTRL and D together. This combination enters the Duplicate control code in the screen buffer and a TAB is performed on the screen.</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Ctrl+K</td>
<td>Press CTRL and K together. This combination enters the Field Mark control code in the screen buffer and displays it as a blank on the screen.</td>
</tr>
<tr>
<td>Redisplay</td>
<td>Ctrl+V</td>
<td>Press CTRL and V together. This combination redistributes the contents of the screen buffer on your screen.</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Ctrl+Y</td>
<td>Press CTRL and Y together. This combination erases all characters in the unprotected fields on the screen and replaces them with blanks. The cursor is placed at the first unprotected character position on the screen.</td>
</tr>
<tr>
<td>Erase EOF</td>
<td>Ctrl+X</td>
<td>Press CTRL and X together. This combination erases all characters in an unprotected field from the cursor position to the end of the field and replaces them with blanks. If the cursor is on a protected field, the screen is inhibited and no characters are erased.</td>
</tr>
<tr>
<td>Delete One Character</td>
<td>Del</td>
<td>The DEL deletes the character at the cursor position, if the field is unprotected. The cursor does not move. All characters in the unprotected field to the right of the cursor are shifted one position to the left and blank characters are added at the end of the field.</td>
</tr>
<tr>
<td>Alphanumeric or Alphanumeric-Kana</td>
<td>Ctrl+B</td>
<td>Press CTRL and B together. This combination is a toggle switch that redisplays the screen by switching between Alphanumeric and Alphanumeric-Kana mode.</td>
</tr>
</tbody>
</table>
Table 28. Special key conversions (continued)

<table>
<thead>
<tr>
<th>Function name</th>
<th>Input keys</th>
<th>Function description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Forward Tab</td>
<td>Ctrl+F</td>
<td>Press CTRL and F together. This combination moves the cursor to the first character position in the next unprotected field. If the screen is unformatted or there are no unprotected fields on the screen, the cursor is placed in the first position on the screen.</td>
</tr>
<tr>
<td>Field Backward Tab</td>
<td>Ctrl+A, Ctrl+F</td>
<td>Press CTRL and A together then press F, or Press CTRL and A together then press CTRL and F together. This combination moves the cursor to the first character position in the previous unprotected field. If the screen is unformatted or there are no unprotected fields on the screen, the cursor is placed in the first position on the screen.</td>
</tr>
<tr>
<td>Home</td>
<td>CSI P, ESC [ P</td>
<td>Press CSI then press P, or Press ESC then press [ then press P. This combination moves the cursor to the first character position in the first unprotected field on the screen. If the screen is unformatted or there are no unprotected fields on the screen, the cursor is placed in the first character position on the screen.</td>
</tr>
<tr>
<td>Move Cursor Up</td>
<td>CSI A, ESC [ A</td>
<td>Press CSI then press A, or press ESC then press [ then press A. This combination moves the cursor up one line in the same column. If the cursor is on the first line on the screen, it moves to the last line on the screen.</td>
</tr>
<tr>
<td>Move Cursor Down</td>
<td>CSI B, ESC [ B</td>
<td>Press CSI then press B, or press ESC then press [ then press B. This combination moves the cursor down one line in the same column. If the cursor is on the last line on the screen, it moves to the first line on the screen.</td>
</tr>
<tr>
<td>Move Cursor Right</td>
<td>CSI C, ESC [ C</td>
<td>Press CSI then press C, or press ESC then press [ then press C. This combination moves the cursor one character to the right. If the cursor is in the last column in a line, it moves to the first position in the next line on the screen. If the cursor is in the last position on the screen, it moves to the first position on the screen.</td>
</tr>
<tr>
<td>Move Cursor Left</td>
<td>CSI D, ESC [ D</td>
<td>Press CSI then press D, or press ESC then press [ then press D. This combination moves the cursor one character to the left. If the cursor is in the first column in a line, it moves to the last position in the previous line on the screen. If the cursor is in the first position on the screen, it moves to the last position on the screen.</td>
</tr>
</tbody>
</table>
Table 28. Special key conversions (continued)

<table>
<thead>
<tr>
<th>Function name</th>
<th>Input keys</th>
<th>Function description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backspace One</td>
<td>Ctrl+H</td>
<td>Press CTRL and H together. This combination deletes one character before the cursor position in an unprotected field. The cursor moves one position to the left and all characters in the field shift one position to the left. If the cursor is on a protected field the screen is inhibited.</td>
</tr>
<tr>
<td>Character</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td>Ctrl+R</td>
<td>Press CTRL and R together. This combination releases the screen inhibit condition. When the screen is inhibited, only the RESET and MASTER RESET key combinations remain active.</td>
</tr>
<tr>
<td>Master Reset</td>
<td>Ctrl+A M, or Ctrl+A M, or Ctrl+A Ctrl+M</td>
<td>Press CTRL and A together then press M, or Press CTRL and A together then press CTRL and M together. This combination produces the same results that you get if you key in REDISPLAY followed by RESET.</td>
</tr>
<tr>
<td>Clear</td>
<td>Ctrl+L</td>
<td>Press CTRL and L together. This combination fills the screen with blanks and places the cursor at the first character position on the screen. MASTER RESET key combinations remain active.</td>
</tr>
<tr>
<td>Enter</td>
<td>Ctrl+M</td>
<td>Press CTRL and M together. This combination sends the data on the screen to the host system.</td>
</tr>
</tbody>
</table>

CSI stands for Control Sequence Indicator.

Operation of PF and PA keys

This section describes the PF and PA keys for TELNET operations when you are using a terminal that is not part of the 3270 family.

When you press a key combination, a code that represents the 3270 equivalent symbol is sent to the application with which you have established TELNET communication. The application that you are using controls how these codes are used.

<table>
<thead>
<tr>
<th>PF Key</th>
<th>Input Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1</td>
<td>Press ESC then press 1</td>
</tr>
<tr>
<td>PF2</td>
<td>Press ESC then press 2</td>
</tr>
<tr>
<td>PF3</td>
<td>Press ESC then press 3</td>
</tr>
<tr>
<td>PF4</td>
<td>Press ESC then press 4</td>
</tr>
<tr>
<td>PF5</td>
<td>Press ESC then press 5</td>
</tr>
<tr>
<td>PF6</td>
<td>Press ESC then press 6</td>
</tr>
<tr>
<td>PF7</td>
<td>Press ESC then press 7</td>
</tr>
<tr>
<td>PF8</td>
<td>Press ESC then press 8</td>
</tr>
</tbody>
</table>
PF9  Press ESC then press 9
PF10 Press ESC then press 0
PF11 Press ESC then press the period key (.)
PF12 Press ESC then press =
PF13 Press ESC ESC then press 1
PF14 Press ESC ESC then press 2
PF15 Press ESC ESC then press 3
PF16 Press ESC ESC then press 4
PF17 Press ESC ESC then press 5
PF18 Press ESC ESC then press 6
PF19 Press ESC ESC then press 7
PF20 Press ESC ESC then press 8
PF21 Press ESC ESC then press 9
PF22 Press ESC ESC then press 0
PF23 Press ESC ESC then press .
PF24 Press ESC ESC then press =

PA Key Input Keys
PA1  Press Ctrl+P then press 1
PA2  Press Ctrl+P then press 2
PA3  Press Ctrl+P then press 3

Sense codes for special key operation
Table 29 describes the sense codes that are returned with the error messages for special key operations when you are using TELNET.

For information about error messages, see z/OS Communications Server: IP Messages Volume 1 (EZA).

Table 29. Sense codes

<table>
<thead>
<tr>
<th>Sense code</th>
<th>Problem description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8001</td>
<td>The command data length is less than 0 bytes.</td>
</tr>
<tr>
<td>8002</td>
<td>There is not enough data in an ERASE/WRITE or an ERASE/WRITE ALTERNATE command.</td>
</tr>
<tr>
<td>8003</td>
<td>There is not enough data in a WRITE command.</td>
</tr>
<tr>
<td>8004</td>
<td>There is not enough data in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>21001</td>
<td>WCC is not a character. The error occurred in a WRITE command.</td>
</tr>
<tr>
<td>21002</td>
<td>There is not enough data in a START FIELD subcommand. The error occurred in a WRITE command.</td>
</tr>
<tr>
<td>21003</td>
<td>There is not enough data in a START FIELD EXTENDED subcommand. The error occurred in a WRITE command.</td>
</tr>
<tr>
<td>21004</td>
<td>There is not enough data in a MODIFY FIELD subcommand. The error occurred in a WRITE command.</td>
</tr>
</tbody>
</table>
Table 29. Sense codes (continued)

<table>
<thead>
<tr>
<th>Sense code</th>
<th>Problem description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21005</td>
<td>There is not enough data in a SET BUFFER ADDRESS subcommand. The error occurred in a WRITE command.</td>
</tr>
<tr>
<td>21006</td>
<td>There is not enough data in a REPEAT TO ADDRESS subcommand. The error occurred in a WRITE command.</td>
</tr>
<tr>
<td>21007</td>
<td>There is not enough data in an ERASE UNPROTECTED TO ADDRESS subcommand. The error occurred in a WRITE command.</td>
</tr>
<tr>
<td>22001</td>
<td>The specified attribute is not a character. The error occurred in a START FIELD subcommand.</td>
</tr>
<tr>
<td>22002</td>
<td>A current buffer addressing error occurred. The error occurred in a START FIELD subcommand.</td>
</tr>
<tr>
<td>23001</td>
<td>An addressing error for the current buffer occurred. The error occurred in a START FIELD EXTENDED subcommand.</td>
</tr>
<tr>
<td>23002</td>
<td>The specified attribute is not a character. The error occurred in a START FIELD EXTENDED subcommand.</td>
</tr>
<tr>
<td>23003</td>
<td>The specified attribute is not an acceptable attribute type. The error occurred in a START FIELD EXTENDED subcommand.</td>
</tr>
<tr>
<td>24001</td>
<td>A current buffer addressing error occurred. The error occurred in a MODIFY FIELD subcommand.</td>
</tr>
<tr>
<td>24002</td>
<td>The specified attribute is not a character. The error occurred in a MODIFY FIELD subcommand.</td>
</tr>
<tr>
<td>24003</td>
<td>The specified attribute is not an acceptable attribute type. The error occurred in a MODIFY FIELD subcommand.</td>
</tr>
<tr>
<td>25001</td>
<td>The specified address is incorrect. The error occurred in a SET BUFFER ADDRESS subcommand.</td>
</tr>
<tr>
<td>26001</td>
<td>A current buffer addressing error occurred. The error occurred in a PROGRAM TAB subcommand.</td>
</tr>
<tr>
<td>28001</td>
<td>A current buffer addressing error occurred. The error occurred in a REPEAT TO ADDRESS subcommand.</td>
</tr>
<tr>
<td>28002</td>
<td>The specified address is incorrect. The error occurred in a REPEAT TO ADDRESS subcommand.</td>
</tr>
<tr>
<td>29001</td>
<td>A current buffer addressing error occurred. The error occurred in an ERASE UNPROTECTED TO ADDRESS subcommand.</td>
</tr>
<tr>
<td>29002</td>
<td>The specified address is incorrect. The error occurred in an ERASE UNPROTECTED TO ADDRESS subcommand.</td>
</tr>
<tr>
<td>30001</td>
<td>A current buffer addressing error occurred. The error occurred in the Write Data Process.</td>
</tr>
<tr>
<td>36001</td>
<td>Another structured field appeared after the READ PARTITION field. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>36002</td>
<td>The length is too long or too short in the READ PARTITION field. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>36003</td>
<td>A reserved character is incorrect in the READ PARTITION field. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>36004</td>
<td>The type is incorrect in the READ PARTITION field. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>38001</td>
<td>The length is too long or too short in the ERASE/RESET field. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>Sense code</td>
<td>Problem description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>38002</td>
<td>The Partition ID is incorrect in the ERASE/RESET field. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>39001</td>
<td>The length is too long or too short in the SET REPLY mode. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>39002</td>
<td>The Partition ID is incorrect in the SET REPLY mode. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
<tr>
<td>39003</td>
<td>The Reply Mode is incorrect in the SET REPLY mode. The error occurred in a WRITE STRUCTURED FIELD command.</td>
</tr>
</tbody>
</table>
| 40001      | The 3270 outbound data stream contains a command that is not in this list.  
  * WRITE  
  * ERASE/ WRITE  
  * ERASE/ WRITE ALTERNATE  
  * ERASE ALL UNPROTECTED  
  The error occurred in a WRITE STRUCTURED FIELD command. |
| 50001      | SO/SI is on longer a pair. SET REPLY mode. The error occurred in the GRFTOMAP Process. |
Table 30 lists the Internet Control Message Protocol (ICMP) types and codes from *TCP/IP Illustrated, Volume 1 The Protocols*, by W. Richard Stevens.

**Table 30. ICMP types and codes**

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply</td>
</tr>
</tbody>
</table>
| 3    | 0    | destination unreachable  
|      | 1    | host unreachable  
|      | 2    | protocol unreachable  
|      | 3    | port unreachable  
|      | 4    | fragmentation needed  
|      | 5    | source route failed  
|      | 6    | destination network unknown  
|      | 7    | destination host unknown  
|      | 8    | source host isolated  
|      | 9    | destination network administratively prohibited  
|      | 10   | destination host administratively prohibited  
|      | 11   | network unreachable for ToS  
|      | 12   | host unreachable for ToS  
|      | 13   | communication administratively prohibited by filtering  
|      | 14   | host precedence violation  
|      | 15   | precedence cutoff in effect  
| 4    | 0    | source quench  
| 5    | redirect  
|      | 0    | redirect for network  
|      | 1    | redirect for host  
|      | 2    | redirect for type of service and network  
|      | 3    | redirect for type of service and host  
| 8    | 0    | echo request  
| 9    | 0    | router advertisement  
| 10   | 0    | router solicitation  
| 11   | time exceeded  
|      | 0    | time-to-live equals 0 during transmit  
|      | 1    | time-to-live equals 0 during reassembly  
| 12   | parameter problem  
|      | 0    | IP header bad  
|      | 1    | required option missing  
| 13   | 0    | timestamp request  
| 14   | 0    | timestamp reply  |
Table 30. ICMP types and codes (continued)

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0</td>
<td>information request</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>information reply</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>address mask request</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>address mask reply</td>
</tr>
</tbody>
</table>

Table 31 lists the Internet Control Message Protocol for IPv6 (ICMPv6) types and codes.

Table 31. ICMPv6 types and codes

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>destination unreachable</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>no route to destination</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>communication with destination administratively prohibited</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>beyond scope of source address</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>address unreachable</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>port unreachable</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>packet too big</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>time exceeded</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>hop limit exceeded in transit</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>fragment reassembly time exceeded</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>parameter problem</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>erroneous header field encountered</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>unrecognized Next Header type encountered</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>unrecognized IPv6 option encountered</td>
</tr>
<tr>
<td>128</td>
<td>0</td>
<td>echo request</td>
</tr>
<tr>
<td>129</td>
<td>0</td>
<td>echo reply</td>
</tr>
<tr>
<td>130</td>
<td>0</td>
<td>group membership query</td>
</tr>
<tr>
<td>131</td>
<td>0</td>
<td>group membership reply</td>
</tr>
<tr>
<td>132</td>
<td>0</td>
<td>group membership reduction</td>
</tr>
<tr>
<td>133</td>
<td>0</td>
<td>router solicitation</td>
</tr>
<tr>
<td>134</td>
<td>0</td>
<td>router advertisement</td>
</tr>
<tr>
<td>135</td>
<td>0</td>
<td>neighbor solicitation</td>
</tr>
<tr>
<td>136</td>
<td>0</td>
<td>neighbor advertisement</td>
</tr>
<tr>
<td>137</td>
<td>0</td>
<td>redirect</td>
</tr>
</tbody>
</table>
Appendix E. How to read a syntax diagram

This syntax information applies to all commands and statements included in this document that do not have their own syntax described elsewhere in this document.

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line (the main path).

Symbols and punctuation

The following symbols are used in syntax diagrams:

| Symbol | Description
|--------|----------------
|       | Marks the beginning of the command syntax.
|       | Indicates that the command syntax is continued.
|       | Marks the beginning and end of a fragment or part of the command syntax.
| ❖❖❖❖ | Marks the end of the command syntax.

You must include all punctuation such as colons, semicolons, commas, quotation marks, and minus signs that are shown in the syntax diagram.

Parameters

The following types of parameters are used in syntax diagrams.

**Required**
- Required parameters are displayed on the main path.

**Optional**
- Optional parameters are displayed below the main path.

**Default**
- Default parameters are displayed above the main path.

Parameters are classified as keywords or variables. For the TSO and MVS console commands, the keywords are not case sensitive. You can code them in uppercase or lowercase. If the keyword appears in the syntax diagram in both uppercase and lowercase, the uppercase portion is the abbreviation for the keyword (for example, OPERand).

For the z/OS UNIX commands, the keywords must be entered in the case indicated in the syntax diagram.

Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a data set is a variable.

Syntax examples

In the following example, the USER command is a keyword. The required variable parameter is `user_id`, and the optional variable parameter is `password`. Replace the variable parameters with your own values.
Longer than one line

If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.

The first line of a syntax diagram that is longer than one line

The continuation of the subcommands, parameters, or both

Required operands

Required operands and values appear on the main path line.

You must code required operands and values.

Choose one required item from a stack

If there is more than one mutually exclusive required operand or value to choose from, they are stacked vertically.

Optional values

Optional operands and values appear below the main path line.

You can choose not to code optional operands and values.

Choose one optional operand from a stack

If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically below the main path line.

Repeating an operand

An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The comma means that each operand or value must be separated from the next by a comma. If no comma appears in the returning arrow, the operand or value must be separated from the next by a blank.
Selecting more than one operand
An arrow returning to the left above a group of operands or values means more
than one can be selected, or a single one can be repeated.

Nonalphanumeric characters
If a diagram shows a character that is not alphanumeric (such as parentheses,
periods, commas, and equal signs), you must code the character as part of the
syntax. In this example, you must code OPERAND=(001,0.001).

Blank spaces in syntax diagrams
If a diagram shows a blank space, you must code the blank space as part of the
syntax. In this example, you must code OPERAND=(001 FIXED).

Default operands
Default operands and values appear above the main path line. TCP/IP uses the
default if you omit the operand entirely.

Variables
A word in all lowercase italics is a variable. Where you see a variable in the syntax,
you must replace it with one of its allowable names or values, as defined in the text.

Syntax fragments
Some diagrams contain syntax fragments, which serve to break up diagrams that
are too long, too complex, or too repetitious. Syntax fragment names are in mixed
case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

Syntax fragment:

```
1ST_OPERAND,2ND_OPERAND,3RD_OPERAND
```
Appendix F. Related protocol specifications (RFCs)

This appendix lists the related protocol specifications for TCP/IP. The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. Some of these protocols are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement these particular functions or protocols. These become the de facto standards, on which the TCP/IP protocol suite is built.

These documents can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

where:

nnnn  Is the RFC number.
TXT  Is the text format.
PS  Is the PostScript format.

You can see Internet drafts at http://www.ietf.org/ID.html. See "Draft RFCs" on page 416 for draft RFCs implemented in z/OS V1R4 Communications Server.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to service@nic.ddn.mil with a subject line of RFC nnnn for text versions or a subject line of RFC nnnn.PS for PostScript versions. To request a copy of the RFC index, send a message with a subject line of RFC INDEX.

For more information, contact nic@nic.ddn.mil.

Many RFCs are available online. Hard copies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available using FTP from the NIC at the following Web address: http://www.rfc-editor.org/rfc.html.

Use FTP to download the files, using the following format:

RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS

Many features of TCP/IP Services are based on the following RFCs:

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title and Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>768</td>
<td>User Datagram Protocol J.B. Postel</td>
</tr>
<tr>
<td>791</td>
<td>Internet Protocol J.B. Postel</td>
</tr>
<tr>
<td>792</td>
<td>Internet Control Message Protocol J.B. Postel</td>
</tr>
<tr>
<td>793</td>
<td>Transmission Control Protocol J.B. Postel</td>
</tr>
<tr>
<td>821</td>
<td>Simple Mail Transfer Protocol J.B. Postel</td>
</tr>
</tbody>
</table>
822  Standard for the Format of ARPA Internet Text Messages D. Crocker
823  DARPA Internet Gateway R.M. Hinden, A. Sheltzer
826  Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware D.C. Plummer
854  Telnet Protocol Specification J.B. Postel, J.K. Reynolds
855  Telnet Option Specification J.B. Postel, J.K. Reynolds
856  Telnet Binary Transmission J.B. Postel, J.K. Reynolds
857  Telnet Echo Option J.B. Postel, J.K. Reynolds
858  Telnet Suppress Go Ahead Option J.B. Postel, J.K. Reynolds
859  Telnet Status Option J.B. Postel, J.K. Reynolds
860  Telnet Timing Mark Option J.B. Postel, J.K. Reynolds
861  Telnet Extended Options—List Option J.B. Postel, J.K. Reynolds
862  Echo Protocol J.B. Postel
863  Discard Protocol J.B. Postel
864  Character Generator Protocol J.B. Postel
877  Standard for the Transmission of IP Datagrams over Public Data Networks J.T. Korb
885  Telnet End of Record Option J.B. Postel
896  Congestion Control in IP/TCP Internetworks J. Nagle
903  Reverse Address Resolution Protocol R. Finlayson, T. Mann, J.C. Mogul, M. Theimer
904  Exterior Gateway Protocol Formal Specification D.L. Mills
919  Broadcasting Internet Datagrams J.C. Mogul
922  Broadcasting Internet Datagrams in the Presence of Subnets J.C. Mogul
950  Internet Standard Subnetting Procedure J.C. Mogul, J.B. Postel
952  DoD Internet Host Table Specification K. Harrenstien, M.K. Stahl, E.J. Feinler
959  File Transfer Protocol J.B. Postel, J.K. Reynolds
974  Mail Routing and the Domain Name System C. Partridge
1006 ISO Transport Service on top of the TCP Version 3 M.T. Rose, D.E. Cass
1009 Requirements for Internet Gateways R.T. Braden, J.B. Postel
1011 Official Internet Protocols J. Reynolds, J. Postel
1014 XDR: External Data Representation Standard Sun Microsystems Incorporated
1027 Using ARP to Implement Transparent Subnet Gateways S. Carl-Mitchell, J.S. Quaterman
1032 Domain Administrators Guide M.K. Stahl
1033 Domain Administrators Operations Guide M. Lottor
<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1034</td>
<td>Domain Names—Concepts and Facilities</td>
<td>P.V. Mockapetris</td>
</tr>
<tr>
<td>1035</td>
<td>Domain Names—Implementation and Specification</td>
<td>P.V. Mockapetris</td>
</tr>
<tr>
<td>1042</td>
<td>Standard for the Transmission of IP Datagrams over IEEE 802 Networks</td>
<td>J.B. Postel, J.K. Reynolds</td>
</tr>
<tr>
<td>1055</td>
<td>Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP</td>
<td>J.L. Romkey</td>
</tr>
<tr>
<td>1058</td>
<td>Routing Information Protocol</td>
<td>C.L. Hedrick</td>
</tr>
<tr>
<td>1060</td>
<td>Assigned Numbers</td>
<td>J. Reynolds, J. Postel</td>
</tr>
<tr>
<td>1073</td>
<td>Telnet Window Size Option</td>
<td>D. Waitzman</td>
</tr>
<tr>
<td>1079</td>
<td>Telnet Terminal Speed Option</td>
<td>C.L. Hedrick</td>
</tr>
<tr>
<td>1091</td>
<td>Telnet Terminal-Type Option</td>
<td>J. VanBokkelen</td>
</tr>
<tr>
<td>1094</td>
<td>NFS: Network File System Protocol Specification</td>
<td>Sun Microsystems Incorporated</td>
</tr>
<tr>
<td>1096</td>
<td>Telnet X Display Location Option</td>
<td>G. Marcy</td>
</tr>
<tr>
<td>1101</td>
<td>DNS encoding of network names and other types</td>
<td>P.V. Mockapetris</td>
</tr>
<tr>
<td>1112</td>
<td>Host Extensions for IP Multicasting</td>
<td>S. Deering</td>
</tr>
<tr>
<td>1118</td>
<td>Hitchhiker’s Guide to the Internet</td>
<td>E. Krol</td>
</tr>
<tr>
<td>1122</td>
<td>Requirements for Internet Hosts—Communication Layers</td>
<td>R.T. Braden</td>
</tr>
<tr>
<td>1123</td>
<td>Requirements for Internet Hosts—Application and Support</td>
<td>R.T. Braden</td>
</tr>
<tr>
<td>1155</td>
<td>Structure and Identification of Management Information for TCP/IP-Based Internets</td>
<td>M.T. Rose, K. McCloghrie</td>
</tr>
<tr>
<td>1156</td>
<td>Management Information Base for Network Management of TCP/IP-Based Internets</td>
<td>K. McCloghrie, M.T. Rose</td>
</tr>
<tr>
<td>1157</td>
<td>Simple Network Management Protocol (SNMP)</td>
<td>J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin</td>
</tr>
<tr>
<td>1158</td>
<td>Management Information Base for Network Management of TCP/IP-based Internets: MIB-II</td>
<td>M.T. Rose</td>
</tr>
<tr>
<td>1179</td>
<td>Line Printer Daemon Protocol</td>
<td>The Wollongong Group, L. McLaughlin III</td>
</tr>
<tr>
<td>1180</td>
<td>TCP/IP Tutorial</td>
<td>T.J. Socolofsky, C.J. Kale</td>
</tr>
<tr>
<td>1184</td>
<td>Telnet Linemode Option</td>
<td>D. Borman</td>
</tr>
<tr>
<td>1187</td>
<td>Bulk Table Retrieval with the SNMP</td>
<td>M.T. Rose, K. McCloghrie, J.R. Davin</td>
</tr>
<tr>
<td>1188</td>
<td>Proposed Standard for the Transmission of IP Datagrams over FDDI Networks</td>
<td>D. Katz</td>
</tr>
<tr>
<td>1191</td>
<td>Path MTU Discovery</td>
<td>J. Mogul, S. Deering</td>
</tr>
<tr>
<td>1198</td>
<td>FYI on the X Window System</td>
<td>R.W. Scheifler</td>
</tr>
</tbody>
</table>
1389  *RIP Version 2 MIB Extension*  G. Malkin

1390  *Transmission of IP and ARP over FDDI Networks*  D. Katz

1393  *Traceroute Using an IP Option*  G. Malkin

1397  *Default Route Advertisement In BGP2 And BGP3 Versions of the Border Gateway Protocol*  D. Haskin

1398  *Definitions of Managed Objects for the Ethernet-Like Interface Types*  F. Kastenholz

1416  *Telnet Authentication Option*  D. Borman, ed.

1464  *Using the Domain Name System to Store Arbitrary String Attributes*  R. Rosenbaum

1469  *IP Multicast over Token-Ring Local Area Networks*  T. Pusateri

1535  *A Security Problem and Proposed Correction With Widely Deployed DNS Software*  E. Gavron

1536  *Common DNS Implementation Errors and Suggested Fixes*  A. Kumar, J. Postel, C. Neuman, P. Danzig, S. Miller

1537  *Common DNS Data File Configuration Errors*  P. Beertema

1540  *IAB Official Protocol Standards*  J.B. Postel

1571  *Telnet Environment Option Interoperability Issues*  D. Borman

1572  *Telnet Environment Option*  S. Alexander

1577  *Classical IP and ARP over ATM*  M. Laubach

1583  *OSPF Version 2*  J. Moy

1591  *Domain Name System Structure and Delegation*  J. Postel


1594  *FYI on Questions and Answers: Answers to Commonly Asked “New Internet User” Questions*  A.N. Marine, J. Reynolds, G.S. Malkin

1695  *Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2*  M. Ahmed, K. Tesink

1706  *DNS NSAP Resource Records*  B. Manning, R. Colella

1713  *Tools for DNS debugging*  A. Romao

1723  *RIP Version 2—Carrying Additional Information*  G. Malkin

1766  *Tags for the Identification of Languages*  H. Alvestrand

1794  *DNS Support for Load Balancing*  T. Brisco

1832  *XDR: External Data Representation Standard*  R. Srinivasan

1850  *OSPF Version 2 Management Information Base*  F. Baker, R. Coltun

1876  *A Means for Expressing Location Information in the Domain Name System*  C. Davis, P. Vixie, T. Goodwin, I. Dickinson

1886  *DNS Extensions to support IP version 6*  S. Thomson, C. Huitema

1901  *Introduction to Community-Based SNMPv2*  J. Case, K. McCloghrie, M. Rose, S. Waldbusser
<table>
<thead>
<tr>
<th>Year</th>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1912</td>
<td>Common DNS Operational and Configuration Errors</td>
<td>D. Barr</td>
</tr>
<tr>
<td>1918</td>
<td>Address Allocation for Private Internets</td>
<td>Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear</td>
</tr>
<tr>
<td>1939</td>
<td>Post Office Protocol-Version 3</td>
<td>J. Myers, M. Rose</td>
</tr>
<tr>
<td>1981</td>
<td>Path MTU Discovery for IP version 6</td>
<td>J. McCann, S. Deering, J. Mogul</td>
</tr>
<tr>
<td>1982</td>
<td>Serial Number Arithmetic</td>
<td>R. Elz, R. Bush</td>
</tr>
<tr>
<td>1995</td>
<td>Incremental Zone Transfer in DNS</td>
<td>M. Ohta</td>
</tr>
<tr>
<td>1996</td>
<td>A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)</td>
<td>P. Vixie</td>
</tr>
<tr>
<td>2010</td>
<td>Operational Criteria for Root Name Servers</td>
<td>B. Manning, P. Vixie</td>
</tr>
<tr>
<td>2011</td>
<td>SNMPv2 Management Information Base for the Internet Protocol Using SMIv2</td>
<td>K. McCloghrie</td>
</tr>
<tr>
<td>2012</td>
<td>SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2</td>
<td>K. McCloghrie</td>
</tr>
<tr>
<td>2013</td>
<td>SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2</td>
<td>K. McCloghrie</td>
</tr>
<tr>
<td>2052</td>
<td>A DNS RR for specifying the location of services (DNS SRV)</td>
<td>A. Gulbrandsen, P. Vixie</td>
</tr>
<tr>
<td>2065</td>
<td>Domain Name System Security Extensions</td>
<td>D. Eastlake, C. Kaufman</td>
</tr>
<tr>
<td>2096</td>
<td>IP Forwarding Table MIB</td>
<td>F. Baker</td>
</tr>
<tr>
<td>2104</td>
<td>HMAC: Keyed-Hashing for Message Authentication</td>
<td>H. Krawczyk, M. Bellare, R. Canetti</td>
</tr>
<tr>
<td>2132</td>
<td>DHCP Options and BOOTP Vendor Extensions</td>
<td>S. Alexander, R. Droms</td>
</tr>
<tr>
<td>2133</td>
<td>Basic Socket Interface Extensions for IPv6</td>
<td>R. Gilligan, S. Thomson, J. Bound, W. Stevens</td>
</tr>
</tbody>
</table>
Secure Domain Name System Dynamic Update D. Eastlake

Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM) C. Allocchio

Resolution of Uniform Resource Identifiers using the Domain Name System R. Daniel, M. Mealling

OSPF Version 2 J. Moy

Clarifications to the DNS Specification R. Elz, R. Bush


The Use of RSVP with IETF Integrated Services J. Wroclawski

Specification of the Controlled-Load Network Element Service J. Wroclawski

Specification of Guaranteed Quality of Service S. Shenker, C. Partridge, R. Guerin

General Characterization Parameters for Integrated Service Network Elements S. Shenker, J. Wroclawski

Use of DNS Aliases for Network Services M. Hamilton, R. Wright

FTP Security Extensions M. Horowitz, S. Lunt

Key Exchange Delegation Record for the DNS R. Atkinson

The Interfaces Group MIB Using SMIV2 K. McCloghrie, F. Kastenholz

A Legal Basis for Domain Name Allocation O. Vaughn

The TLS Protocol Version 1.0 T. Dierks, C. Allen

Negative Caching of DNS Queries (DNS NCACHE) M. Andrews

Classless IN-ADDR.ARPA delegation H. Eidnes, G. de Groot, P. Vixie

Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIV2 M. Greene, J. Luciani, K. White, T. Kuo

OSPF Version 2 J. Moy

Domain Names and Company Name Retrieval J. Klensin, T. Wolf, G. Oglesby

A Convention for Using Legal Names as Domain Names O. Vaughn

TN3270 Enhancements B. Kelly

IP Version 6 Addressing Architecture R. Hinden, M. O’Dell, S. Deering

An IPv6 Aggregatable Global Unicast Address Format R. Hinden, M. O’Dell, S. Deering

IPv6 Multicast Address Assignments R. Hinden, S. Deering

Feature negotiation mechanism for the File Transfer Protocol P. Hethmon, R. Elz

FTP Extensions for IPv6 and NATs M. Allman, S. Ostermann, C. Metz

Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden


IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten
Transmission of IPv6 Packets over Ethernet Networks M. Crawford
Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers K. Nichols, S. Blake, F. Baker, D. Black
Domain Name System Security Extensions D. Eastlake
Storage of Diffie-Hellman Keys in the Domain Name System (DNS) D. Eastlake
Basic Socket Interface Extensions for IPv6 R. Gilligan, S. Thomson, J. Bound, W. Stevens
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen
SNMP Applications D. Levi, P. Meyer, B. Stewart
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie
Internationalization of the File Transfer Protocol B. Curtin
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson
Non-Terminal DNS Name Redirection M. Crawford
Multicast Listener Discovery (MLD) for IPv6 S. Deering, W. Fenner, B. Haberman
IPv6 Router Alert Option C. Partridge, A. Jackson
Definitions of Managed Objects for Service Level Agreements Performance Monitoring K. White
Secret Key Transaction Authentication for DNS (TSIG) P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington
DNS Extensions to Support IPv6 Address Aggregation and Renumbering M. Crawford, C. Huitema
Telnet Authentication Option T. Ts'o, ed., J. Altman
Telnet Authentication: Kerberos Version 5 T. Ts'o
Telnet Data Encryption Option T. Ts'o
Telnet Encryption: DES 64 bit Cipher Feedback T. Ts'o
Telnet Encryption: DES 64 bit Output Feedback T. Ts'o, ed.
Policy Core Information Model—Version 1 Specification B. Moore, E. Ellesson, J. Strassner, A. Westerinen

Draft RFCs

Several areas of IPv6 implementation include elements of the following draft RFCs and are subject to change during the RFC review process.
Advanced Sockets API for IPv6
   W. Richard Stevens, Matt Thomas, Erik Nordmark, Tatuya Jinmei

Basic Socket Interface Extensions for IPv6
   R.E. Gilligan, S. Thomson, J. Bound, J. McCann, W. R. Stevens

Default Address Selection for IPv6
   R. Draves

Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
   A. Conta, S. Deering

IP Version 6 Addressing Architecture
   R. Hinden, S. Deering
Appendix G. Information APARs

This appendix lists information APARs for IP and SNA documents.

Notes:
1. Information APARs contain updates to previous editions of the manuals listed below. Documents updated for V1R4 are complete except for the updates contained in the information APARs that may be issued after V1R4 documents went to press.
2. Information APARs are predefined for z/OS V1R4 Communications Server and may not contain updates.
3. Information APARs for OS/390 documents are in the document called OS/390 DOC APAR and PTF ++HOLD Documentation, which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IDDOCMST/CCONTENTS.
4. Information APARs for z/OS documents are in the document called z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation, which can be found at http://publibz.boulder.ibm.com:80/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS.

Information APARs for IP documents

Table 32 lists information APARs for IP documents.

Table 32. IP information APARs

<table>
<thead>
<tr>
<th>Title</th>
<th>z/OS CS V1R4</th>
<th>z/OS CS V1R2</th>
<th>CS for OS/390 2.10 and z/OS CS V1R1</th>
<th>CS for OS/390 2.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP API Guide</td>
<td>ii13255</td>
<td>ii12861</td>
<td>ii12371</td>
<td>ii11635</td>
</tr>
<tr>
<td>IP CICS Sockets Guide</td>
<td>ii13257</td>
<td>ii12862</td>
<td></td>
<td>ii11626</td>
</tr>
<tr>
<td>IP Configuration</td>
<td></td>
<td></td>
<td></td>
<td>ii11620</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ii12068</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ii12353</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ii12649</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ii13018</td>
</tr>
<tr>
<td>IP Configuration Guide</td>
<td>ii13244</td>
<td>ii12498</td>
<td>ii12362</td>
<td>ii11620</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii13087</td>
<td>ii12493</td>
<td>ii11630</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ii13006</td>
<td>ii13230</td>
</tr>
<tr>
<td>IP Configuration Reference</td>
<td>ii13245</td>
<td>ii12499</td>
<td>ii12363</td>
<td>ii11631</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ii12494</td>
<td>ii12883</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ii12712</td>
<td></td>
</tr>
<tr>
<td>IP Diagnosis</td>
<td>ii13249</td>
<td>ii12503</td>
<td>ii12366</td>
<td>ii11628</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ii12495</td>
<td></td>
</tr>
<tr>
<td>IP Messages Volume 1</td>
<td>ii13250</td>
<td>ii12857</td>
<td>ii12367</td>
<td>ii11630</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii13229</td>
<td></td>
<td>13230</td>
</tr>
<tr>
<td>IP Messages Volume 2</td>
<td>ii13251</td>
<td>ii12858</td>
<td>ii12368</td>
<td>ii11631</td>
</tr>
<tr>
<td>IP Messages Volume 3</td>
<td>ii13252</td>
<td>ii12859</td>
<td>ii12369</td>
<td>ii11632</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12990</td>
<td>ii12883</td>
</tr>
<tr>
<td>IP Messages Volume 4</td>
<td>ii13253</td>
<td>ii12860</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Migration</td>
<td>ii13242</td>
<td>ii12497</td>
<td>ii12361</td>
<td>ii11618</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1990, 2002
### Table 32. IP information APARs (continued)

<table>
<thead>
<tr>
<th>Title</th>
<th>z/OS CS V1R4</th>
<th>z/OS CS V1R2</th>
<th>CS for OS/390 2.10 and z/OS CS V1R1</th>
<th>CS for OS/390 2.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Network and Application Design Guide</td>
<td>ii13243</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Network Print Facility</td>
<td></td>
<td>ii12864</td>
<td></td>
<td>ii11627</td>
</tr>
<tr>
<td>IP Programmer’s Reference</td>
<td>ii13256</td>
<td>ii12505</td>
<td></td>
<td>ii11634</td>
</tr>
<tr>
<td>IP and SNA Codes</td>
<td>ii13254</td>
<td>ii12504</td>
<td>ii12370</td>
<td>ii11917</td>
</tr>
<tr>
<td>IP User’s Guide</td>
<td></td>
<td></td>
<td>ii12365 ii13060</td>
<td>ii11625</td>
</tr>
<tr>
<td>IP User’s Guide and Commands</td>
<td>ii13247</td>
<td>ii12501</td>
<td>ii12365 ii13060</td>
<td>ii11625</td>
</tr>
<tr>
<td>IP System Admin Guide</td>
<td>ii13248</td>
<td>ii12502</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quick Reference</td>
<td>ii13246</td>
<td>ii12500</td>
<td>ii12364</td>
<td></td>
</tr>
</tbody>
</table>

### Information APARs for SNA documents

Table 33 lists information APARs for SNA documents.

### Table 33. SNA information APARs

<table>
<thead>
<tr>
<th>Title</th>
<th>z/OS CS V1R4</th>
<th>z/OS CS V1R2</th>
<th>CS for OS/390 2.10 and z/OS CS V1R1</th>
<th>CS for OS/390 2.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anynet SNA over TCP/IP</td>
<td></td>
<td></td>
<td></td>
<td>ii11922</td>
</tr>
<tr>
<td>Anynet Sockets over SNA</td>
<td></td>
<td></td>
<td></td>
<td>ii11921</td>
</tr>
<tr>
<td>CSM Guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP and SNA Codes</td>
<td>ii13254</td>
<td>ii12504</td>
<td>ii12370</td>
<td>ii11917</td>
</tr>
<tr>
<td>SNA Customization</td>
<td>ii13240</td>
<td>ii12872</td>
<td>ii12388</td>
<td>ii11923</td>
</tr>
<tr>
<td>SNA Diagnosis</td>
<td>ii13236</td>
<td>ii12490 ii13034</td>
<td>ii12389</td>
<td>ii11915</td>
</tr>
<tr>
<td>SNA Messages</td>
<td>ii13238</td>
<td>ii12491</td>
<td>ii12382 ii12383</td>
<td>ii11916</td>
</tr>
<tr>
<td>SNA Network Implementation Guide</td>
<td>ii13234</td>
<td>ii12487</td>
<td>ii12381</td>
<td>ii11911</td>
</tr>
<tr>
<td>SNA Operation</td>
<td>ii13237</td>
<td>ii12489</td>
<td>ii12384</td>
<td>ii11914</td>
</tr>
<tr>
<td>SNA Migration</td>
<td>ii13233</td>
<td>ii12486</td>
<td>ii12386</td>
<td>ii11910</td>
</tr>
<tr>
<td>SNA Programming</td>
<td>ii13241</td>
<td>ii13033</td>
<td>ii12385</td>
<td>ii11920</td>
</tr>
<tr>
<td>Quick Reference</td>
<td>ii13246</td>
<td>ii12500</td>
<td>ii12364</td>
<td>ii11913</td>
</tr>
<tr>
<td>SNA Resource Definition Reference</td>
<td>ii13235</td>
<td>ii12488</td>
<td>ii12380 ii12567</td>
<td>ii11912 ii12568</td>
</tr>
<tr>
<td>SNA Resource Definition Samples</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNA Data Areas</td>
<td>ii13239</td>
<td>ii12492</td>
<td>ii12387</td>
<td>ii11617</td>
</tr>
</tbody>
</table>

### Other information APARs

Table 34 on page 421 lists information APARs not related to documents.
### Table 34. Non-document information APARs

<table>
<thead>
<tr>
<th>Content</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMPROUTE</td>
<td>ii12026</td>
</tr>
<tr>
<td>iQDIO</td>
<td>ii11220</td>
</tr>
<tr>
<td>index of recommended maintenance for VTAM</td>
<td>ii11220</td>
</tr>
<tr>
<td>CSM for VTAM</td>
<td>ii12657</td>
</tr>
<tr>
<td>CSM for TCP/IP</td>
<td>ii12658</td>
</tr>
<tr>
<td>AHHC, MPC, and CTC</td>
<td>ii01501</td>
</tr>
<tr>
<td>DLUR/DLUS for z/OS V1R2</td>
<td>ii12986</td>
</tr>
<tr>
<td>Enterprise Extender</td>
<td>ii12223</td>
</tr>
<tr>
<td>Generic resources</td>
<td>ii10986</td>
</tr>
<tr>
<td>HPR</td>
<td>ii10953</td>
</tr>
<tr>
<td>MNPS</td>
<td>ii10370</td>
</tr>
<tr>
<td>Performance</td>
<td>ii11710</td>
</tr>
<tr>
<td></td>
<td>ii11711</td>
</tr>
<tr>
<td></td>
<td>ii11712</td>
</tr>
</tbody>
</table>
Appendix H. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to z/OS TSO/E Primer, z/OS TSO/E User's Guide, and z/OS ISPF User's Guide Volume 1 for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.
IBM may not offer all of the products, services, or features discussed in this document. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs
and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel  
IBM Corporation  
P.O.Box 12195  
3039 Cornwallis Road  
Research Triangle Park, North Carolina 27709-2195  
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly
tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

This product includes cryptographic software written by Eric Young.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats of unlicensed books and the z/OS Licensed Product Library (LK3T-4307), which contains BookManager and PDF formats of licensed books.
Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
Advanced Peer-to-Peer Networking
AFP
AD/Cycle
AIX
AIX/ESA
AnyNet
APL2
AS/400
AT
BookManager
BookMaster
CBPDO
C/370
CICS
CICS/ESA
C/MVS
Common User Access
C Set ++
CT
CUA
DATABASE 2
DatagLANce
DB2
DFSMS
DFSMShfp
DFSMShsm
DFSMS/MVS
DPI
Domino
DRDA
eNetwork
Enterprise Systems Architecture/370
ESA/390
ESCON
eServer
ES/3090
ES/9000
ES/9370
EtherStreamer
Extended Services
FAA
Micro Channel
MVS
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
MQ
Natural
NetView
Network Station
Nways
Notes
NTune
NTuneNCP
OfficeVision/MVS
OfficeVision/VM
Open Class
OpenEdition
OS/2
OS/390
OS/400
Parallel Sysplex
Personal System/2
PR/SM
PROFS
PS/2
RACF
Resource Link
Resource Measurement Facility
RETAIL
RFM
RISC System/6000
RMF
RS/6000
S/370
S/390
SAA
SecureWay
Slate
SP
SP2
SQL/DS
System/360
Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

DB2 and NetView are registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the U.S., other countries, or both.

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see [http://www.intel.com/sites/corporate/tradmarx.htm](http://www.intel.com/sites/corporate/tradmarx.htm).

Other company, product, and service names may be trademarks or service marks of others.
Index

Special characters
- d (RSH parameter) 370
- l (RSH parameter) 370
- s port (RSH parameter) 370
! (FTP subcommand) 130

Numerics
3270 DBCS Transform mode conversion types
DECKANJI 26
EUCKANJI 26
HANGEUL 26
JIS78KJ 26
JIS83KJ 26
KSC5601 26
SJISKANJI 26
TCHEBON 26
3270 DBCS Transform mode description 24
3270 DBCS Transform mode terminal types
TTY 26
VT100 26
VT226 26
802.3 network protocol 2

A
A (FTP TYPE parameter) 276
accessibility features 423
ACCOUNT (FTP subcommand) 131
account_information (FTP ACCOUNT parameter) 131
addresses
class A network 6
description 2, 3
loopback 6
path 297
TCP/IP (idx4) 3
AIX files 386
all
FTP HELP parameter 172
LPQ parameter 324
allocating data sets
FTP input and output data sets 35
new data sets 86
alphanumeric cursor (GDDM/MVS) 344
ANFontn (GDDM/MVS option) 350
AO (TELNET subcommand) 14
APL2 Character Set keyboard 362, 389
APPEND (FTP subcommand) 132
appending a local data set to a remote host 132
applications, functions, and protocols (continued)
Simple Mail Transfer Protocol (SMTP) 285
Telnet 9
AS/400 files 387
ASAtrans (FTP.DATA parameter) 64
ASAtrans parameter 243
ASCII
ASCII control characters 22
FTP JIS78KJ parameter 174
FTP JIS83KJ parameter 175
FTP subcommand 134
transferring binary data to EBCDIC 276
transferring text data to EBCDIC 276
AT host
LPQ parameter 324
LPR parameter 328
LPRM parameter 339
attached graphics cursor (GDDM/MVS) 344
authorizations for data and programs.6
AUTOMOUNT
FTP LOCSITE and SITE parameter 185, 244
FTP.DATA parameter 64
AUTORECALL
FTP LOCSITE and SITE parameter 185, 244
FTP.DATA parameter 64
AUTOTAPEMOUNT (FTP.DATA parameter) 64
AYT (TELNET subcommand) 15

B
B parameter
FTP MODE 213
FTP TYPE 276
BATCH (SMTPNOTE parameter) 288
batch commands
DATA 299
EXPN 301
HELO 302
HELP 303
MAIL FROM 304
NOOP 305
QUEU 306
QUIT 308
RCPT TO 309
RSET 310
STATS 293
TICK 311
VERB 312
VRFY 313
batch SMTP examples 315
batch, submitting FTP requests in batch 91
BIG5
FTP subcommand 135
LPR parameter 328
BINARY
FTP subcommand 137
LPR parameter 328
blinking character attribute (GDDM/MVS) 344

© Copyright IBM Corp. 1990, 2002
BLKsize (FTP.DATA parameter) 64
BLOCK (FTP subcommand) 138
block mode (FTP) 213
BBlocks (FTP LOCSITE and SITE parameter) 185, 244
BLOCKSIZE
   FTP LOCSITE and SITE parameter 64, 186, 244
   FTP.DATA parameter 64
BRK (TELNET subcommand) 16
BUFNO
   FTP LOCSITE and SITE parameter 244
   FTP.DATA parameter 64
BURST (LPR parameter) 329
changing
directory of a VM FTP server 141
directory of an MVS FTP server 140
directory on a foreign host 140
local site defaults using FTP.DATA 63
to the parent of the current directory 143
TSO user ID password 281
working directory 179
working level qualifier 179
carriage return, suppressing (TELNET) 23
CC
   LPR parameter 329
   SMTPNOTE parameter 287
   CCONNTIME (FTP.DATA parameter) 64
   CCTRANS (FTP.DATA parameter) 65, 124
   CD (FTP subcommand) 140
   CDUP (FTP subcommand) 143
clearing the data path (TELNET) 21
class A network addresses 6
class, LPR parameter 329
CLEAR (FTP subcommand) 145
copies
   data sets to a foreign host 225
   files from a foreign host 166
   multiple data sets to a foreign host 214
   multiple files from a foreign host 207
creating
   a directory on a foreign host 211
   a PDS on the local host 181
   an input data set with the SQL query 118
CTRLConn (FTP.DATA parameter) 66
cyberspace
   connecting to a foreign host FTP server 38, 217
   control characters, sending 22
   converting DBCS mail 316
codes
   data sets transfer
   methods 50, 386
types 50
data transfer
   using z/OS Communications Server  xv
datatransfer
   ASCII 134
   EBCDIC 161
   EUCKANJI 162
   HANGEUL 170
   IBMKANJI 173
   image 137
   JIS78KJ 174
   JIS83KJ 175
   KSC-5601 176
   SJISKANJI 263
   TCHINESE 273
data set (FTP LMKDIR parameter) 181
data set name (LPR parameter) 329
commands (continued)
   REXEC 5
   RSH 369, 370
   SMSG (general user) 292
   SMSG (privileged user) 295
telnet
   communication media 1
   Communications Server for z/OS, commands 4
   Communications Server for z/OS, online information xv
   Compr (GDDMXD option) 352
   COMPRESS (FTP subcommand) 147
   compressed mode (FTP) 213
copying
   data sets transfer
   using z/OS Communications Server  xv
data transfer
   methods 50, 386
types 50
data transfer type conversion
   ASCII 134
   EBCDIC 161
   EUCKANJI 162
   HANGEUL 170
   IBMKANJI 173
   image 137
   JIS78KJ 174
   JIS83KJ 175
   KSC-5601 176
   SJISKANJI 263
   TCHINESE 273
   data set (FTP LMKDIR parameter) 181
   data set name (LPR parameter) 329

432  z/OS V1R4.0 CS: IP User's Guide and Commands
DATACLAS
   FTP LOCSITE and SITE parameter 189, 246
   FTP.DATA parameter 66
DATACTTIME (FTP.DATA parameter) 66
DATASET (SMTPNOTE parameter) 288
DATASETMODE (FTP LOCSITE and SITE parameter) 43, 189, 247
DATATIMEOUT (FTP.DATA parameter) 66
DATE (SMTP QUEU parameter) 306

DB2
   FTP LOCSITE and SITE parameter 189, 247
   FTP.DATA parameter 66
   SQL queries with FTP 117
   subsystems in FTP 119
   DB2 database support 117
   DB2PLAN (FTP.DATA parameter) 67

DBCS
   converting mail 316
   DBCS support for FTP 123
   DBCS support for SMTP 316
   DBCS support for TELNET 24
   DBCS translation tables 123
   setting transfer type 124
   DBCS subcommands
     QUOTE 229
     TYPE 276
     TYPE aliases 124

DCBDSN
   FTP LOCSITE and SITE parameter 190, 247
   FTP.DATA parameter 67
DCONNTIME (FTP.DATA parameter) 67
DEBUG
   FTP subcommand 149
   privileged user SMSG parameter 295
   Telnet parameter 10
DEBUG (FTP.DATA parameter) 67
Default Vector Symbol Set (GDDM/MVS) 344
DELETE (FTP subcommand) 153
deleting
   a job 106
   files on a foreign host 153
   multiple files on a foreign host 205
DELIMIT (FTP subcommand) 154
DEST (FTP LOCSITE and SITE parameter) 250
destination_file (FTP APPEND parameter) 132
detached graphics cursor (GDDM/MVS) 344
detectable fields (GDDM/MVS) 344
determining a foreign host operating system 272
DEV.NULL directory 141, 180
devices, network 2
DIR (FTP subcommand) 42, 43, 155, 157
directories
   changing the directory of a VM FTP server 141
   changing the directory of an MVS FTP server 140
   changing the directory on the foreign host 140
   changing the working directory 179
   changing to the parent of the current directory 143
   creating a directory on a foreign host 211
   DEV.NULL 141, 180
   obtaining a list of directory entries 155
   removing a directory from a foreign host 234
   directories (continued)
     transferring PDS directory information 386
     working with directories on the foreign host 41
     working with directories on the local host 45, 46
   directory
   FTP CD parameter 140
   FTP MKDIR parameter 211
   FTP RMDIR parameter 234
   DIRECTORY
   FTP LOCSITE and SITE parameter 190, 250
   FTP.DATA parameter 67
   DIRECTORYMODE
   FTP LOCSITE and SITE parameter 43, 190, 250
   FTP.DATA parameter 67
disability, physical 423
disconnecting from a host using FTP 146
DISK
   FTP DIR parameter 155
   FTP LS parameter 203
   displaying
   FTP help information 172
   local status information (FTP) 201
   TELNET help information 17
   the current working directory 227
   the current working level qualifier 202
   the file name delimiter 154
   the operating system name (FTP) 272
   the status of an FTP job 103
DNS, online information xvi
documents, licensed xvi
domain_name SMTP HELO parameter 302
dotted decimal notation 3
dump (FTP subcommand) 159
dump (FTP.DATA parameter) 67

e
E (FTP TYPE parameter) 276
EBCDIC
   FTP subcommand 161
   transferring binary data to ASCII 276
   transferring binary data to EBCDIC 276
   transferring text data to ASCII 276
   transferring text data to EBCDIC 276
electronic mail
   e-mail xv
electronic mail (idx4)
   sending and receiving 4
   encoding 67
ENCODING (FTP.DATA parameter) 67
ending a TELNET session 20
Enter (GDDMXD option) 353
establishing
   a connection to a foreign host 38
   default working directory 39
   Ethernet protocol 2
EUC (Extended UNIX Code) 162
EUCKANJI
   FTP subcommand 162
   LPR parameter 328
examples
  batch SMTP  315
  DEBUG command  295
  Generation Data Group (GDG)  88, 89
  JES  101
  LPQ  325
  LPR  336
  LPRM  340
  LPRSET  341
  nondelivery note  290
  preparing and sending mail  289
  SMTP STATS command  293
  SQL query output  122
  TELNET Help  17
  TSO RECEIVE command  290
  unknown recipient note  291
  using TELNET to log on to a foreign host  11

examples, FTP
  APPEND  132
  differences between DIR and LS output  42, 43
  establishing a connection  38
  FTP as a batch job  92
  FTP EXEC  94, 95
  GET and MGET  51
  issuing subcommands from the EXEC interface  96
  LMKDIR  181
  MKDIR  211
  PUT and MPUT  56
  showing the results of STATUS  267
  showing the results with and without DEBUG  149
  transferring data  50
  working with foreign directories  41
  working with local directories  46

EXEC interface usage  94
EXPIRE (privileged user SMSG parameter)  295
EXPN (SMTP command)  301
EXTENSIONS UTF8 (FTP.DATA parameter)  67

FTP
  command  4, 5, 30
  data transfer methods  50, 386
  DB2 subsystems for SQL queries  119
  DBCS support  123
  DD name support  58
  EXEC interface  94
  EXIT return codes  97
  FTP-supplied DB2 column headings  120
  internal error codes  100
  issuing subcommands from a data set  95
  logging on  35
  parameters  33
  reply codes  100
  security considerations  46
  subcommand codes  98
  transferring data  29, 49, 50
  user-level options  35, 75

FTP Client, configuring for SOCKS server  76
FTP examples
  FTP as a batch job  92
  FTP EXEC  95
  Generation Data Group (GDG)  88, 89

FTP format options
  NOSPREAD  120
  SPREAD  120
  SQLCOL  118

FTP requests in batch, JCL for  91
FTP subcommands
  !  130
  ACCOUNT  131
  APPEND  132
  ASCII  134
  BIG5  135
  BINARY  137
  BLOCK  138
  CD  140
  CDUP  143
  CLEAR  145
  CLOSE  146
  COMPRESS  147
  DEBUG  149
  DELETE  153
  DELIMIT  154
  DIR  155
  DUMP  159
  EBCDIC  161
  EUCKANJI  162

FILTER (LPR parameter)  330
foreign_file
  FTP DELETE parameter  153
  FTP GET parameter  166
  FTP MDELETE parameter  205
  FTP MGET parameter  207
  FTP PUT parameter  225

foreign_host
  FTP parameter  30
  RSH parameter  370
  TELNET parameter  10

format of batch SMTP command data sets  298
formatting batch SMTP command data sets  298

FTP
  command  4, 5, 30
  data transfer methods  50, 386
  DB2 subsystems for SQL queries  119
  DBCS support  123
  DD name support  58
  EXEC interface  94
  EXIT return codes  97
  FTP-supplied DB2 column headings  120
  internal error codes  100
  issuing subcommands from a data set  95
  logging on  35
  parameters  33
  reply codes  100
  security considerations  46
  subcommand codes  98
  transferring data  29, 49, 50
  user-level options  35, 75

FTP Client, configuring for SOCKS server  76
FTP examples
  FTP as a batch job  92
  FTP EXEC  95
  Generation Data Group (GDG)  88, 89

FTP format options
  NOSPREAD  120
  SPREAD  120
  SQLCOL  118

FTP requests in batch, JCL for  91
FTP subcommands
  !  130
  ACCOUNT  131
  APPEND  132
  ASCII  134
  BIG5  135
  BINARY  137
  BLOCK  138
  CD  140
  CDUP  143
  CLEAR  145
  CLOSE  146
  COMPRESS  147
  DEBUG  149
  DELETE  153
  DELIMIT  154
  DIR  155
  DUMP  159
  EBCDIC  161
  EUCKANJI  162

FILETYPE
  FTP LOCSITE and SITE parameter  190, 252
  FTP.DATA parameter  67

FILE
  FTP STRUCT parameter  270
  FTP TYPE parameter  276
  FEAT (FTP subcommand)  164
  FILE (FTP subcommand)  165

file name
  delimiter  154
  obtaining a list  203
  specifying  383

file transfer types
  ASCII  50, 134, 276, 386
  EBCDIC  50, 161, 276, 278, 386
  image  50, 137, 276, 278, 386
  kanji  50, 276, 278, 386

files
  AIX  386
  AS/400  387
  specifying  383

FILETYPE
  FTP LOCSITE and SITE parameter  190, 252
  FTP.DATA parameter  67

FILTER (LPR parameter)  330
foreign_file
  FTP DELETE parameter  153
  FTP GET parameter  166
  FTP MDELETE parameter  205
  FTP MGET parameter  207
  FTP PUT parameter  225

foreign_host
  FTP parameter  30
  RSH parameter  370
  TELNET parameter  10

format of batch SMTP command data sets  298
formatting batch SMTP command data sets  298

FTP
  command  4, 5, 30
  data transfer methods  50, 386
  DB2 subsystems for SQL queries  119
  DBCS support  123
  DD name support  58
  EXEC interface  94
  EXIT return codes  97
  FTP-supplied DB2 column headings  120
  internal error codes  100
  issuing subcommands from a data set  95
  logging on  35
  parameters  33
  reply codes  100
  security considerations  46
  subcommand codes  98
  transferring data  29, 49, 50
  user-level options  35, 75

FTP Client, configuring for SOCKS server  76
FTP examples
  FTP as a batch job  92
  FTP EXEC  95
  Generation Data Group (GDG)  88, 89

FTP format options
  NOSPREAD  120
  SPREAD  120
  SQLCOL  118

FTP requests in batch, JCL for  91
FTP subcommands
  !  130
  ACCOUNT  131
  APPEND  132
  ASCII  134
  BIG5  135
  BINARY  137
  BLOCK  138
  CD  140
  CDUP  143
  CLEAR  145
  CLOSE  146
  COMPRESS  147
  DEBUG  149
  DELETE  153
  DELIMIT  154
  DIR  155
  DUMP  159
  EBCDIC  161
  EUCKANJI  162
FTP subcommands (continued)

FEATure 164
FILE 165
GET 166
GLOB 168
HANGEUL 170
HELP 172
IBMKANJI 173
JIS78KJ 174
JIS83KJ 175
KSC5601 176
LANGUAGE 178
LCD 179
LMKDIR 181
LOCSITE 183
LOCSTAT 201
LPWD 202
LS 203
MDELETE 205
MGET 207
MKDIR 211
MODE 213
MPUT 214
NOOP 216
OPEN 217
PASS 218
PRIVATE 219
PROMPT 220
PROTECT 221
PROXY 222
PUT 225
PWD 227
QUIT 228
QUOTE 229
RECORD 231
RENAME 232
RESTART 233
RMDIR 234
SCHINESE 236
SENDPORT 238
SENDSITE 239
SITE 240
SJISKANJI 263
SRestart 265
STATUS 267
STREAM 269
STRUCTURE 270
SUNIQUE 271
SYSTEM 272
TCHINESE 273
TSO 275
TYPE 276
UCS2 280
USER 281

FTP-supported SQL data types (continued)

SMALLINT 117
TIME 117
TIMESTAMP 117
VARCHAR 117

FTP.DATA data set 63, 77
FTP.DATA parameter, data set
ASATrans 64
AUTOMount 64
AUTOPrompt 64
AUTOTAPEMount 64
BLKsize 64
BLOCKSsize 64
BUfno 64
CCONNTIME 64
CCTRANS 65
CHKPTInt 65
CHKPTPrefix 65
CIPHERSUITE 66
CLIENTERRCODES 66
CONDdisp 66, 71
CTRLConn 66
DATAClass 66
DATACTTIME 66
DATATimeOUT 66
DB2 66
DB2PLAN 67
DCbdsn 67
DCONNTIME 67
DEBUG 67
Directory 67
DIRECTORYMode 67
DUMP 67
ENCODING 67
EXTENSIONS UTF8 67
FILEtype 67
FTPKEEPALIVE 68
FWFriendly 68
INACTTIME 68
ISPFSstats 68
KEYRING 68
LRecl 68
MBDATAConn 68
MGmtclass 68
MIGratevol 68
MYOPENTIME 68
NETRCLEVEL 69
PRImary 69
QUOtesoverride 69
RDW 69
RECFm 69
RESTGet 70
RETpd 70
SBDataconn 70
SBSUB 70
SBSUBCHAR 70
SBTRANS 71
SECondary 71
SECURE_DATAConn 72
SECURE_DATAConn 72
SECURE_FTP 72
SECURE_MECHANISM 72
FTP.DATA parameter, data set (continued)
SECURE_PBSZ 72
SOCKSCONFIGFILE 72
SPACETYPE 72
SPRead 72
SQLCol 73
STORclass 73
TLSTIMEOUT 73
TRAILingblanks 73
TRUNCate 73
UCOUNT 73
UCSHOSTCS 73
UCSSUBL 74
UCSTRUNC 74
UMask 74
UNITNAME 74
VCOUNT 74
VOLUME 74
WRAPrecord 74
WRTAPEFastio 75
FTPKEEPALIVE (FTP.DATA parameter) 68
function keys for TELNET 23
FWFriendly (FTP.DATA parameter) 68

G
GColornn (GDDMXD option) 354
GDDM
   application limitations 343
displaying graphics on X Windows workstations xv
GDDM display limitations 344
GDDMXD
   CLIST 343, 345
   Graphics Window 348
GDDMXD/MVS
   APL2 character set keyboard 362, 389
   GDXAPLCS.MAP 362
   keyboard functions 361
   overview 343
target display, identifying 347
   TSO EXEC command 346
   usage 348
   using 343, 345
   X.DEFAULTS data set 348
GDDMXD/MVS user-specified options
   ANFontn 350
   CMap 351
   Compr 352
   GColornn 354
   Geometry 355
   GMCPnn 356
   HostRast 357
   XSync 359
   ZWL 360
GDDMXD/MVS with X Windows 343
Generation Data 88
Generation Data Group Support (GDG) 88
Geometry (GDDMXD option) 355
GET FTP subcommand 51, 166
getting started 1
GLOB (FTP subcommand) 168
GMCPnn (GDDMXD option) 356

H
HANGEUL
   FTP subcommand 170
   LPR parameter 331
HEADER, LPR parameter 331
HELO (SMTP command) 302
host
   foreign 41
   local 45
   LPQ parameter 324
   LPR parameter 331
   LPRM parameter 339
   names 6
   remote 3, 9, 41, 140, 146, 211, 214, 225, 316, 339,
   341, 387
host_name (FTP OPEN parameter) 217
HostRast (GDDMXD option) 357
hosts, using other 5
how TCP/IP uses networks 2
logging on
  to a host using Telnet 9
  to FTP 35
  to other hosts 4
LOOPBACK 6
LPD (line printer daemon) 323
LPQ (remote printing command)
  description 5, 324
  examples 325
  usage 325
LPR (remote printing command)
  description 5, 326
  examples 336
  usage 337
LPRM (remote printing command)
  description 5, 339
  examples 340
  usage 340
LPRSET (remote printing command)
  description 5, 341
  examples 341
  usage 342
LPWD (FTP subcommand) 202
LRECL
  FTP LOCSITE and SITE parameter 191, 253
  FTP.DATA parameter 68
ls (FTP subcommand) 42, 43, 203
MAIL (LPR parameter) 333
MAIL FROM (SMTP command) 304
mail transfer (SMTP) commands
  DATA 299
  EXPN 301
  HELO 302
  HELP 303
  MAIL FROM 304
  NOOP 305
  QUEU 306
  QUIT 308
  RCPT TO 309
  RSET 310
  STATS 293
  TICK 311
  VERB 312
  VRFY 313
mailbox
  SMTP EXPN parameter 301
  SMTP VRFY parameter 313
mapping values 362, 389
MBCS translation tables, and FTP 85
MBDATACONN 68
MBDATACONN (FTP.DATA parameter) 68
MDELETE (FTP subcommand) 205
MGET (FTP subcommand) 51, 207
MGMTCLAS
  FTP LOCSITE and SITE parameter 191, 254
  FTP.DATA parameter 68
MIGRATEVOL
  FTP LOCSITE and SITE parameter 191, 254
MIGRATEVOL (continued)

FTP DATA parameter 68
MKDIR (FTP subcommand) 211
MODE (FTP subcommand) 213
MPUT (FTP subcommand) 56, 214
MYOPENTIME (FTP DATA parameter) 68

N

NAME
FTP DIR parameter 155
FTP LS parameter 203
FTP STATUS parameter 267
LPR parameter 333
name server, description 3
names
host 6
network, description 3
printer 6
NDB 375
NETRC DATA data set 37, 366
NETRCLEVEL (FTP DATA parameter) 69
network address format 2
network devices 2
network names 3
network protocols
  802.3 2
  Ethernet 2
  SNA 2
token ring 2
  X.25 2
networks
  NJE 286
  TCP/IP 2
new_name (FTP RENAME parameter) 232
NewLiner (GDDMXD option) 358
NJE 285
NOASAtrans parameter 254
NOAUTOMOUNT (FTP LOCSITE and SITE parameter) 192, 254
NOBINARY (LPR parameter) 333
NOBURST (LPR parameter) 329
NOCC
  LPR parameter 329
  SMTPNOTE parameter 287
nodebug, privileged user SMSG parameter 295
nodes, descriptions 1, 2
noheader, LPR parameter 329
NOLinecount (LPR parameter) 331
NOOP
  FTP subcommand 216
  SMTP command 305
NOPORTSCRIPT (LPR parameter) 333
NORDW (FTP LOCSITE parameter) 192
NOSPREAD
  FTP format option 120
  FTP LOCSITE and SITE parameter 192, 254
notation system, dotted decimal 3
NOTOPMARGIN (LPR parameter) 333
NOTRACE (privileged user SMSG parameter) 295
NOTRAILINGBLANKS (FTP SITE parameter) 254
NOTYPE
  FTP EUCKANJI parameter 162
  FTP HANGEUL parameter 170
  FTP IBMKANJI parameter 173
  FTP JIS78KJ parameter 174
  FTP JIS83KJ parameter 175
  FTP KSC5601 parameter 177
  FTP SJISKANJI parameter 263
  FTP TCHINESE parameter 273
NOWRAPRECORD (FTP LOCSITE and SITE parameter) 193, 255

O

obtaining
  a list of directory names 155
  a list of file names 203
  status and system information 40
OFF
  GDDMXD parameter 346
  SMTP VERB parameter 312
ON
  GDDMXD parameter 346
  SMTP VERB parameter 312
OPEN (FTP subcommand) 217
original_name (FTP RENAME parameter) 232
OS/390 UNIX orexec/reexec 373
OS/390 UNIX REXEC
  command 5, 363, 372
  format 364, 373
requests, submitting in batch 366
overviews
  differences between DIR and LS output 42
  GDDMXD/MVS 343

P

PA1 (TELNET subcommand) 19
parameter
  FTP LOCSITE parameter 183
  FTP SITE parameter 240
parameters, FTP
  EXIT 30
  FOREIGN HOST 30
  PORT NUMBER 30
  TCP 30
  TIMEOUT 30
  TRACE 30
  TRANSLATE 30
parameters, SNALINK LU6.2 218
passing TSO commands to your local host 275
password
  FTP PASS parameter 218
  FTP USER parameter 281
password, use with FTP
  ACCOUNT 131
  PASS 218
  USER 281
path address (SMTP) 297
Index 439

PDS 181, 383, 384, 386
performing a DB2 SQL query
from an FTP client 121
from an FTP server 121
with FTP 117
physical network, description 1
pixel spacing (GDDM/MVS) 344
port numbers, description 3
port_number
FTP OPEN parameter 217
FTP parameter 30
TELNET parameter 10
ports, description 3
POSTSCRIPT (LPR parameter) 335
preparing and sending mail 289
preparing the FTP environment 49
primary
FTP LOCSITE and SITE parameter 193, 255
FTP DATA parameter 69
printer
LPQ parameter 324
LPR parameter 333
LPRM parameter 339
names 6
printer_host (LPRSET parameter) 341
printing
remote xv, 323
to or from other hosts 5
PRIVATE (FTP subcommand) 219
privileged user SMSS command 295
PROFILE command 39
PROMPT (FTP subcommand) 220
PROTECT (FTP subcommand) 221
protocols
description 2
File Transfer Protocol 29, 49
OS/390 UNIX Remote Execution Protocol 363, 372
Simple Mail Transfer Protocol 285
telnet Protocol 9
X Window System Protocol 343
PROXY
bounce attack 224
FTP subcommand 222
security 224
PUT (FTP subcommand) 56, 225
PWD (FTP subcommand) 39, 227
queue resolution (SMTP) (continued)
resolver wait 307
retry (SMTP) 306
spool (SMTP) 306
undeliverable (SMTP) 306
QUEUES (MSG parameter) 293
QUIT
FTP subcommand 228
SMTP command 308
TELNET subcommand 20
QUOTE
DBCS subcommand 229
FTP subcommand 229
QUOTESoverride (FTP.DATA parameter) 69
QUOTESoverride, FTP LOCSITE parameter 193
R
RACF 6, 39
RCPT TO (SMTP command) 309
RDW
FTP LOCSITE parameter 195
FTP DATA parameter 69
READTAPEFormat, FTP LOCSITE parameter 194
READTAPEFormat, FTP Site parameter 256
RECEIVE command 5
receiving mail 4
receiving spool output
in a group 105
individually 104
RECFM
FTP LOCSITE and SITE parameter 195, 257
FTP DATA parameter 69
recipient (SMTPNOTE parameter) 287
recipient_path_address (SMTP RCPT TO parameter) 309
RECORD (FTP subcommand) 231
Record Descriptor Words (RDW) 192
remote host logon xv
remote node, description 2
remote printing 323
remote printing commands
LPQ 324
LPR 326
LPRM 339
LPRSET 341
removing a directory (FTP) 234
RENAME (FTP subcommand) 232
renaming files on a foreign host 232
REPLACE
FTP GET parameter 166
FTP MGET parameter 207
requests in batch, submitting FTP 91
resizing the GDDMXD graphics window 348
responses, SMTP 314
RESTART (FTP subcommand) 233
restarting a checkpointed data transfer 233
RESTGet (FTP DATA parameter) 70
RETPD
FTP LOCSITE and SITE parameter 196, 257
FTP DATA parameter 70

Q
Qdisk (FTP LOCSITE and SITE parameter) 193, 255
qualifier (FTP LCD parameter) 179
query, LPRSET parameter 341
querying a connection (TELNET) 15
querying SMTP delivery queues 315
QUEU (SMTP command) 306
queue resolution (SMTP)
resolver completed 307
resolver error pending 307
resolver process 307
resolver retry 307
resolver send 307

Index 439
retrieving status information from a remote host 267
return codes, FTP 97
REUSE (SMTPNOTE parameter) 288
REXEC 363, 372
REXX command list language 33
RFC (request for comment)
   list of 409
RFC (request for comments)
   accessing online xv
RHOSTS.DATA data set 372
RMDIR (FTP subcommand) 234
RSCS 297
RSET (SMTP command) 310
RSH 369, 370
S
S (FTP MODE parameter) 213
sample FTP.DATA data set 77
SBCS translation tables, and FTP 85, 123
SBDataconn (FTP.DATA parameter) 70
SBSUB 70
SBTRANS (FTP.DATA parameter) 71, 124
SCHINESE
   FTP subcommand 236
   LPR parameter 333
SECONDARY
   FTP LOCSITE and SITE parameter 197, 259
   FTP.DATA parameter 71
SECURE_CTRLCONN (FTP.DATA parameter) 71
SECURE_DATACON (FTP.DATA parameter) 72
SECURE_FTP (FTP.DATA parameter) 72
SECURE_MECHANISM (FTP.DATA parameter) 72
SECURE_PBSZ (FTP.DATA parameter) 72
sender_path_address (SMTP MAIL FROM parameter) 304
sending
   ASCII control characters to a host in line mode 22
   break or attention keystroke to a host 16
   data using the QUOTE subcommand 229
   electronic mail using OS/390 UNIX sendmail 319
   electronic mail using SMTP commands 4, 285
   mail to a TCP network recipient 315
   PA1 keystroke to a host 19
site-specific information to a host 240
   uninterpreted string of data 229
SENDPORT (FTP subcommand) 238
SEDSITE (FTP subcommand) 239
sequential data sets 384
server
   description 2
   FTP HELP parameter 172
setting
   characteristics for an SQL query 118
   data set or file structure 270
   data transfer mode 213
   data transfer type 124, 276
setting up tcpipv3r1.GDXAPLCS.MAP 362
shortcut keys 423
SHUTDOWN (privileged user SMSG parameter) 295
SITE (FTP subcommand) 240
SITE parameters 240
SJISKANJI
   FTP subcommand 263
   LPR parameter 333
SMS 86, 186, 244
SMSG command
   general user 292
   privileged user 295
SMSGAUTHLIST statement 295
SMTP
   DBCS support 316
   description 4
   Exit, Anti-Spamming 291
   format of batch SMTP command data sets 298
   monitoring the status of 292
   responses 314
   SMSG interface 292
SMTP electronic mail
   nondelivery 290
   preparing and sending 289
   receiving 290
SMTP interfaces
   interactively 285
   JES 285
SMTPNOTE from your terminal 286
SNA network protocol 2
SOCKSCONFIGFILE (FTP.DATA parameter) 72
SPACEYTYPE (FTP.DATA parameter) 72
specifying
   column headings for an SQL query 119
   data sets and files 383
   report format of your output data set 120
   site information to the local host 183
   spreadsheet format of your output data set 120
   the DB2 subsystem to perform a query 119
   values for new data sets 86
SPREAD
   FTP LOCSITE and SITE parameter 120, 197, 259
   FTP.DATA parameter 72
SQL
   FTP-supported data types 117
   imbedded statements 117
   with FTP on the client 121
   with FTP on the server 121
SQL data type 117
SQLCOL
   FTP LOCSITE and SITE parameter 197, 259
   FTP.DATA parameter 73
SRestart (FTP subcommand) 265
STATS (SMSG parameter) 293
STATUS (FTP subcommand) 267
status and system information 40
STORCLASS
   FTP LOCSITE and SITE parameter 197, 259
   FTP.DATA parameter 73
store command (STOR) 271
STREAM (FTP subcommand) 269
stream mode (FTP) 213
string (FTP QUOTE parameter) 229
STRUCT (FTP subcommand) 270
SUBCOMMAND (FTP HELP parameter) 172
submitting
  FTP requests in batch 91
  job and automatically receiving output 106
  job using FTP 102
  requests without input and output data sets 94
  REXEC requests in batch 366
SQL query using FTP 121
SUNIQUE (FTP subcommand) 271
supplying
  a password to a foreign host 218
  account information to a foreign host 131
suppressing carriage return and line feed 23
SYNCH (TELNET subcommand) 21
SYSTEM (FTP subcommand) 272

T
  tables
    translation 123
target display (GDDM/MVS) 347
target_screen (XWINDOWS DISPLAY parameter) 347
target_server (XWINDOWS DISPLAY parameter) 347
tasks
  GDDMXD/MVS
    overview 343
  submitting a job
    steps for 102
  submitting a job and automatically receiving output
    steps for 106
  Using a DCBDSN model to create a new data set
    steps for 87
TCHINESE
  FTP subcommand 273
  LPR parameter 334
TCP, FTP parameter 30
TCP/IP
  addresses 3
  commands 4
  description 1
  layers 2
  networks 2
  online information xv
  protocol specifications 409
  understanding 1
TELNET
  3270 DBCS Transform Mode 24
  command 4, 10
  supported display stations 12
TELNET examples
  command format 10
  logging on to a foreign host 11
  logging onto a host using 9
  using Help 17
TELNET function keys
  in line mode 23
  in transparent mode 23
TELNET parameters
  DEBUG 10
  foreign_host 10
  Help 10
  Linemode 10
  TELNET parameters (continued)
    port_number 10
    TRANSLate data_set_name 10
  Telnet Protocol 9
TELNET subcommands
  AO 14
  AYT 15
  BRK 16
  HELP 17
  IP 18
  PA1 19
  QUIT 20
  SYNCH 21
  terminating
    access to JES 107
    output of TELNET information 14
testing
  commands with loopback 6
  FTP connection 216
  throughput with *DEV.NULL 141, 180
  TICK (SMTP command) 311
  Timeout (FTP parameter) 30
  TITLE (LPR parameter) 334
  TLSTIMEOUT (FTP.DATA parameter) 73
  TO (SMTPNOTE parameter) 287
toggling
  internal debug options (FTP) 149
  sending of port information 238
  sending of site information 239
  storage method 271
token ring network protocol 2
TOPMARGIN (LPR parameter) 335
TRACE
  FTP parameter 30
  LPQ parameter 324
  LPR parameter 335
  LPRM parameter 339
  LPRSET parameter 341
  privileged user SMSG parameter 295
TRACKS (FTP LOCSITE and SITE parameter) 197, 259
trademark information 428
TRAILingblanks (FTP.DATA parameter) 73
TRAILINGBLANKS, FTP SITE parameter 259
transferring
  data sets between hosts 4
  data using FTP 29, 49, 50
  DBCS data sets with FTP 123
  PDS directory information 386
  TRANslate data_set_name
    FTP parameter 30
    TELNET parameter 10
  TRANSLATETABLE (LPR parameter) 335
  transparent mode 23
  TRUNCate (FTP.DATA parameter) 73
TSO
  entering TCP/IP commands 1
  FTP subcommand 275
  Session Manager 348
TSO commands
  EDIT 286, 289
TSO commands (continued)

PROFILE 39
RECEIVE 290

TYPE
  DBCS subcommand 276
  FTP subcommand 276
  LPQ parameter 325
  LPR parameter 335
  LPRM parameter 339
  LPRSET parameter 341

U

UCOUNT (FTP.DATA parameter) 73
UCS2 (FTP subcommand) 280
UCSHOSTCS (FTP.DATA parameter) 73
UCSSUB (FTP.DATA parameter) 74
UCSTRUNC (FTP.DATA parameter) 74
UMask (FTP.DATA parameter) 74
undelivered notes 290
UNIT (FTP LOCSITE and SITE parameter) 199, 261
Unit of Work (UOW) 5
UNITNAME (FTP.DATA parameter) 74
USCFXLATE (LPR parameter) 335
user
  ID 6
  password 6
USER (FTP subcommand) 281
user interface
  ISPF 423
  TSO/E 423
USER name (FTP parameter) 335
user_id (FTP USER parameter) 281
user-specified options (GDDM/MVS) 348
uses of TCP/IP
  data transfer 4, 29, 49
  electronic mail 4, 285
  printing on other hosts 5, 323
  remote login 4, 9
  using other hosts 5, 363, 372

V

VCOUNT (FTP.DATA parameter) 74
VERB (SMTP command) 312
VERSION
  LPQ parameter 325
  LPR parameter 335
  LPRM parameter 339
  LPRSET parameter 341
visual appearance (GDDM/MVS) 344
VM files 388
VOLUME
  FTP LOCSITE and SITE parameter 199, 261
  FTP.DATA parameter 74
VRFY (SMTP command) 313
VTAM 20
VTAM, online information xv

W

WAN (wide area network) 1
well-known ports, description 3
what you need to get started 6
WIDTH (LPR parameter) 335
working directory 39, 179
working with directories
  on the foreign host 41
  on the local host 45, 46
working-level qualifier 179, 202
WRAPRECORD
  FTP LOCSITE and SITE parameter 199, 261
  FTP.DATA parameter 74, 75

X

X Window System 343
X.25 network protocol 2
XLATETABLE (LPR parameter) 335
XSync (GDDMXD option) 359

Z

z/OS, documentation library listing xvii
z/OS, listing of documentation available 419
ZWL (GDDMXD option 360

442  z/OS V1R4.0 CS: IP User’s Guide and Commands
Communicating Your Comments to IBM

If you especially like or dislike anything about this document, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. 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.

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 are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this document.
- If you prefer to send comments by FAX, use this number: 1-800-254-0206
- If you prefer to send comments electronically, use this network ID: usib2hpd@vnet.ibm.com

Make sure to include the following in your note:

- Title and publication number of this document
- Page number or topic to which your comment applies.
### Readers’ Comments — We’d Like to Hear from You

**z/OS Communications Server**  
**IP User’s Guide and Commands**  
**Version 1 Release 4**

**Publication No. SC31-8780-02**

**Overall, how satisfied are you with the information in this book?**

<table>
<thead>
<tr>
<th>Overall satisfaction</th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How satisfied are you that the information in this book is:**

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to find</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to understand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicable to your tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you?  [ ] Yes  [ ] No

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.

---

**Name**

**Address**

**Company or Organization**

**Phone No.**
Program Number: 5694–A01 and 5655–G52

Printed in U.S.A.